

■ 认识RTC

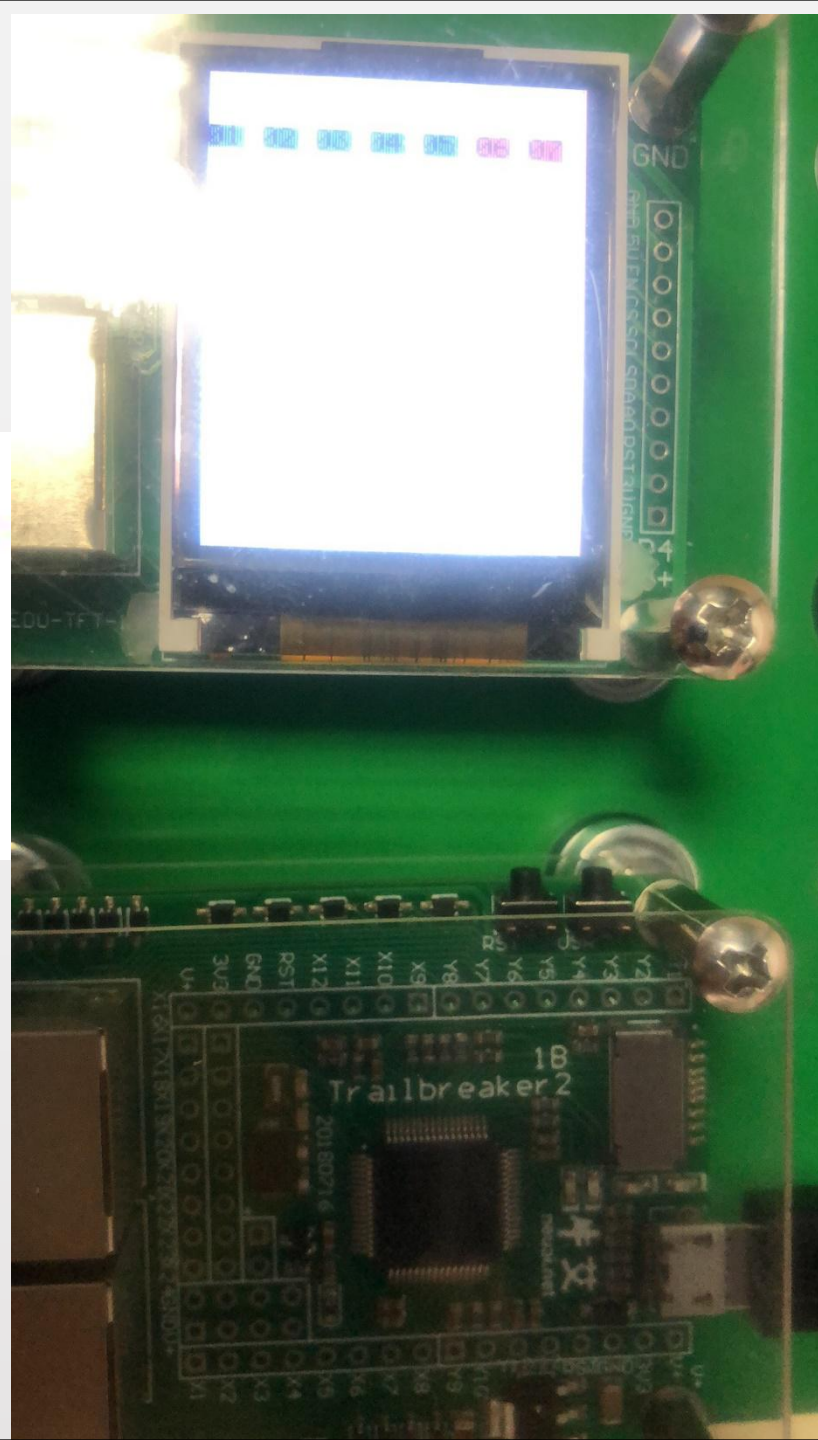
- RTC (Real-Time Clock) 即实时时钟
- PC主板上的晶振及相关电路组成的时钟电路的生成脉冲, RTC经过8254电路的变频产生一个频率较低一点的OS(系统)时钟TSC, 系统时钟每一个cpu周期加一, 每次系统时钟在系统初起时通过RTC初始化
- 提供稳定的时钟信号给后续电路用。主要功能有:时钟, 日历, 闹钟, 周期性中断输出, 32KHz时钟输出。

