




牛艾科技

由点到面

Python base program

目录Contents

第一部分  LED屏幕简介

第二部分  屏幕的本质——点阵

第三部分  LDO简介

第四部分  LED点阵编程

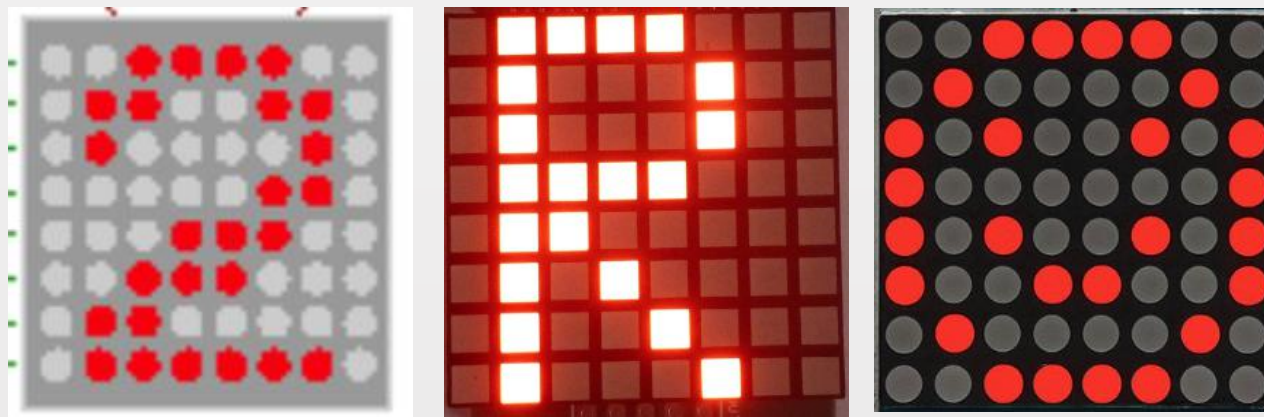
■ 常见的LED显示屏

- 公交报站
- 车站信息
- 广告招牌



■ LED点阵字符显示原理

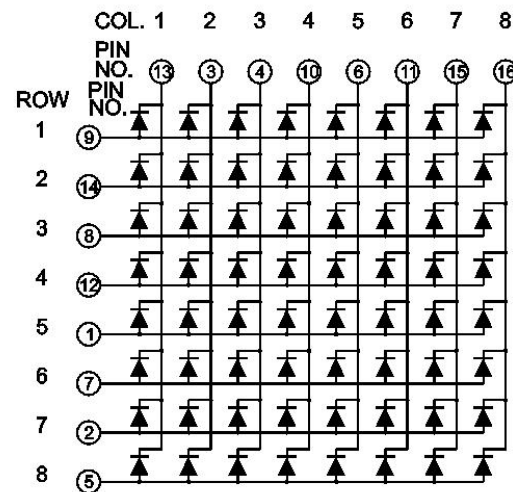
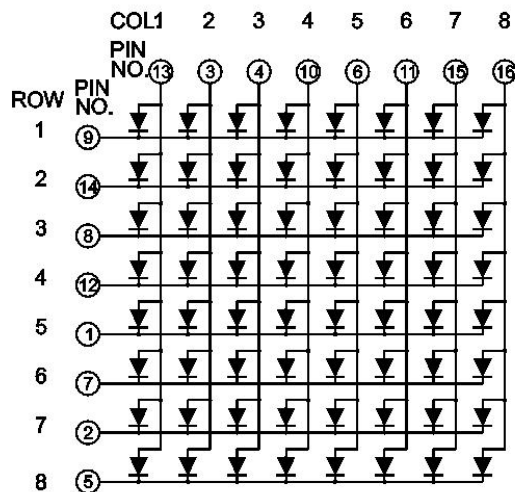
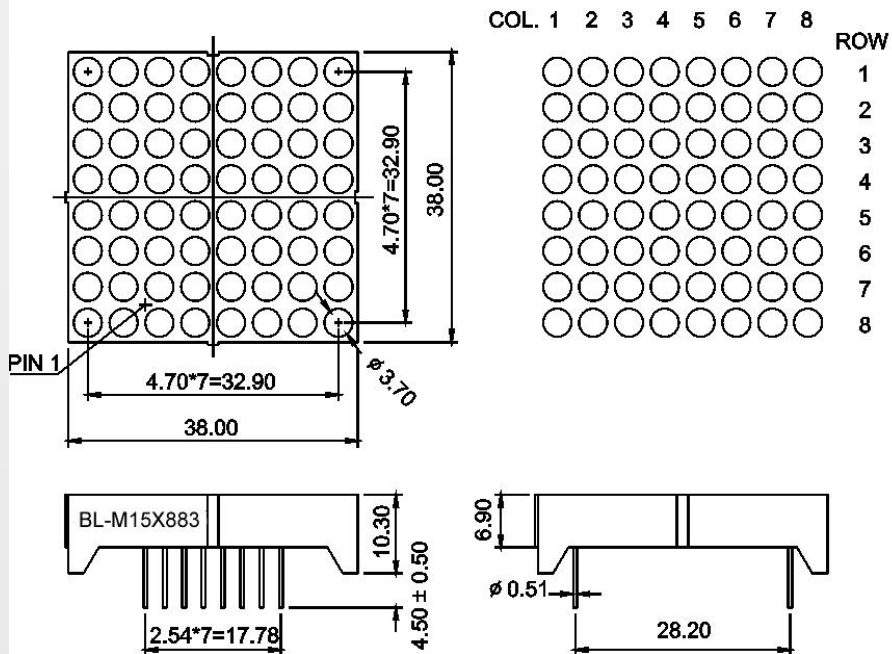
- LED点阵通过发光二极管组成，以灯珠亮灭来显示文字，图片，动画，视频等



