

串口液晶模块 SLCM全系列

(咨询 QQ: 56154083 电话: 13714759758 刘工)

淘宝样品邮购: <http://shop60210621.taobao.com/>

一、串口液晶概述

串口液晶是由深圳市炜烨兴电子有限公司开发生产的 SLCM 系列液晶显示模块。包括的点阵分别有：122*32，160*32，256*32，128*64，192*64，240*64，240*128 和 320*240 七种分辨率。随着用户的需求以后还会开发出更多其它分辨率！**同时接受批量的定制非标准显示屏** 现对所有能批量供货的串口液晶列一表格如下：

分辨率	布局	尺寸(PCB)	字库点阵	显示颜色	接口电平	工作电压	备注
122*32	分离方式	80*36	16*16	黄绿/蓝屏	UART	5V	外接按键
160*32	分离方式	116*35		蓝屏	TTL/232	5V 或 3.3V 3.3 要订货	可外接按键和 LED 灯
256*32	分离方式	129*33		蓝屏/灰屏			
128*64	独立方式	93*70 113*65 52*46	16*16 或 12*12	蓝屏或 黄绿	TTL/I0 串口	5V 或 3.3V	
	分离方式	54*50			TTL/232 I0 串口		
192*64	分离方式	130*65	16*16 或 12*12	蓝屏或 黄绿	TTL/232	5V 或 3.3V	
	独立方式				I0 串口		
240*64	独立方式	180*65	16*16 或 12*12	蓝屏或 黑白屏	TTL/232 I0 串口	5V 或 3.3V	
240*128	独立方式	144*104	16*16 或 12*12	蓝屏或 黄绿	TTL/232 I0 串口	5V 或 3.3V	
320*240	分离方式	160.9*109	16*16	蓝屏或 黑白屏	TTL/232 I0 串口	5V 或 3.3V	