■ 认识RTC

- RTC类中有四个方法：
- `RTC.datetime([datetimeuple])`：设置RTC的日期与时间，其中`datetimeuple`的格式为：(year, month, day, weekday, hours, minutes, seconds, subseconds)
- `RTC.wakeup(timeout, callback=None)`：唤醒函数，第一个参数为时间长度，单位为微秒，第二个参数为函数入口
- `RTC.info()`：获取关于启动时间以及重置源的信息
- `RTC.calibration(cal)`：校准RTC

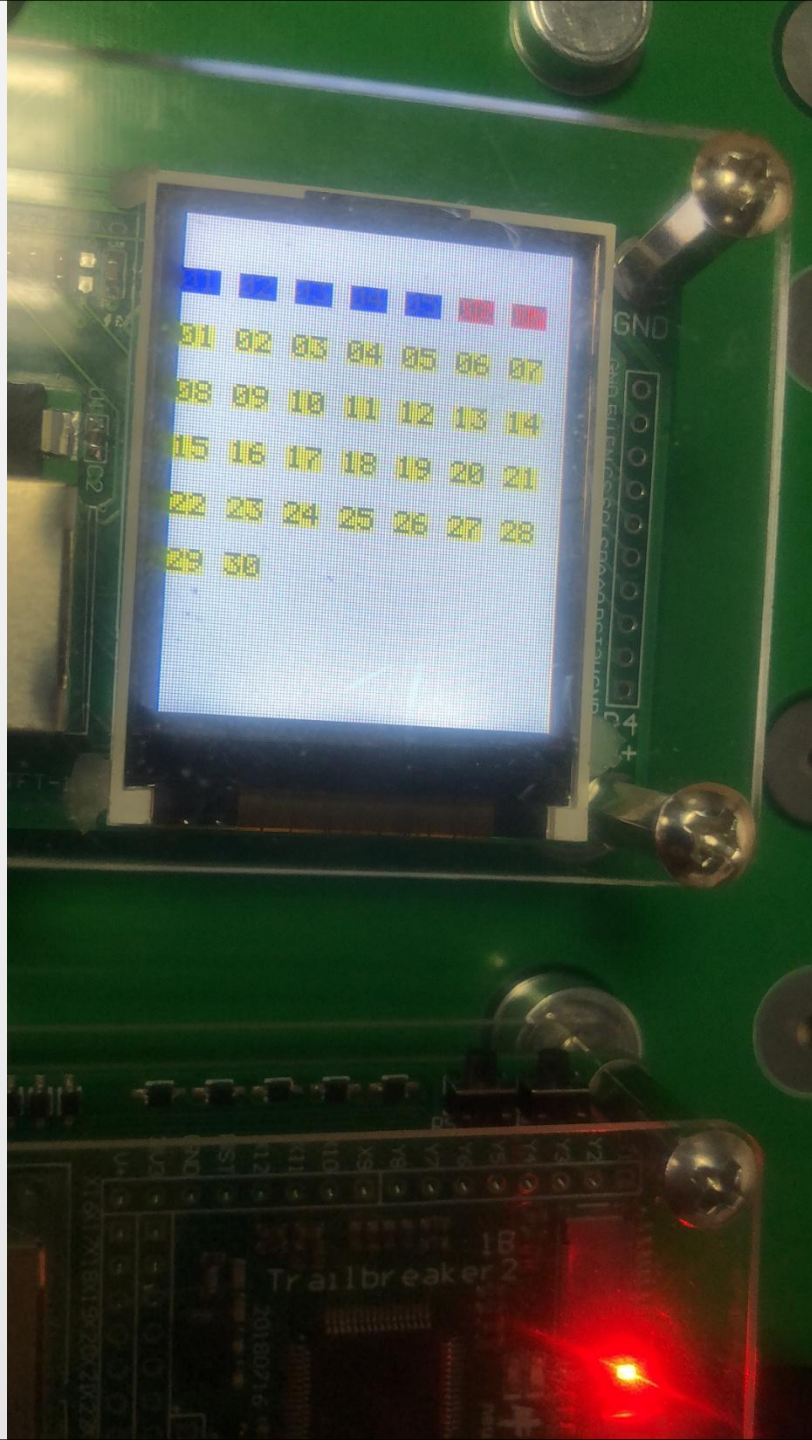
■ 日历界面设计

```
disp.clr(disp.WHITE) #清屏
disp.putstr_back(0,2,"01",disp.BLACK,disp.BLUE) #用蓝色显示周一到周五
disp.putstr_back(3,2,"02",disp.BLACK,disp.BLUE)
disp.putstr_back(6,2,"03",disp.BLACK,disp.BLUE)
disp.putstr_back(9,2,"04",disp.BLACK,disp.BLUE)
disp.putstr_back(12,2,"05",disp.BLACK,disp.BLUE)
disp.putstr_back(15,2,"06",disp.BLACK,disp.RED) #红色显示周六周日
disp.putstr_back(18,2,"07",disp.BLACK,disp.RED)
```