■ 需要过多的IO控制LED

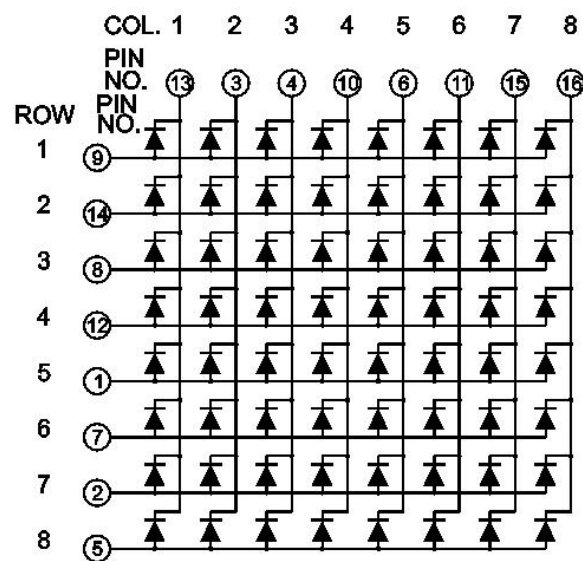
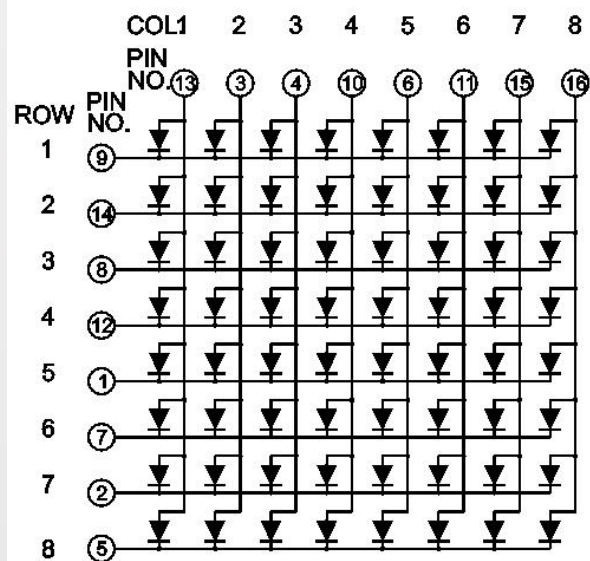
- 例如一个8x8的阵列，需要64个GPIO