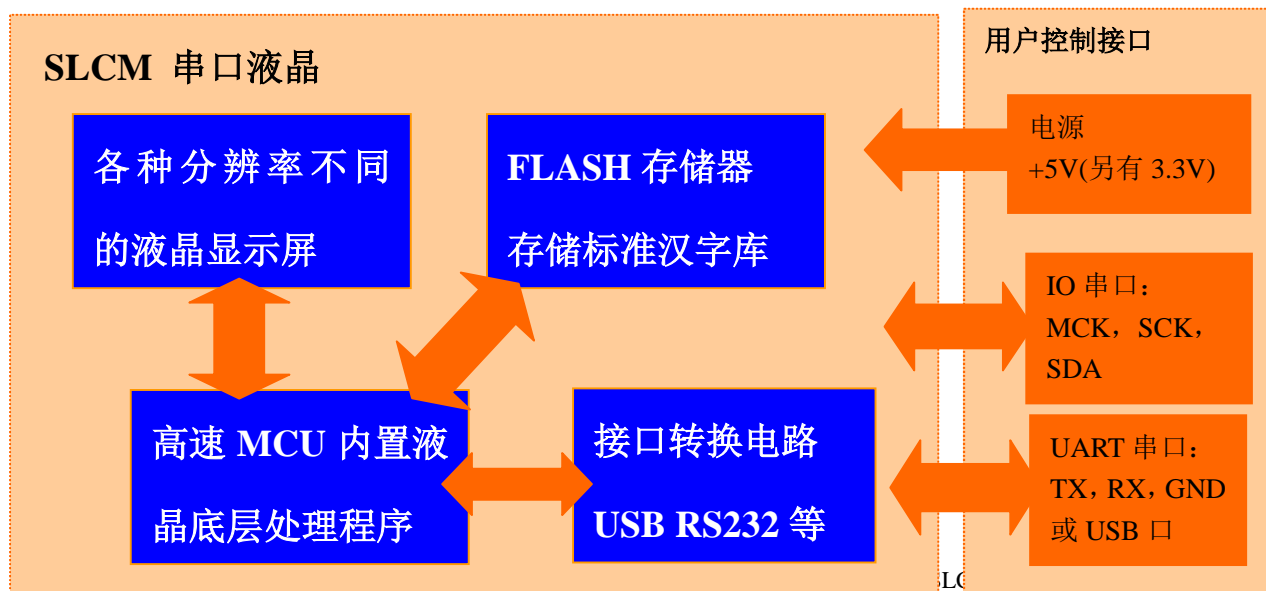
说明：

分离方式：整个显示屏驱动电路和串口控制电路分别有两块 PCB 板上，整体厚度增加 10mm。

独立方式：串口控制电路和液晶显示驱动电路整合在一块 PCB 上，结构紧凑，厚度不增加。

工作电压：单电源 5V 和 3.3V。**出厂的时候是 5V 的不能用 3.3V 供电，3.3V 的不能用 5V 供电。**

二、串口液晶结构



三、基本功能简介:

1. 驱动 128*64~320*240 点阵的液晶显示模块, 单电源供电, 分 5V 和 3.3V 两个版本
2. 国标 GB_2312 二级字库, 标准 ASCII 码, 编码调用
3. 可显示 16*16/32*32 点阵字体或 12*12/24*24 点阵字体
4. 两种接口形式: UART 串行接口(RS232 或 TTL 电平)和普通 IO 三线通信(TTL 电平)
5. 选购 USB 转串口控制板可在电脑上用串口助手调试(UART 串口方式)

四、电源及控制接口:

电源接口由 SLCM 板上的一白色两芯插座接入直流 5V 或 3.3V 电源, 电源正负极请参照 PCB 上的标识。控制接口由一白色三芯插座接入 (**122*32, 160*32, 256*32 两种分辨率没有普通 IO 接口**)。

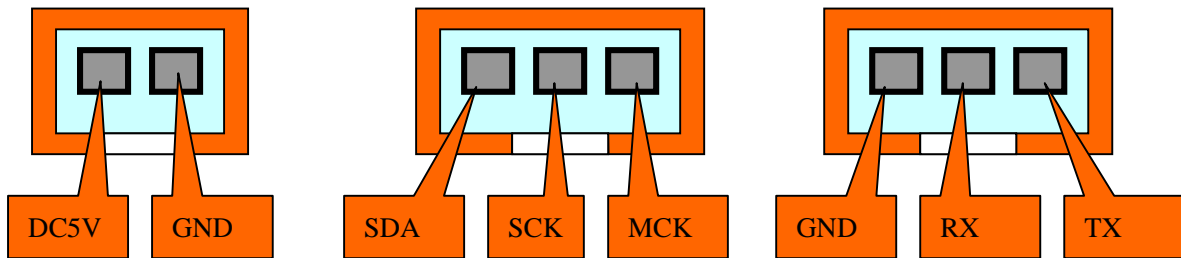
PWR (DC5V)	
+5V	0V

COM(UART 接口)		
GND	RX	TX
公共端	液 晶 接收端	液 晶 发送端

COM(普通 IO 接口)		
SDA	SCK	MCK
串行数据 出入	液晶握 手信号	用户握 手信号

电源及通信插座定义示意图: 正对着插座的缺口, 从左到右的顺序如下图所示。

我们提供的接线颜色无任何意义, 请不要根据线的颜色来先入为主地判定接线的功能定义!!!



五、SLCM 选购型号说明:

(一)	(二)	(三)	(四)	(五)	(六)
SLCM	xxxxxx	- AA	BB	C	DD
SLCM	240128A	- WB	02	A	S5
					___S/T=简体/繁体字库(目前只有简体)
					___5/3=5V 电压/3.3V 电压
					___A/B/C/D=接口类型: UART/USB/I0/RS232
					___01/02=12*12 点阵/16*16 点阵汉字库
					___W/B/Y=背光颜色: 白/蓝/黄
					___B/G/F=显示屏颜色: 蓝膜/黄绿/FSTN
					___显示屏分辨率及版本号: 12232/16032/12864/19264/24064/240128/320240

|___Wayeah LCM 串口液晶标志

例如: SLCM12864J3-WB01AS5

六、串口液晶控制指令

下表指令适用于 128*64,192*64,240*64,240*128,320*240 的屏上:

序号	控制命令	参数	功能说明
1	???	无	查询模块的状态,模块准备好就回传字符"OK"到主机(UART),模块收到此命令后,在显示屏最后一行显示波特率等提示
2	asd	英文串	在当前位置起显示一串英文字符,最多显示到行尾(地址不改变)
3	bis	a	开或关背光
4	brt	a	重新设置波特率(a=0-8),设置新的波特率后,主机也要重新设置 a=0 2400 a=1 4800 a=2 7200 a=3 9600 a=4 14400 a=5 19200 a=6 38400 a=7 57600 a=8 115200 ;UART/USB 接口有效 ;如果 a>8 则不做处理
5	chr	abc	显示内码为 ab 的字符,地址自动加 1.c=0 显示普通字,c=1 显示大字(对于英文字母 a=0)
6	cir	abcd	画圆.原点(a,b),半径 c.d=0 消圆, d=1 画圆
		abcde	画圆.原点(a,b,c),半径 d.e=0 消圆, e=1 画圆(320*240)
7	cIn	abcd	清除从点(a,b)到点(c,d)的一直线
		abcdef	清除从点(ab,c)到点(de,f)的一直线(320*240)
8	clr	无	清屏.清屏后,显示坐标也回复到起点(左上角)
9	clw	ablh	清除以(a,b)为起点的 l 个字符宽,h 个字符高的窗口内容
10	cpl	ablh	反显以(a,b)为起点的 l 个字符宽,h 个字符高的窗口内容
11	dat	abc	在坐标(a,b)处显示 c(主要用来输入图形或用户自定义文字数据) 128*64,192*64,240*64 均以点列,点行定位,数据竖排 240*128 以点列点行定位,数据横排 320*240 以字节列,点行定位,数据横排
12	def	ab+串	设置开机时在 a 列,b 行显示字符串 如果不需要开机显示,则发送: def+0
13	dot	abc	在(a,b)处显示一个点 c,c=0 则为清除点,c=1 则是画点
		abcd	在(ab,c)处显示一个点 d,d=0 则为清除点,d=1 则是画点
14	fhz	abcde	以点定位显示一个字符 a=00 正显小字 a=01 正显大字, a=10 为反显小字 a=11 反显大字 (b,c)为点坐标,de 为字符内码 (反显功能为 2014 年 8 月增加,8 月之前出货的无此功能)
15	lin	abcd	从点(a,b)到点(c,d)画一直线(屏分辨率 128*64-240*64)
		abcdef	从点(ab,c)到点(de,f)画一直线(屏分辨率 320*240)
16	loc	abc	设置(a,b)为显示坐标,c=1 则显示光标块,在进行 str,chr 命令前需先发送此命令
17	res	无	液晶复位.复位不影响显示内容! 出现花屏时可用此指令恢复
18	sad	a	选择显示区,a=0~7 共 8 个显示区,第 8 个显示区为文本区,可显示 40*16 个 5*8 点阵的英文字符。默认 0 区,每个显示区 240*128 个点 (此指令仅 T6963 控制的 240*128 有效) 12*12 点阵时只有 7 个区
19	str	a+串	显示字符串,a=0 显示正常字,a=1 显示大字;串长度<=100 字符(50 汉字). 注 1

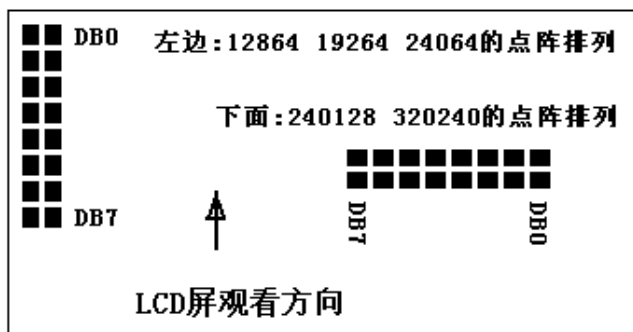
注 1: 大字体

如果当前为 6*12/12*12 点阵标准字形, 则大字符就是 12*24/24*24

如果当前为 8*16/16*16 点阵标准字形, 则大字符就是 16*32/32*32

注意事项:

1. RS232/UART 串行口模式设为模式 1 (1 个起始位, 8 个数据位, 1 个停止位);
2. RS232/UART 波特率设为 19200 (默认是 19200, 可通过命令修, 但断电后恢复为 19200)
3. 在发送命令之后需检测接收数据是否为 FFH, 否则所发的命令可能失效;
4. 串口发送的必须是完整命令的 16 进制格式;
5. 在每条命令末尾还要加上命令结束码“ODH”, 以表明该命令结束。
6. 液晶接收到一个正确的指令并处理完毕后, 会返回一个固定值: 0xff。
液晶处理指令需要一定时间, 所以请在接收到返回的 0xff 后再发下一条指令!
7. 请注意 122*32, 160*32, 256*32 和 320*240 点阵的屏无 12*12/24*24 点阵字库
8. 关于 dat 指令输入图形数据的说明:
12864 19264 24064 三种的 dat 数据排列是按点列字节行进行排列的, 数据高位在下, 低位在上; 而 240128 320240 的排列则是按点行字节列进行排列的, 数据的高位在左, 低位在右, 如下图:



USB 转 TTL 串口板 (需要单独购买)



下面的指令描述适合于 **128*64**, **192*64**, **240*64** 和 **240*128** 两种型号

 * 附: 指令及参数列表说明(128*64, 192*64, 240*64, 240*128, 320*240)
 *****/

序号	指令码	参数	功能说明及参数说明
01	s t r	a, Str	显示一串字符. a=0 显示小字, a=1 显示大字
02	a s d	Str	显示一串字符(仅 ASCII 码)
03	c h r	CHR, CHR_SS	显示一个字符. CHR_SS=0 显示小字, CHR_SS=1 显示大字. CHR 为双字节
04	f h z	CHR_SS, x, y, CHR	在 x, y 点行列显示一个字符. CHR_SS=0 显示小字, CHR_SS=1 显示大字
05	l o c	x, y, CSR	设定显示位置. CSR=0 关光标块, CSR=1 半角光标, CSR=2 全角光标
06	c p l	x, y, l, h	反显一个字符窗口. xy 为窗口起点列行, lh 为反显的长度和高度
07	c l w	x, y, l, h	清除一个字符窗口. xy 为窗口起点列行, lh 为清除的长度和高度
08	c l r	----	清除当前屏内容, 关光标块
09	d a t	x, y, Dat	在 xy 点列点行处显示 8 个点(12864 为竖向, 240128 为横向)
10	d o t	x, y, Dot	在 xy 点列点行处显示(Dot=1)/清除(Dot=0)1 个点
11	c i r	x, y, r, Cir	在 xy 点列行处显示(Cir=1)/清除(Cir=0)一个半径 r 的圆
12	l i n	xs, ys, xe, ye	在两点间画直线(xs, ys)-(xe, ye)
13	c l n	xs, ys, xe, ye	在两点间消直线(xs, ys)-(xe, ye)
14	b l s	----	开关背光
15	b r t	BRT	设置从机波特率. BRT=0-8
16	d e f	x, y, Def	设置开机时显示的串. xy 为显示位置, Def=0 则不显示
17	???	----	返回从机的波特率
18	s a d	SAD	设置 240128 的显示区. SAD=0-7, 7 区为文本区, 仅能显示 6*8 的字符
19	r e s	----	复位 LCD 显示(防止干扰或者别的原因造成 LCD 显示乱码或不显示)

七、SLCM 指令使用介绍:

指令共分为三大类: 控制类, 显示文字类, GUI 类(指令码均为小写字母)

一、控制类指令如下:

1、(???) 发送这个指令进去则会在显示屏上显示串口的波特率, 并回传字符” OK”

指令格式: 03FH 03FH 03FH 0DH

最后的 0DH 是指令的结束标志, 每一条指令均以它来做结束!