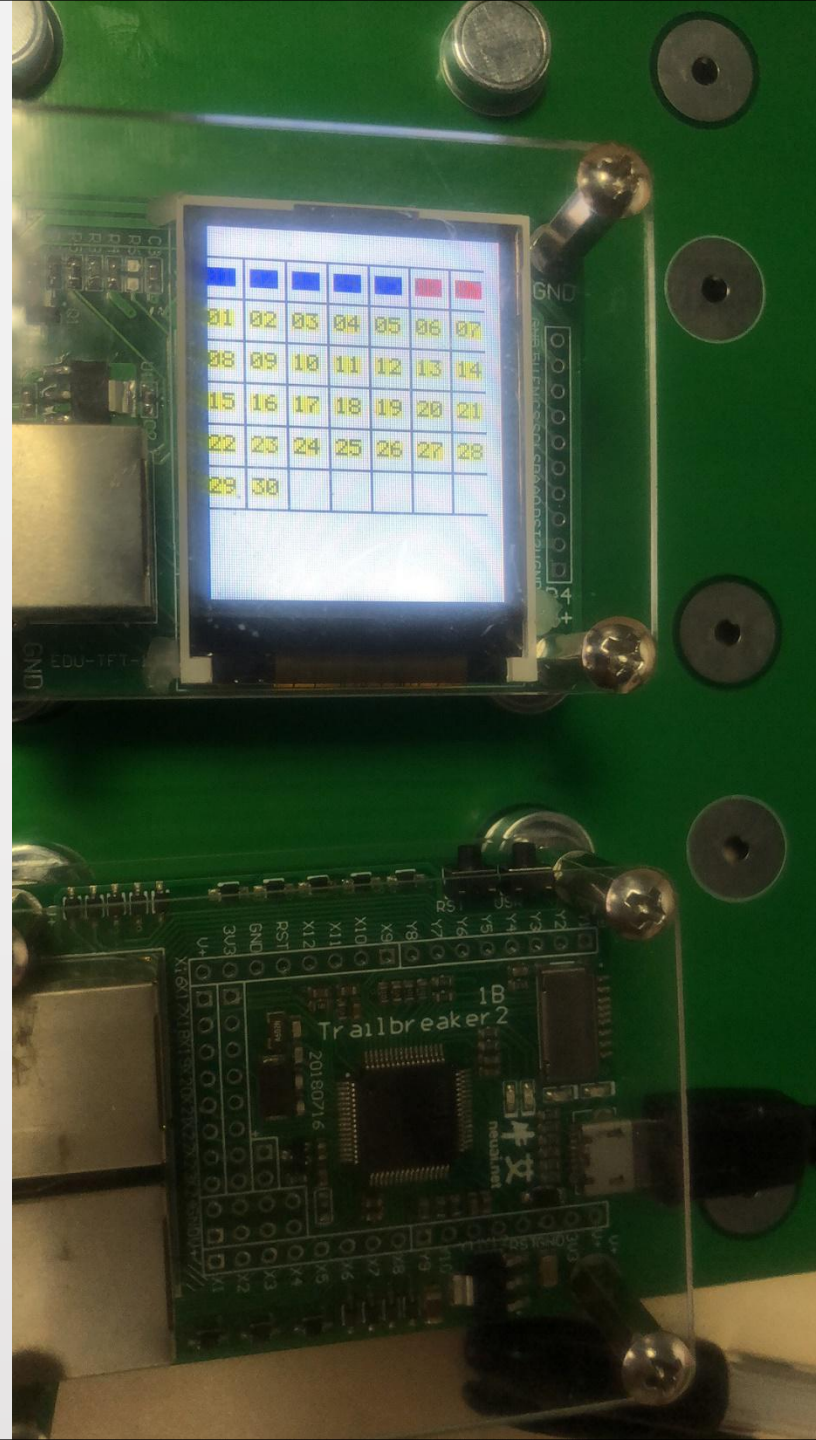
■ 日历界面设计

```
disp.putstr_back(0,4,"01",disp.BLACK,disp.YELLOW)#黄色显示日期 1-30
disp.putstr_back(3,4,"02",disp.BLACK,disp.YELLOW)
disp.putstr_back(6,4,"03",disp.BLACK,disp.YELLOW)
disp.putstr_back(9,4,"04",disp.BLACK,disp.YELLOW)
disp.putstr_back(12,4,"05",disp.BLACK,disp.YELLOW)
disp.putstr_back(15,4,"06",disp.BLACK,disp.YELLOW)
disp.putstr_back(18,4,"07",disp.BLACK,disp.YELLOW)
disp.putstr_back(0,6,"08",disp.BLACK,disp.YELLOW)
disp.putstr_back(3,6,"09",disp.BLACK,disp.YELLOW)
disp.putstr_back(6,6,"10",disp.BLACK,disp.YELLOW)
disp.putstr_back(9,6,"11",disp.BLACK,disp.YELLOW)
disp.putstr_back(12,6,"12",disp.BLACK,disp.YELLOW)
disp.putstr_back(15,6,"13",disp.BLACK,disp.YELLOW)
disp.putstr_back(18,6,"14",disp.BLACK,disp.YELLOW)
