

目录 Contents

第一部分  界面及按键设计

第二部分  逻辑思路

第三部分  效果图

■ 界面及按钮设计

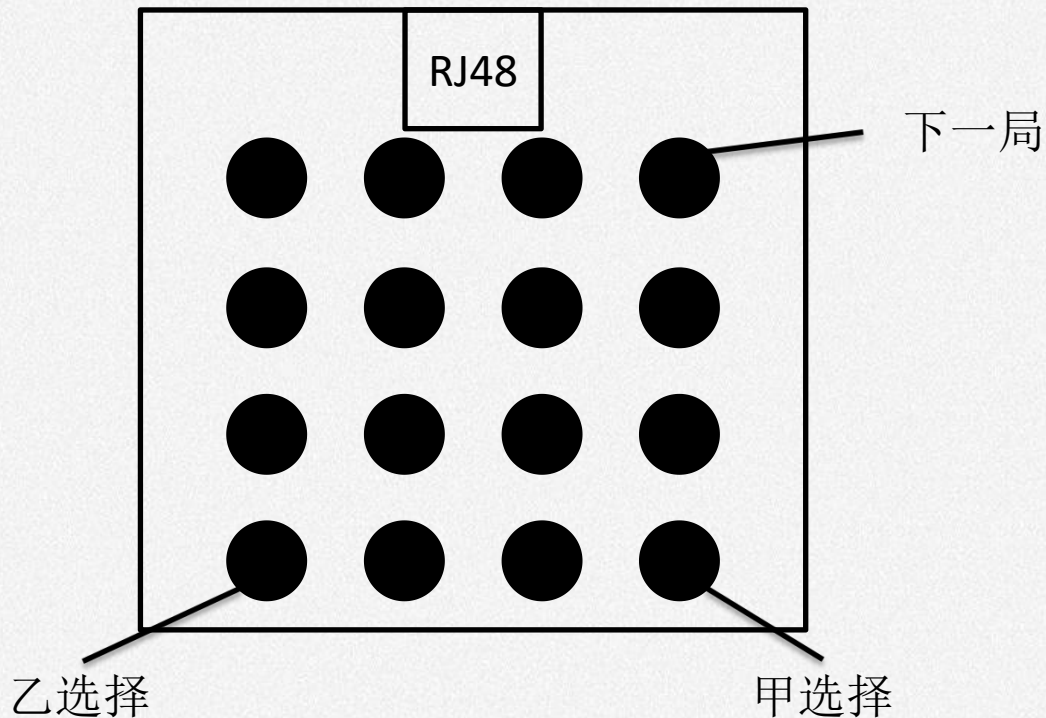
第一步，界面设计如下：

左上角写“甲：”，右上角写“乙：”，顶部中间写“胜者为”
两边中间显示甲乙出的选择，可以为“石头”、“剪刀”、“布”
屏幕中间写“得分：”，上面有不断刷新的“石头”、“剪刀”、
“布”三种字样，再上有数字表示局数，
下面依次写“甲：”、“乙：”



■ 界面及按钮设计

按钮需要4*4按钮作为输入，左下角表示甲的选择，右下角表示乙的选择，左上角表示开始下一局



思路

界面中会不停刷新“石头”、“剪刀”、“布”。在响应的阶段甲或者乙摁下，则视为选择，当双方选择完毕后，比较两者的选择，将结果显示在顶部，然后胜者得分加1

```
def.detect():
    global player1,player2,choose1,choose2,R,C,res
    if choose1==0:
        for i in range(0,4):
            R[i].low()
            for k in range(0,4):
                if k!=i:
                    R[k].high()
            for j in range(0,4):
                if i==0 and j==0 and C[j].value()==0:
                    pyb.delay(30)
                    if C[j].value()==0:
                        player1=1
                elif i==3 and j==0 and C[j].value()==0:
                    pyb.delay(30)
                    if C[j].value()==0:
                        res=1
    if choose2==0:
        for i in range(0,4):
            R[i].low()
            for k in range(0,4):
                if k!=i:
                    R[k].high()
            for j in range(0,4):
                if i==0 and j==2 and C[j].value()==0:
                    pyb.delay(30)
                    if C[j].value()==0:
                        player2=1
                elif i==3 and j==0 and C[j].value()==0:
                    pyb.delay(30)
                    if C[j].value()==0:
                        res=1
```

解析按键函数如左图，主要检测三个键，
对应三个全局变量：
甲的选择（player1）
乙的选择（player2）
重启（res）

逻辑思路

主程序大体如下，主要分为两部分，选择部分和判断部分：

```
· disp.scissor(64,64,0x0000)
· detect()
· if player1==1:#选择剪刀
· if player2==1:
· disp.putrect(64,64,16,32,0xffff)
· disp.stone(64,64,0x0000)
· detect()
· if player1==1:#选择石头
· if player2==1:
· disp.putrect(64,64,16,32,0xffff)
· disp.cloth(64,64,0x0000)
· detect()
· if player1==1:#选择布
· if player2==1:
· disp.putrect(64,64,16,16,0xffff)
· if choose1>0 and choose2>0:
· if res==1:
```

第一行，显示剪刀

第二行，检测是否有按键

第三第四行，判断甲和乙是否有任意一人选择，此时选择则为选择剪刀

第五行，显示空白，把显示剪刀的位置置位空白

之后选择石头与选择布与前五五行相同

最后两行为判断选择后的结果与开始下一局

详细代码将会在后面一一讲解

■ 思路

选择部分代码如下：

```
..disp.scissor(64,64,0x0000)
..detect()
}..if player1==1:#选择剪刀
    ...choose1=1
    ...disp.scissor(56,128,0x0000)#甲下方显示剪刀
    ...player1=0
}..if player2==1:
    ...choose2=1
    ...disp.scissor(56,0,0x0000)#乙下方显示剪刀
    ...player2=0
}..disp.putrect(64,64,16,32,0xffff)
```