2、(bls) 液晶模块的背光开关。如果背光是开的发送此指令后则会关闭, 反之则打开。

指令格式: 062H 06CH 073H 0DH

指令有更新, 具体更新如下: 原指令后带的参数 00H 为打开背光, 01H 则为关背光

指令格式: 062H 06CH 073H 00H 0DH 打开背光

指令格式: 062H 06CH 073H 01H 0DH 关闭背光

3、(brt) 更改液晶串口的波特率。液晶的波特率分为 9 种分别如下:

2400、4800、7200、9600、14400、19200、38400、57600 115200

指令格式: 062H 072H 074H XX 0DH

其中的参数 XX 取值 0-8, 对应着上面的 9 个波特率值。

4、(loc) 设置显示字符字符的位置

指令格式: 06CH 06FH 063H XX1 XX2 XX3 0DH

此指令带三个参数:

XX1 显示列地址为显示屏的横方向坐标, 以 8 个点的递增, 写入字符后自动增加

XX2 显示行地址为显示屏的纵方向坐标, 以 16 个点的递增, 写入字符后自动增加

XX3 显示位置指示 (光标), 0 值不显示, 非 0 值显示

XX1 的最大值 = LCD 屏最大点列值 / 8 - 1

XX2 的最大值 = LCD 屏最大点行值 / 16 - 1

例如 240*128 的屏 XX1 和经 XX2 的值分别是 $240/8-1=29$ (1DH), $128/16-1=7$ 。就是能显示 8 行, 一行 30 个字符或 15 个汉字。其它显示文字类的位置定位也是根据此来计算。

5、(res) 显示屏复位, 不影响显示内容

指令格式: 072H 065H 073H 0DH

二、显示文字类指令

1、(asd) 显示英文串。从当前设置的位置开始一直显示到行尾为止, 超出部分不显示。

指令格式: 061H 073H 064H XX 0DH

参数 XX 为一串纯英文数字的字符, 如果带有中文则以空格显示。

2、(chr) 显示单个的文字。可以是中文也可以是英文数字

指令格式: 063H 068H 072H XX1 XX2 XX3 0DH

此指令带三个参数:

XX1 显示字符的高位 (英文时这个参数为 00H)

XX2 显示字符的低位

XX3 显示字符类型, 0 值为小字, 非 0 值为大字

3、(clw) 清除一个文本窗口

指令格式: 063H 06CH 077H XX1 XX2 XX3 XX4 0DH

带四个参数, 说明如下:

XX1 显示窗口起始列位置 XX1+XX3 的值不超过显示屏最大显示的字符列数

XX2 显示窗口起始行位置 XX2+XX4 的值不超过显示屏最大显示的字符行数

XX3 清除的窗口列方向字符个数

XX4 清除窗口的纵方向字符个数

4、(cpl) 反显一个文本窗口

指令格式: 063H 070H 06CH XX1 XX2 XX3 XX4 0DH

带四个参数, 说明如下:

XX1 显示窗口起始列位置 XX1+XX3 的值不超过显示屏最大显示的字符列数

XX2 显示窗口起始行位置 XX2+XX4 的值不超过显示屏最大显示的字符行数

XX3 清除的窗口列方向字符个数

XX4 清除窗口的纵方向字符个数

5、(def) 设置液晶模块上电时显示的一串字符

指令格式: 064H 065H 066H XX1 XX2 XX3 0DH

三个参数说明如下:

XX1 显示字符串起始列位置

XX2 显示字符串起始行位置

XX3 要显示的字符串

如果 XX1=0, 且 XX2=0DH, 则清除已经设置的字符串

6、(fhz) 以点来定位显示一个字符

指令格式: 066H 068H 07AH XX1 XX2 XX3 XX4 XX5 0DH

五个参数说明如下:

XX1 显示字符类型, 00 值为小字正显, 10 小字反显, 01 值为大字正显, 11 值为大字反显

XX2 列方向的点位置 $XX2 < \text{显示屏最大分辨率列数} - \text{字符所占的点列数}$

XX3 行方向的点位置 $XX3 < \text{显示屏最大分辨率行数} - \text{字符所占的点行数}$

XX4 显示字符的高位

XX5 显示字符的低位

这是一个唯一以显示点来定位的显示文字的指令!

