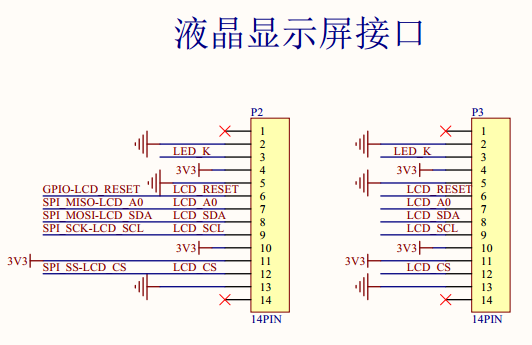
**使用pyboard驱动ST7735s液晶屏幕**

ST7735S液晶屏幕的分辨率为128\*160，显示色彩种类为65K,支持四线SPI接口，采用白色的背光灯。本测试程序包含三个文件，分别为main.py为主函数，lcd\_show.py为液晶的驱动程序，font.py为液晶显示ASCII的5\*7字库

首先设置IO（模拟SPI，液晶背光引脚，复位引脚），共6根需要设置。本程序将这六根引脚分成两个类进行定义。



然后开始编写时序驱动，通过查找手册确定时序，并编写相应函数

创建USR\_SPI类，内部定义“写时序”的函数功能。

USR\_SPI类包含以下几个函数：

对液晶控制器进行写操作函数如下

def write\_u8(self,data):

对液晶控制器写入command命令函数如下

def write\_cmd(self,cmd\_buf):

对液晶控制器写入data命令函数如下

def write\_data(self,buf):

接下来开始编写显示部分的程序

创建DISPLAY类，内部定义对液晶控制器的初始化以及对描点，显示字符等函数

DISPLAY类包含以下几个函数：

对液晶进行初始化函数如下：

首先片选cs使能，背光使能，对液晶进行软件复位，再通过指令0X11唤醒液晶控制芯片，配置Frame rate，power sequence，gamma sequence，RGB模式为RGB565等，然后写入命令0X29让液晶开始显示

def init(self):

清屏函数如下：

入口参数：clr\_color为RGB565格式，例如：0xf8为红色，0x07e0为绿色（以下涉及到颜色部分，均为此格式）

def clr(self,clr\_color):

画点函数如下：

入口参数:x,y起始坐标,屏幕左上角是0,0;屏幕右下角是127,159;color: 特定颜色

def putpixel(self,x,y,color):

画矩形函数如下：

入口参数:x,y起始坐标,屏幕左上角是0,0;屏幕右下角是127,159;x\_len和y\_len分别是

横向和纵向的长度;color: 特定颜色

def putrect(self,x,y,x\_len,y\_len,color):

\*\*\*查看代码，**对比以上三个函数就会发现共性，先写入命令0x2a,再写入数据X轴的起始点和终止点；写入命令0x2b,再写入数据Y轴的起始点和终止点；写入命令0x2c,再分别写入数据颜色的高八位和低八位。（其中你所设置的XY轴围成面积覆盖了多少个点，就将循环多少次写入数据颜色）**

**起点：一个cmd 两个data，终点一样**

显示字符函数如下：

入口参数:x,y起始坐标,屏幕左上角是0,0;屏幕右下角是127,159;ch是字符;color: 特定颜色

显示ASCII码，显示值为20H-7FH(若为其它值，则显示' '。

**代码分析**：首先将ch转化成整数型赋值给char,判断char是否在20H-7FH内，在范围内则正常显示，显示时调用font.py文件中的ASCII库，通过描点函数写到屏幕上。

def putchar(self,x,y,ch,color):

显示字符串函数如下：

入口参数:x,y起始行和列的坐标,屏幕左上角是0,0;屏幕右下角是20,16;str是字符串;color: 特定颜色

代码分析：此函数内部做了处理，防止输出的字符串重合显示，所以入口参数为行列的起始坐标

def putstr(self,x,y,str,color):

显示字符函数如下（带背景色）：

入口参数:x,y起始坐标,屏幕左上角是0,0;屏幕右下角是127,159;ch是字符;color: 特定颜色

color\_back,背景颜色

def putchar\_back(self,x,y,ch,color,color\_back):

显示字符串函数如下（带背景色）：

入口参数:x,y起始行和列的坐标,屏幕左上角是0,0;屏幕右下角是20,16;str是字符串;color: 特定颜色color\_back,背景颜色

def putstr\_back(self,x,y,str,color,color\_back):

画水平线函数如下：

入口参数:x,y起始坐标,屏幕左上角是0,0;屏幕右下角是127,159;len是长度;color: 特定颜色

def put\_hline(self,x,y,len,color):

画垂直线函数如下：

入口参数:x,y起始坐标,屏幕左上角是0,0;屏幕右下角是127,159;len是长度;color: 特定颜色

def put\_vline(self,x,y,len,color):

画圆函数如下:

入口参数:x,y为圆心坐标,屏幕左上角是0,0;屏幕右下角是127,159;r为圆的半径;color: 特定颜色

def put\_circle(self,x,y,r,color):

测试代码如下：

import lcd\_show

import pyb

from pyb import Pin

#LCD

usrspi=USR\_SPI(scl=Pin('X6',Pin.OUT\_PP),sda=Pin('X7',Pin.OUT),dc=Pin('X8', Pin.OUT))

disp=DISPLAY(usrspi,cs=Pin('X5',Pin.OUT),res=Pin('X4',Pin.OUT),led\_en=Pin('X3', Pin.OUT))

disp.clr(disp.PINK)

disp.put\_vline(8,5,40,disp.RED)

disp.put\_vline(15,5,60,disp.GREEN)

disp.put\_hline(40,120,50,disp.RED)

disp.put\_hline(40,130,80,disp.GREEN)

disp.putrect(100,100,10,25,0x0ff0)

disp.putstr(6,5," gu yue ",0xf0f0)

disp.putstr(6,6," gu yue ",0x0f0f)

disp.putstr\_back(6,7," gu yue ",0x0000,0xffff)

disp.putstr\_back(6,8," gu yue ",0x0000,0xffff)

disp.put\_circle(63,79,50,disp.BLUE)

现象如下：