disp.putstr_back(0,8,"15",disp.BLACK,disp.YELLOW)
disp.putstr_back(3,8,"16",disp.BLACK,disp.YELLOW)
disp.putstr_back(6,8,"17",disp.BLACK,disp.YELLOW)
disp.putstr_back(9,8,"18",disp.BLACK,disp.YELLOW)
disp.putstr_back(12,8,"19",disp.BLACK,disp.YELLOW)
disp.putstr_back(15,8,"20",disp.BLACK,disp.YELLOW)
disp.putstr_back(18,8,"21",disp.BLACK,disp.YELLOW)
disp.putstr_back(0,10,"22",disp.BLACK,disp.YELLOW)
disp.putstr_back(3,10,"23",disp.BLACK,disp.YELLOW)
disp.putstr_back(6,10,"24",disp.BLACK,disp.YELLOW)
disp.putstr_back(9,10,"25",disp.BLACK,disp.YELLOW)
disp.putstr_back(12,10,"26",disp.BLACK,disp.YELLOW)
disp.putstr_back(15,10,"27",disp.BLACK,disp.YELLOW)
disp.putstr_back(18,10,"28",disp.BLACK,disp.YELLOW)
disp.putstr_back(0,12,"29",disp.BLACK,disp.YELLOW)
disp.putstr_back(3,12,"30",disp.BLACK,disp.YELLOW)
```