7、(str) 显示一串字符

指令格式: 073H 074H 072H XX1 XX2 0DH

两个参数说明如下:

XX1 显示字符类型,

XX2 要显示的字符串

三、GUI 类指令 (注意: 在 320*240 的屏上列地址为双字节, 其它的分辨率为单字节)

1、(cir) 画或消一个圆

指令格式: 063H 069H 072H XX1 XX2 XX3 XX4 0DH

四个参数说明如下:

XX1 圆心列方向位置 点定位

XX2 圆心纵方向位置 点定位

XX3 圆的半径

XX4 0 值消圆, 非 0 值画圆

2、(cln) 消一条直线

指令格式: 063H 06CH 06EH XX1 XX2 XX3 XX4 0DH

四个参数说明如下:

XX1 直线起始列方向位置 点定位

XX2 直线起始纵方向位置 点定位

XX3 直线结束列方向位置 点定位

XX4 直线结束纵方向位置 点定位

3、(lin) 画一条直线

指令格式: 06CH 069H 06EH XX1 XX2 XX3 XX4 0DH

四个参数说明如下:

- XX1 直线起始列方向位置 点定位
- XX2 直线起始纵方向位置 点定位
- XX3 直线结束列方向位置 点定位
- XX4 直线结束纵方向位置 点定位

4、(dat) 写一个字节 (8 点)

指令格式: 064H 061H 074H XX1 XX2 XX3 0DH

三个参数说明如下:

- XX1 要写的 8 个点的开始列方向位置 点定位
- XX2 要写的 8 个点的开始纵方向位置 点定位
- XX3 要写的字节值

5、(dot) 画或消一个点

指令格式: 064H 06FH 074H XX1 XX2 XX3 0DH

三个参数说明如下:

- XX1 点的列方向位置 点定位
- XX2 点的纵方向位置 点定位
- XX3 0 值消点, 非 0 值画点

八、附上测试程序及三线 IO 串口通讯协议(Keil C51)

```
#include <reg51.h>
#include <intrins.h>
sbit MCLK=P1^0;
sbit SCLK =P1^1;
sbit SDA =P1^2;

unsigned char bdata DATA;
sbit BIT0=DATA^0;

// 发送一个数据到模块 通信双方均进行检测状态,所以发送数据之间不需要延时-自动匹配速度
send_byte(unsigned char ch) //三线 IO 通讯协议:发送一字节数据
{
    unsigned char i;
    if(!MCLK)return; //MCLK=0 时不发送数据直接返回,对此有可能会造成此次发送数据的丢失!
        //如果发送不成功,请自行处理数据重发
    MCLK=0; //申请通讯
    DATA=ch;
    for(i=4;i!=0;i--) // 一个周期传两个 bits !
    {
        while(SCLK); //等待从机响应
        SDA=BIT0; //放一个 bit 到 SDA
    }
}
```



```

    MCLK=1;
    DATA=DATA>>1; //准备下一个 bit
    while(!SCLK); //等待从机响应
    SDA=BIT0; //放一个 bit 到 SDA
    MCLK=0;
    DATA=DATA>>1; //准备下一个 bit
}
while(SCLK); //检测从机收完
MCLK=1;
while(!SCLK); //等待从机准备好
SDA=1;
}

//发送一串字符串,字符串以 00H 作为结束码
lcd_string(bit char_bit, unsigned char *pst)
{
    send_byte('s');
    send_byte('t');
    send_byte('r');
    send_byte(char_bit);
    while(*pst!=0)
        send_byte(*pst++);
    send_byte(0x0d); //串口发送结束标志
}

//显示直线 对于 SLCM320240,x1 x2 应定义为 unsigned int 数据类型
dis_line(unsigned char x1,unsigned char y1,unsigned char x2,unsigned char y2)
{
    send_byte('l');
    send_byte('i');
    send_byte('n');
    send_byte(x1);
    send_byte(y1);
    send_byte(x2);
    send_byte(y2);
    send_byte(0x0d);
}

//清除直线
del_line(unsigned char x1,unsigned char y1,unsigned char x2,unsigned char y2)
{
    send_byte('c');
    send_byte('l');
    send_byte('n');
    send_byte(x1);
    send_byte(y1);
    send_byte(x2);
    send_byte(y2);
}

