

256 色程序编写方法

256 色程序编写方法

本文内容适用于所有 256 色的BS系列总线型液晶显示器。

一 256 色颜色数据定义

256 色下，每个象素对应一个字节，格式是R3G3B2，也就是说，红色占3位D[7: 5]，绿色占3位D[4: 2]，蓝色2位D[1: 0]。具体格式如下：

字节数据		D7、D6、D5	D4、D3、D2	D1、D0
颜色灰度		R2、R1、R0	G2、G1、G0	B1、B0
基本颜色	最黑	000	000	00
	亮蓝	000	000	11
	亮绿	000	111	00
	亮青	000	111	11
	亮红	111	000	00
	亮紫	111	000	11
	亮黄	111	111	00
	亮白	111	111	11
	蓝色最黑	000	000	00
	蓝色较暗	000	000	01
灰度绿色	较亮	000	000	10
	最亮	000	000	11
	最黑	000	000	00
	较暗	000	001	00
灰度红色
	较亮	000	110	00
	最亮	000	111	00
	最黑	000	000	00
灰度	较暗	001	000	00

	较亮	110	000	00
	最亮	111	000	00

编写 256 色的液晶显示程序，可以分为两种类型：

一种是直接将 PC 机上的256 色图片送到画面上显示，这种方法可以显示颜色丰富的画面，但将占用较多的存储器空间；

另一种方法是将单色的图片或字库用彩色的方式显示出来，这种方式显示颜色比较单一，但是占用较少的存储器空间。

针对上述两种类型的应用，我司所有256 色总线型显示器都提供了单点写和 8 点写模式，第一种类型选用单点写模式，第二种类型选用8 点写模式。

在 8 点写模式下，需要先定义好前景色和背景色，硬件会自动做颜色扩展，1 扩展为前景色，0 扩展为背景色。例如定义前景色为红色（0xe0），定义背景色为蓝色（0x03），在8 点写模式下写入一字节的显示数据0x55，那么从当前（x，

256 色程序编写方法

y) 坐标开始, 连续8 个像素点的颜色将变成: (蓝色、红色、蓝色、红色、蓝色、红色、蓝色、红色)。8 点写模式下, 一次写入8 个点, 写屏速度快。

另外, 在 8 点写模式下, 还提供有背景色禁止位。在背景色禁止位有效下硬件在做颜色扩展时, 只扩展前景色, 背景色不变。使用这种功能, 很容易实现只变化前景色, 而背景色保持不变的显示效果。

下面分别详细讨论两种方式的编程方法。

二 直接显示 PC 机上的256 色图片

这种方式要借助软件 image2lcd, 将PC 机的BMP 文件(256 色)先转化为点阵数据, 每点对应一个字节, 然后将点阵数据送显示。文件夹中包含以下5个文件。

image2lcd.exe

picture.bmp

picture.c

color256.c

color256.hex

image2lcd.exe 是图片提取软件, 安装后使用。

Picture.bmp 是PC 机上使用的图片, 可以用“画图”等工具软件编辑, 一定要将其保存为256 色格式。

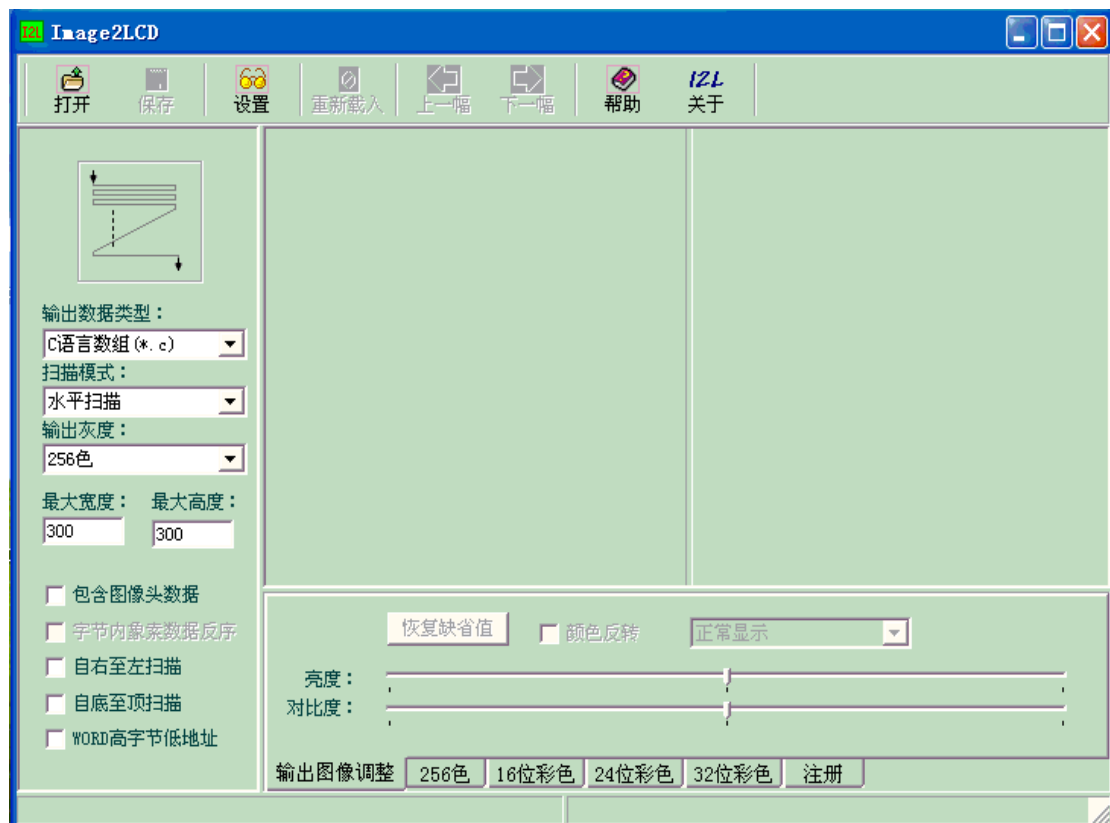
Picture.c 是用image2lcd 将图片picture.bmp 转换所得的点阵数据, 格式是C语言的数组。