■ 日历界面设计

```
disp.put_hline(0,13,121,disp.BLACK) #黑色线画网格
disp.put_hline(0,31,121,disp.BLACK)
disp.put_hline(0,49,121,disp.BLACK)
disp.put_hline(0,67,121,disp.BLACK)
disp.put_hline(0,85,121,disp.BLACK)
disp.put_hline(0,103,121,disp.BLACK)
disp.put_hline(0,121,121,disp.BLACK)
disp.put_vline(15,13,108,disp.BLACK)
disp.put_vline(33,13,108,disp.BLACK)
disp.put_vline(51,13,108,disp.BLACK)
disp.put_vline(69,13,108,disp.BLACK)
disp.put_vline(87,13,108,disp.BLACK)
disp.put_vline(105,13,108,disp.BLACK)
```



■ 显示时间

- 通过RTC类的datetime（）方法对RTC进行时间的设定
- 实例化一个RTC对象，之后对齐进行设计，方法变量为一个数组，按照年、月、日、星期、小时、分、秒、亚秒（1/255秒）进行赋值

```
rtc = pyb.RTC()  
rtc.datetime((2014, 5, 1, 4, 13, 0, 0, 0))  
print(rtc.datetime())
```

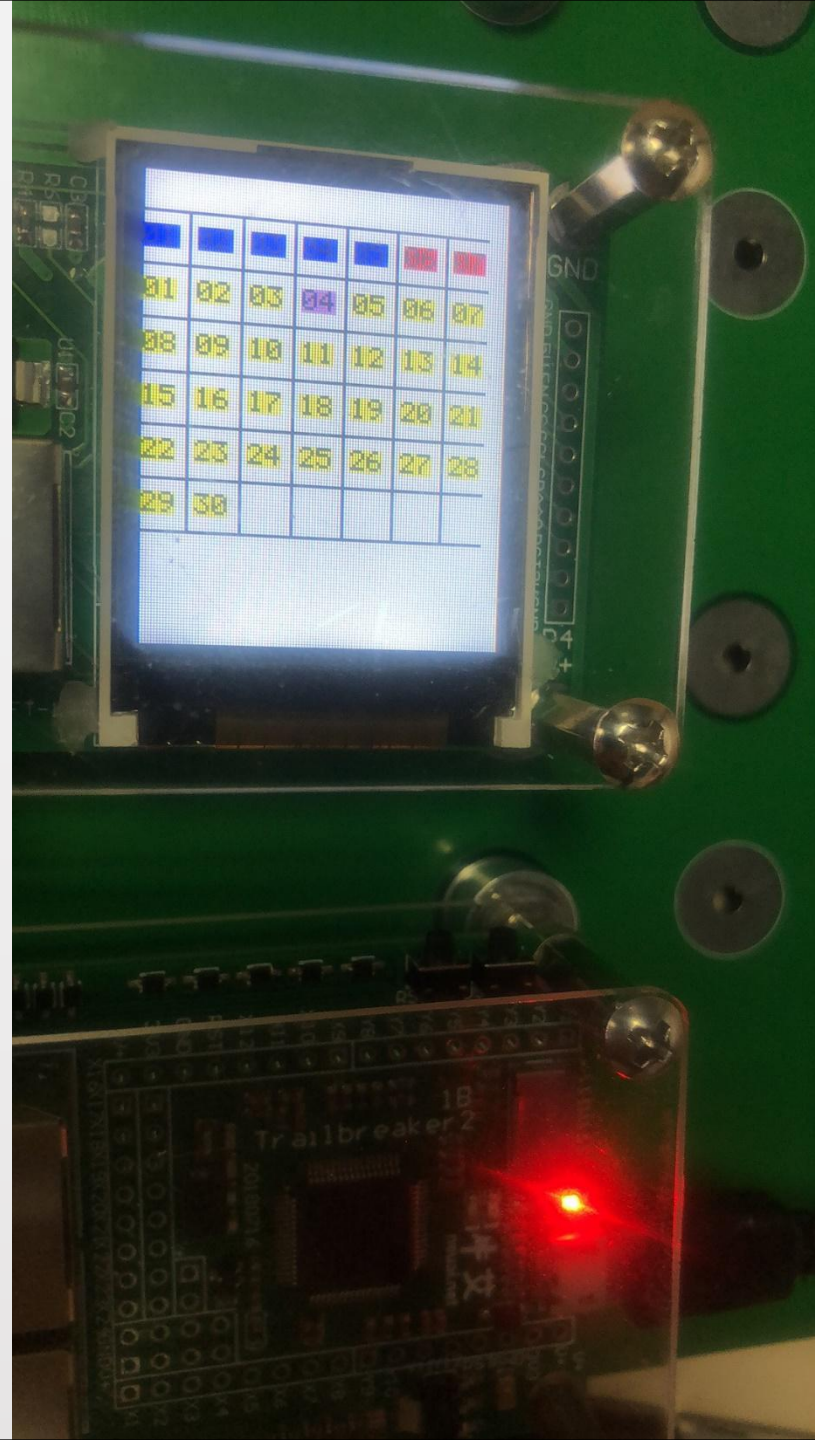
- 调用datetime（）方法查看RTC时间，该方法会返回一个长度为8的数组

```
>>> rtc.datetime()  
(2014, 5, 1, 4, 13, 3, 28, 142)  
>>> rtc.datetime()  
(2014, 5, 1, 4, 13, 3, 29, 82)  
>>> rtc.datetime()  
(2014, 5, 1, 4, 13, 3, 30, 192)  
>>> rtc.datetime()  
(2014, 5, 1, 4, 13, 3, 30, 83)  
>>> rtc.datetime()  
(2014, 5, 1, 4, 13, 3, 30, 83)
```