上图是剪刀的部分，石头和布的部分基本相同

逻辑思路

判断部分：

choose1和choose2的值分别代表：

1-剪刀

2-石头

3-布

根据以上的值，判断输赢。之后改变score1和score2的值表示比分

```
if choose1>0 and choose2>0:
    if choose1==1:
        if choose2==1:
            disp.putChinese(0,48,UI_6,disp.BLACK)
        if choose2==2:
            disp.putChinese(0,48,UI_3,disp.BLACK)
            score2=score2+1
        if choose2==3:
            disp.putChinese(0,48,UI_2,disp.BLACK)
            score1=score1+1
    if choose1==2:
        if choose2==1:
            disp.putChinese(0,48,UI_2,disp.BLACK)
            score1=score1+1
        if choose2==2:
            disp.putChinese(0,48,UI_6,disp.BLACK)
        if choose2==3:
            disp.putChinese(0,48,UI_3,disp.BLACK)
            score2=score2+1
    if choose1==3:
        if choose2==1:
            disp.putChinese(0,48,UI_3,disp.BLACK)
            score2=score2+1
        if choose2==2:
            disp.putChinese(0,48,UI_2,disp.BLACK)
            score1=score1+1
        if choose2==3:
            disp.putChinese(0,48,UI_6,disp.BLACK)
    #更新甲得分
    disp.putrect(96,64,16,16,0xffff)
    disp.show_number(96,64,score1,disp.BLACK)
    #更新乙得分
    disp.putrect(112,64,16,16,0xffff)
    disp.show_number(112,64,score2,disp.BLACK)
    choose1=0
    choose2=0
```

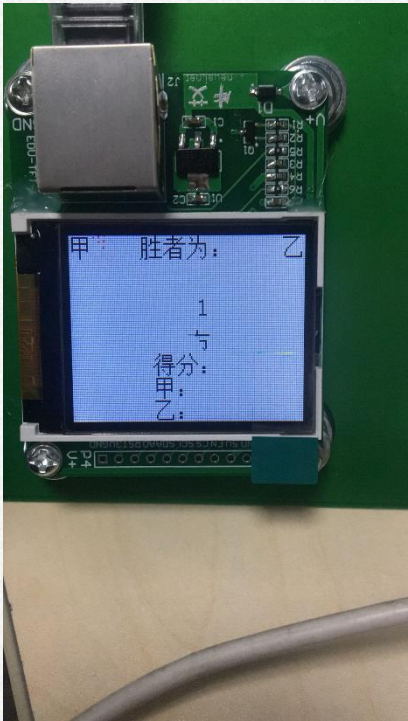

■ 思路

一局结束后摁下res键开始下一局

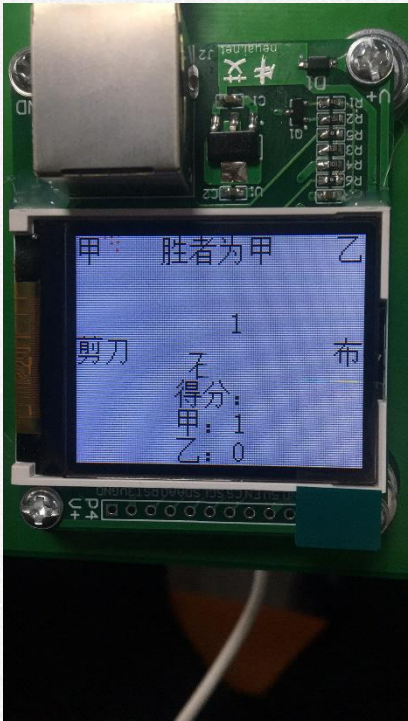
```
...if res==1:
...    choose1=0
...    choose2=0
...    disp.putrect(0,48,16,16,0xffff) #清除结果
...    disp.putrect(56,128,16,32,0xffff) #清除甲的选择
...    disp.putrect(56,0,16,32,0xffff) #清除乙的选择
...    disp.putrect(40,66,16,8,0xffff) #清除回合数
...    disp.show_number(40,66,round,disp.BLACK) #局数
...    round=round+1
...    res=0
```

效果显示

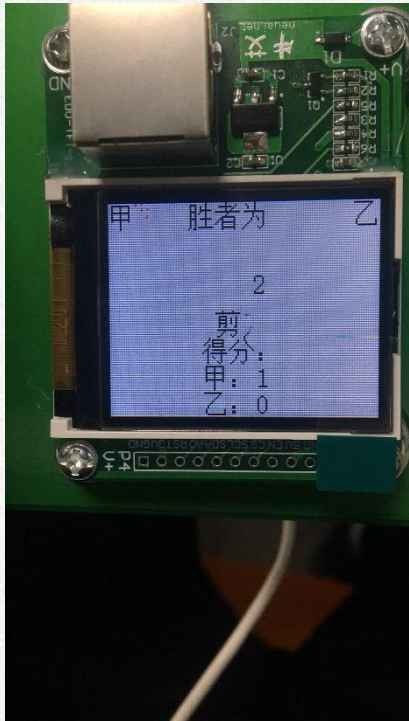
效果如图：



开始



第一局结束



开始下一局