解决方法：行列扫描，需要16个



LED点阵介绍

- LED分**共阳极**和共阴极
- LED点阵内部原理图



- 实物图

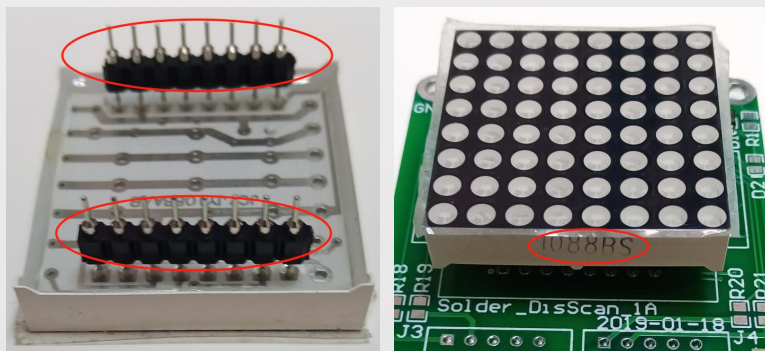


LED点阵电路板-清单

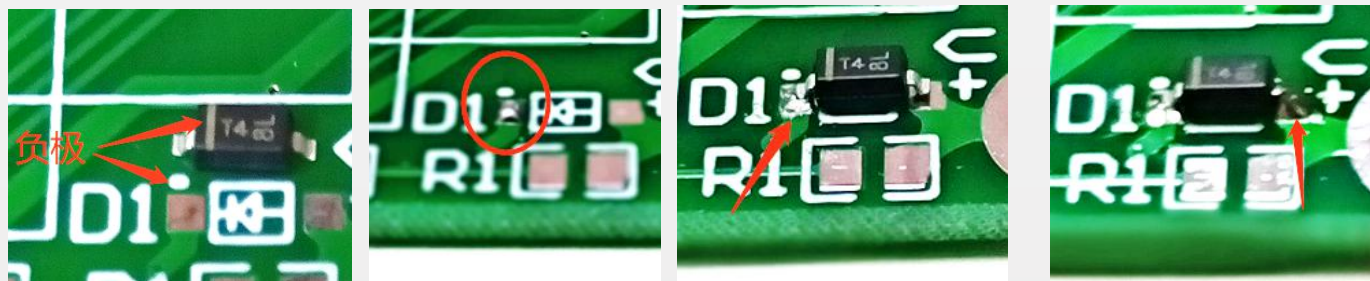
序号	规格/参数	位号	数量 需/配
1	PCB板	-	1/1
2	0805 10uF	C1,C2	2/3
3	1N5819W	D1	1/2
4	0805 RED	D2	1/2
5	8X8点阵	J1	1
6	RJ48	J3,J4	2
7	BSS84	Q1-Q8	8/10
8	0805 1KΩ	R1-R21	21/25
9	AMS1117-3.3	U1	1/1
10	4.5*4.5按键	S1-S4	4/4
11	8p单排插针	J1	2/2
12	磁柱、螺丝	-	4/4

LED点阵电路板-焊接

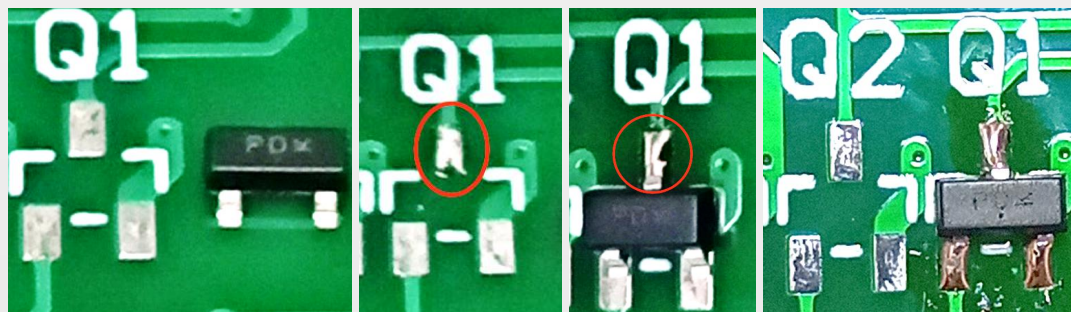
- 贴片类2脚器件焊接
 - 二极管为例，电阻，电容类同
- 贴片类3脚器件焊接
 - 三极管为例
- 电源芯片焊接
- 其他直插类器件焊接；点阵应装在单排插座再焊接，防止插座歪扭