■ 显示时间

计算当前日期的行列，并用粉色标记出来
(先将原位置用白色写，再用粉色标记)

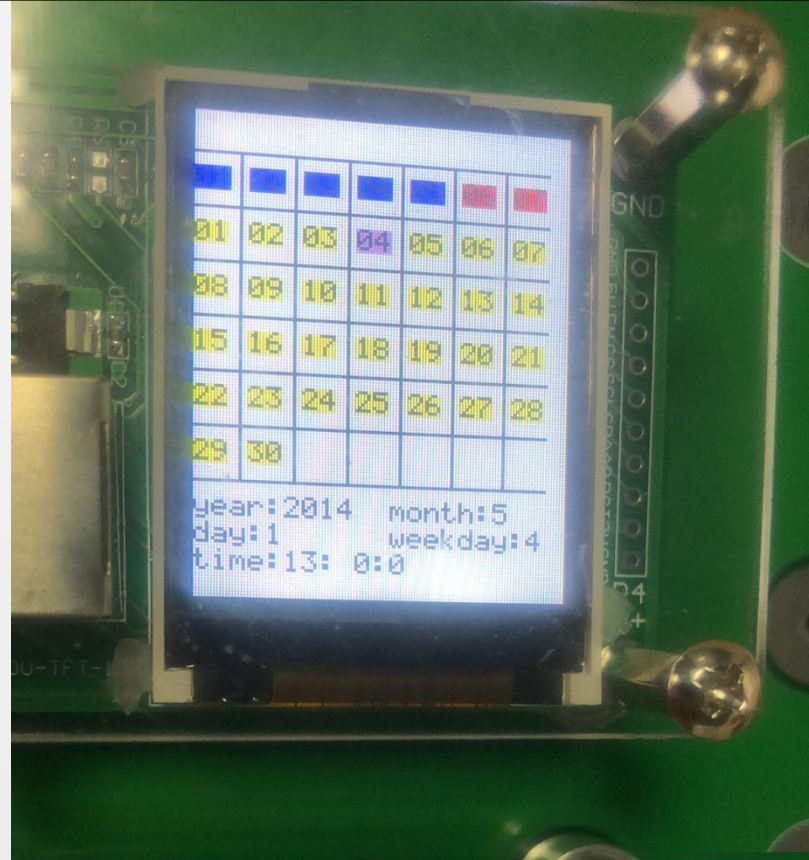
```
x=(a[3]-1)%7*3
y=((a[3]-1)//7+2)*2
if a[3]>=10:#将当前日期用粉色标记出来
    disp.putstr_back(x,y,str(a[3]),disp.WHITE,disp.WHITE)
    disp.putstr_back(x,y,str(a[3]),disp.BLACK,disp.PINK)
else:
    disp.putstr_back(x,y,'0'+str(a[3]),disp.WHITE,disp.WHITE)
    disp.putstr_back(x,y,'0'+str(a[3]),disp.BLACK,disp.PINK)
```