Color256.c 是演示程序, 首先是清屏, 然后显示图片。

Color256.hex 是编译生成的十六进制文件。

Image2lcd 使用方法:

1 启动画面

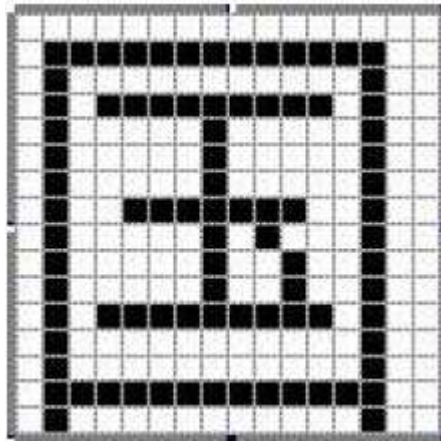


2 使用方法, 见下图中的1、2、3。

256 色程序编写方法



三 用 8 点写模式，将单色的字库或图片用彩色的方式显示
这种方法下，需要先设置好前景色和背景色，然后和操作单色 LCD 的方法一样，直接将 8 个点的数据（1 字节）写入 LCD 即可。
例如，“国”字的字模如下，16x16 点阵=2x16 字节。定义背景色为 0xff(白色)，前景色为 0xe0(红色)。



0x00,0x00

0x7f,0xfc

.....

0x40,0x04

例程：

```
#include <reg51.h>
```

```
#include <absacc.h>
```

```
#define X_ADDR XBYTE[0x0000]
```

256 色程序编写方法

```
#define Y_ADDR XBYTE[0x0100]
#define CMD XBYTE[0x0200]
#define DAT XBYTE[0x0300]
unsigned char code zk[32] = { //国
0x00,0x00,0x7F,0xFC,0x40,0x04,0x5F,0xF4,0x41,0x04,0x41,0x04,0x41,0x04,0x4F,0xE4,
0x41,0x44,0x41,0x24,0x41,0x24,0x5F,0xF4,0x40,0x04,0x40,0x04,0x7F,0xFC,0x40,0x04};
main()
{
unsigned int x,y;
unsigned int i;
unsigned char j,k,z,m,n;
////////////////////以下是测试8 点写功能////////////////////
//8 点写模式下清屏
CMD = 0x02;DAT=0xe0;//前景色
CMD = 0x04;DAT=0xff;//背景色
CMD = 0x01;
for( y=0;y<600;y++ )//用前景色清屏
{
X_ADDR = 0;
X_ADDR = 0;
Y_ADDR = y;
Y_ADDR = y/256;
for( x=0;x<100;x++ )DAT=0xff;
}
for(y=0;y<1;y++){x=1;while(x!=0)x++;}
for(y=0;y<1;y++){x=1;while(x!=0)x++;}
//8 点写模式下写汉字
CMD = 0x02;DAT=0xe0;//前景色
CMD = 0x04;DAT=0xff;//背景色
CMD = 0x01;
for ( n=0; n<=15; n++ )
{
X_ADDR = 0;
X_ADDR = 0;
Y_ADDR = n;
Y_ADDR = n/256;
DAT = zk[2*n];
DAT = zk[2*n+1];
}
////////////////////以上是测试8 点写功能////////////////////
while(1);
}
```

关于字库提取可参考《自制汉字库的原理与应用》《硬件汉字库的原理与应用》。