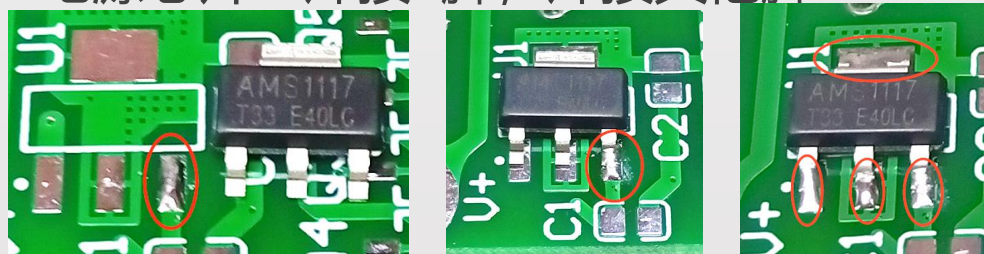
- 二极管：确认方向，焊接1脚，焊接2脚



- 三极管：焊接1脚，焊接2、3脚

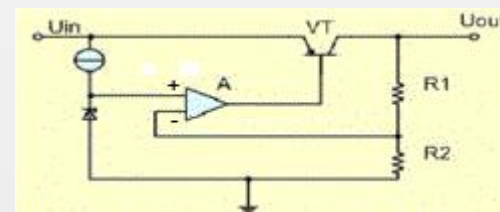


- 电源芯片：焊接1脚，焊接其他脚



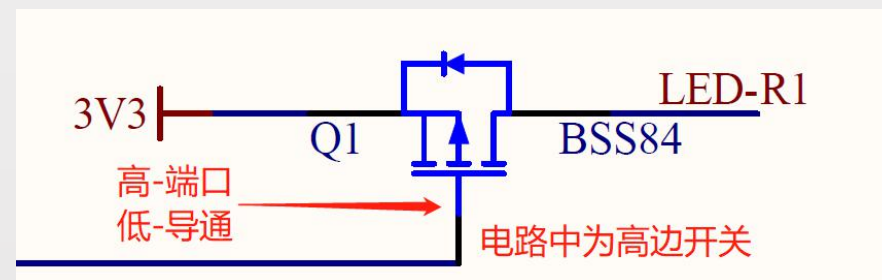
■ 什么是LDO

- LDO即low dropout regulator，是一种低压差线性稳压器。这是相对于传统的线性稳压器来说的。传统的线性稳压器，如78XX系列的芯片都要求输入电压要比输出电压至少高出2V~3V，否则就不能正常工作
- 右图为LDO，由调整管VT、取样电阻R1和R2、比较放大器A组成
- LDO常用于输入、输出电压差较小的情况，几十毫伏~几伏
- 更大压差情况例如输入大于输出十几甚至几十伏特，情况下多使用DC/DC
- DC/DC表示的是高压（低压）直流电源变换为低压（高压）直流电源，DC(直流)是相对于家用市电AC（交流）的。
- LDO：噪声低，成本低，效率随压差增大而降低，
- DC/DC：噪声较高，成本较高，效率高，可以升压或降压



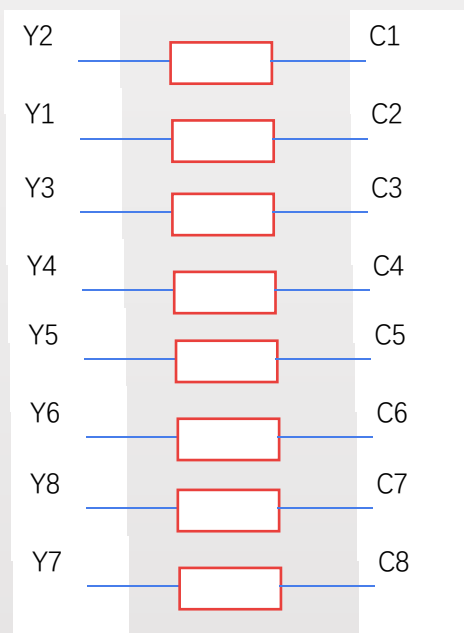
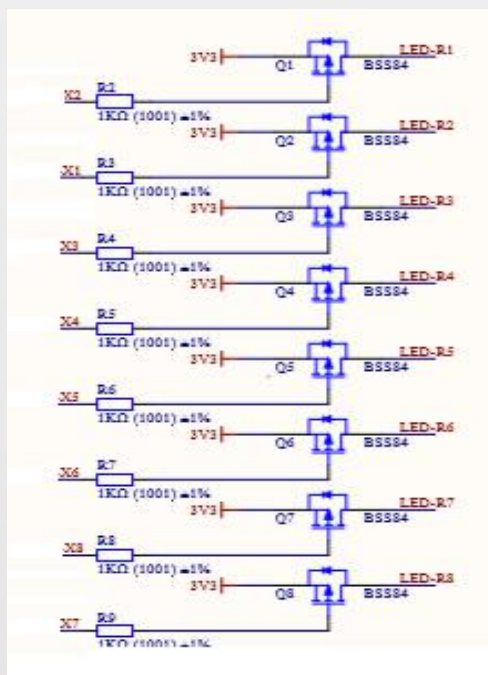
■ MOSFET管开关控制原理