■ 显示时间

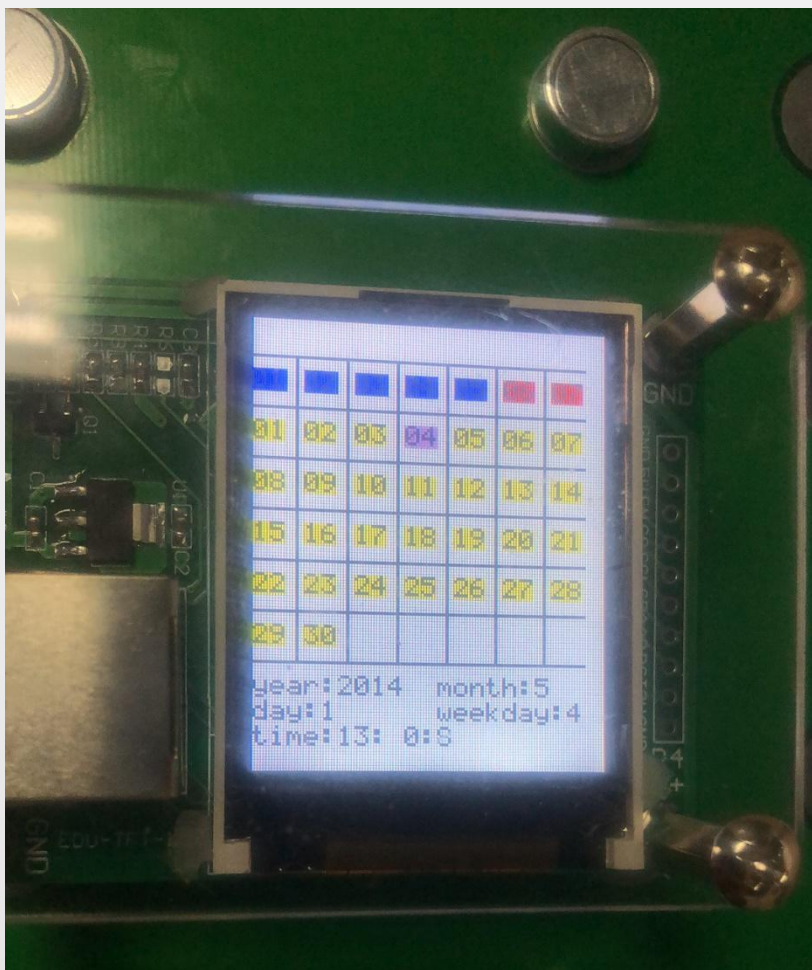
显示时间信息

```
s1='year: '+str(t1)
s2='month: '+str(t2)
s3='day: '+str(t3)
s4='weekday: '+str(t4)
if t6>=10:
    s5='time: '+str(t5)+' ':' '+str(t6)+' ':' '+str(t7)
else:
    s5='time: '+str(t5)+' ':' '+str(t6)+' ':' '+str(t7)
disp.putstr(0,14,s1,disp.BLACK) #显示"年"信息
disp.putstr(11,14,s2,disp.BLACK) #显示"月"信息
disp.putstr(0,15,s3,disp.BLACK) #显示"日"信息
disp.putstr(11,15,s4,disp.BLACK) #显示"星期"信息
disp.putstr(0,16,s5,disp.BLACK) #显示"时间"信息
```



■ 显示时间

更新时间变化信息：
若时间有变化，在变化时间的对应位置
用白色重写原信息，达到清除效果
再用黑色写入更新后的信息，达到更新效果



```
while True: # 显示时间变化
    print(rtc.datetime())
    a=rtc.datetime()

    if t1!=a[0]:
        disp.putstr(0,14,s1,disp.WHITE)
        t1=a[0]
        s1='year: '+str(a[0])
        disp.putstr(0,14,s1,disp.BLACK)
    if t2!=a[1]:
        disp.putstr(11,14,s2,disp.WHITE)
        t2=a[1]
        s2='month: '+str(a[1])
        disp.putstr(11,14,s2,disp.BLACK)
    if t3!=a[2]:
        disp.putstr(0,15,s3,disp.WHITE)
        t3=a[2]
        s3='day: '+str(a[2])
        disp.putstr(0,15,s3,disp.BLACK)
    if t4!=a[3]:
        disp.putstr(11,15,s4,disp.WHITE)
        t4=a[3]
        s4='weekday: '+str(a[3])
        disp.putstr(11,15,s4,disp.BLACK)
    if t5!=a[4]:
        if t5>=10:
            disp.putstr(5,16,str(t5),disp.WHITE)
        else:
            disp.putstr(5,16,' '+str(t5),disp.WHITE)
        t5=a[4]
        if t5>=10:
            disp.putstr(5,16,str(a[4]),disp.BLACK)
        else:
            disp.putstr(5,16,' '+str(a[4]),disp.BLACK)
    if t6!=a[5]:
        if t6>=10:
            disp.putstr(8,16,str(t6),disp.WHITE)
        else:
            disp.putstr(8,16,' '+str(t6),disp.WHITE)
        t6=a[5]
        if t6>=10:
            disp.putstr(8,16,str(a[5]),disp.BLACK)
        else:
            disp.putstr(8,16,' '+str(a[5]),disp.BLACK)
    if t7!=a[6]:
        disp.putstr(11,16,str(t7),disp.WHITE)
        t7=a[6]
        disp.putstr(11,16,str(a[6]),disp.BLACK)
```