- 三极管是电流驱动的，MOSFET是电压驱动的，三极管和MOS管开关电路原则
- **BJT三极管Transistors** 只要发射极e 对电源短路 就是电子开关用法
 - N管 发射极E 对电源负极短路. 低边开关, b-e 正向电流 饱和导通; 注, 底边-靠近GND侧
 - P管 发射极E 对电源正极短路. 高边开关, b-e 反向电流 饱和导通; 注, 高边-靠近VCC侧
- **FET场效应管MOSFET** 只要源极S 对电源短路 就是电子开关用法
 - N管 源极S 对电源负极短路. 低边开关, 栅-源 正向电压 导通
 - P管 源极S 对电源正极短路. 高边开关, 栅-源 反向电压 导通
- 三极管 b-e 必须有大于 C-E 饱和导通的电流
- 场效应管理论上栅-源有大于 漏-源导通条件的电压
- 右图为LED板局部, MOS管在高边
- 当输入信号为低电平时, 场效应管Q1处于导通状态,
- 输出电压为高电平, 对应LED行控制信号为高电平



LED点阵编程测试

- 实验所用的LED点阵屏是共阳极的
- 通过场效应管做开关来控制LED的行控制信号
- 通过X1-X8控制对应的MOS开关闭合
- Y1-Y8控制对应的列信号



```
from pyb import Pin
```

```
r=[Pin(i, Pin.OUT_PP) for i in ['X2','X1','X3','X4','X5','X6','X8','X7']]  
for i in range(8):  
    r[i].high()
```

```
c=[Pin(i, Pin.OUT_PP) for i in ['Y2','Y1','Y3','Y4','Y5','Y6','Y8','Y7']]  
for i in range(8):  
    c[i].high()
```

```
def ledDemo1(): #逐个点亮
```

```
    for i in range(8): #行循环  
        r[i].low() #MOS开关闭合  
        for j in range(8): #列循环  
            c[j].low() #点亮  
            pyb.delay(100) #延时  
            c[j].high() #熄灭  
        r[i].high() #MOS开关断开
```

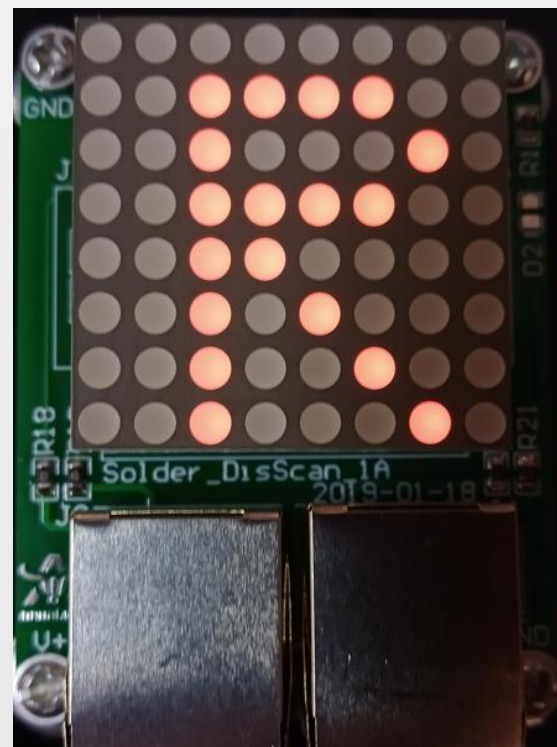
```
while True:  
    ledDemo1()
```

■ LED点阵显示字母R

- Demo为显示字母R
- 自己动手依次显示0-9
- 自己动手滚动显示0-9

```
#显示一行
def line(row,colDisp):
    r[row-1].low()
    for j in colDisp:
        c[j].low()
        pyb.delay(1)
        c[j].high()
    r[row-1].high()
```

```
#显示字母R
def ledDemo4():
    for k in range(300):
        line(1,[])
        line(2,[2,3,4,5])
        line(3,[2,6])
        line(4,[2,3,4,5])
        line(5,[2,3])
        line(6,[2,4])
        line(7,[2,5])
        line(8,[2,6])
```



LED点阵代码举例：0、A和滚动

```
def init():
    global Letter
    temp=Letter
    #更改数组内的值并赋给temp
    for i in range(len(temp)): # 2 loops 选择字母
        for j in range(8): # 8 loops 选择行
            for k in range(len(temp[i][1][j])): # 每行的每个点加8
                temp[i][1][j][k]=temp[i][1][j][k]+8*i
    #值更改完毕，开始左移，每显示一个，左移一个单位，即所有的数字减一
    #还需加上判断，小于0则灭，等于0为亮，大于7位灭，等于7位亮

def move():
    global Letter
    temp=Letter
    for i in range(len(temp)): # 2 loops 选择字母
        for j in range(8): # 8 loops 选择行
            for k in range(len(temp[i][1][j])):
                temp[i][1][j][k]=temp[i][1][j][k]-1
    #一个循环后，某行的所有的点向左移动一位
    #8个循环后，一个字母的所有点向左一位
    #show_LED(temp)
```

滚动

```
def show_A():
    for k in range(300):
        line(1,[3,4])
        line(2,[3,4])
        line(3,[2,5])
        line(4,[2,5])
        line(5,[1,2,3,4,5,6])
        line(6,[1,6])
        line(7,[0,7])
        line(8,[1])
```

显示字母A

```
def show_0():
    for k in range(300):
        line(1,[2,3,4])
        line(2,[1,5])
        line(3,[1,5])
        line(4,[1,5])
        line(5,[1,5])
        line(6,[1,5])
        line(7,[1,5])
        line(8,[2,3,4])
```

显示数字0

LED点阵代码举例：滚动I♥NEU

```
def init():#右移
    global Letter
    temp=Letter
    #更改数组内的值并赋给temp
    for i in range(len(temp)):# 2 loops 选择字母
        for j in range(8):#8 loops 选择行
            for k in range(len(temp[i][1][j])):#每行的每个点加8
                temp[i][1][j][k]=temp[i][1][j][k]+8*i
    #值更改完毕，开始左移，每显示一个，左移一个单位，即所有的数字减一
    #还需加上判断，小于0则灭，等于0为亮，大于7位灭，等于7位亮
def move():#每次左移一列
    global Letter
    temp=Letter
    for i in range(len(temp)):# 2 loops 选择字母
        for j in range(8):#8 loops 选择行
            for k in range(len(temp[i][1][j])):
                temp[i][1][j][k]=temp[i][1][j][k]-1
            #一个循环后，某行的所有的点向左移动一位
    #8个循环后，一个字母的所有点向左一位
```

LED点阵代码举例：猜拳

石头，剪刀，布的定义

```
png1=[ #剪刀
    "0*0000*0",
    "0*0000*0",
    "0*0000*0",
    "0*0000*0",
    "00*****0",
    "000***00",
    "000***00",
    "00*000*0"
]
png2=[ #石头
    "00000000",
    "00*****0",
    "00*****0",
    "00*****0",
    "00*****0",
    "00*****0",
    "000***00",
    "00000000"
]
png3=[ #布
    "00000000",
    "000*0000",
    "00***000",
    "00****00",
    "0*****0",
    "0*****0",
    "0*****00",
    "00000000"
]
```

结果图形的定义

```
png_0=[#0 平局
    "00****00",
    "0*0000*0",
    "0*0000*0",
    "0*0000*0",
    "0*0000*0",
    "0*0000*0",
    "0*0000*0",
    "00****00"
]
png_1=[#1 1号胜出
    "000**000",
    "000**000",
    "00***000",
    "000**000",
    "000**000",
    "000**000",
    "000**000",
    "000**000",
    "0*****0"
]
png_2=[#2 2号胜出
    "000**000",
    "0**00**0",
    "*000**00",
    "0000**00",
    "000**000",
    "000**000",
    "00**0000",
    "0*****"
]
```

胜负逻辑的判断

```
while True:
    show_png(png1)
    if u1.value()==0:
        for k in range(30):
            show_png(png1)
            i=1
    if u2.value()==0:
        for k in range(30):
            show_png(png1)
            j=1
    if i==1:
        for k in range(30):
            show_png(png_0)
    elif i==2:
        for k in range(30):
            show_png(png_1)
    else:
        for k in range(30):
            show_png(png_2)
```