```

```

    send_byte(0x0d);
}
clr_lcd() //清屏-清空所有显示 RAM 的内容,如果仅清空当前显示窗口的请使用 clw 命令
{
    send_byte('c');
    send_byte('l');
    send_byte('r');
    send_byte(0x0d);
}
////////////////////////////////////
main()
{
    unsigned int n;
    unsigned char i;
    for(n=0;n<1000;n++); //wait...
    while(1)
    {
        clr_lcd(); //清屏
        for(n=0;n<10000;n++);
        //查状态 建议只在调试的时候用
        send_byte('?');
        send_byte('?');
        send_byte('?');
        send_byte(0x0d);
        for(n=0;n<60000;n++);

        //显示一个串 1
        send_byte('l');
        send_byte('o');
        send_byte('c');
        send_byte(2);
        send_byte(0);
        send_byte(0x0d); //定位字符串起始地址
        lcd_string(0,"Wayeah LCM ABC123 中文 a 混 B 排 D 液晶显示器。");
        for(n=0;n<60000;n++);

        //显示一个串 2
        lcd_string(0,"三 0 线串口,UART");
        for(n=0;n<60000;n++);

        //显示一个串 3
        send_byte('l'); //没指定显示地址则从上次的结束地址开始
        send_byte('o');
        send_byte('c');
        send_byte(0);
        send_byte(1);
    }
}

```

```

send_byte(0x0d); //定位字符串起始地址
lcd_string(1,"深圳炜烨");
for(n=0;n<60000;n++);
for(n=0;n<60000;n++);
//显示一线段
dis_line(0,10,117,33);
for(n=0;n<60000;n++);

//清除一线段
del_line(0,10,117,33);
for(n=0;n<60000;n++);

//反显一个区域:0 列开始反显 12 个字符位置
//          1 行开始反显两行
//注意:
send_byte('c');
send_byte('p');
send_byte('l');
send_byte(0);
send_byte(1);
send_byte(12);
send_byte(2);
send_byte(0x0d);
for(n=0;n<60000;n++);

//清除一个区域:1 列开始反显 12 个字符位置
//          1 行开始清除两行
send_byte('c');
send_byte('l');
send_byte('w'); //clw 指令
send_byte(1); //开始清除的坐标
send_byte(1);
send_byte(12); //清除的宽度-12 个字符位置
send_byte(2); //清除的高度-两行
send_byte(0x0d);
for(n=0;n<60000;n++);
}
}

```

以下为使用 UART 串口通信的设置:

//串口初始化 晶振为 11.0592M 方式 1 波特率 19200, THTL=253

```
void InitCom(unsigned char BaudRate)
```

```

{
    unsigned char THTL=253;
    SCON = 0x50; //串口方式 1,允许接收
    TMOD = 0x20; //定时器 1 定时方式 2

```

```

TCON = 0x40; //设定定时器 1 开始计数
TH1 = THTL;
TL1 = THTL;
PCON = 0x80; //波特率加倍控制,SMOD 位
RI = 0; //清收发标志
TI = 0;
TR1 = 1; //启动定时器
}
// UART 串口输出一个字节（非中断方式）
// 每发送完一个指令串,请等待 30~80ms 或者是读取串口返回数据=0FFH 再进行下一次指令的发送!
void ComOutChar(unsigned char OutData) //用此代替 send_byte(unsigned char ch)可使用 UART 串口
{
    SBUF = OutData;
    while(!TI); //空语句判断字符是否发完
    TI = 0; //清 TI
}

```

显示实例：

1、显示文字：首先定义了一个显示文字串的子程序如下

```

lcd_string(bit char_bit, unsigned char *pst)
{
    send_byte('s');
    send_byte('t');
    send_byte('r');
    send_byte(char_bit);
    while(*pst!=0)
        send_byte(*pst++);
    send_byte(0x0d); //串口发送结束标志
}

```

然后就可以使用这个子程序来显示一串字符了。例如要显示“我来自北京”则可以这样操作：

Lcd_string(0,“我来自北京”); 在显示屏上就会显示“我来自北京”

当然在使用此子程序前还需要使用 loc 来设置显示的起始位置,如 loc(1,2,0)来定义到屏上的第三行的第二列。

以下为一些显示图片：

240*64 点阵液晶显示 12*12/ 16*16 16*32 点阵文字



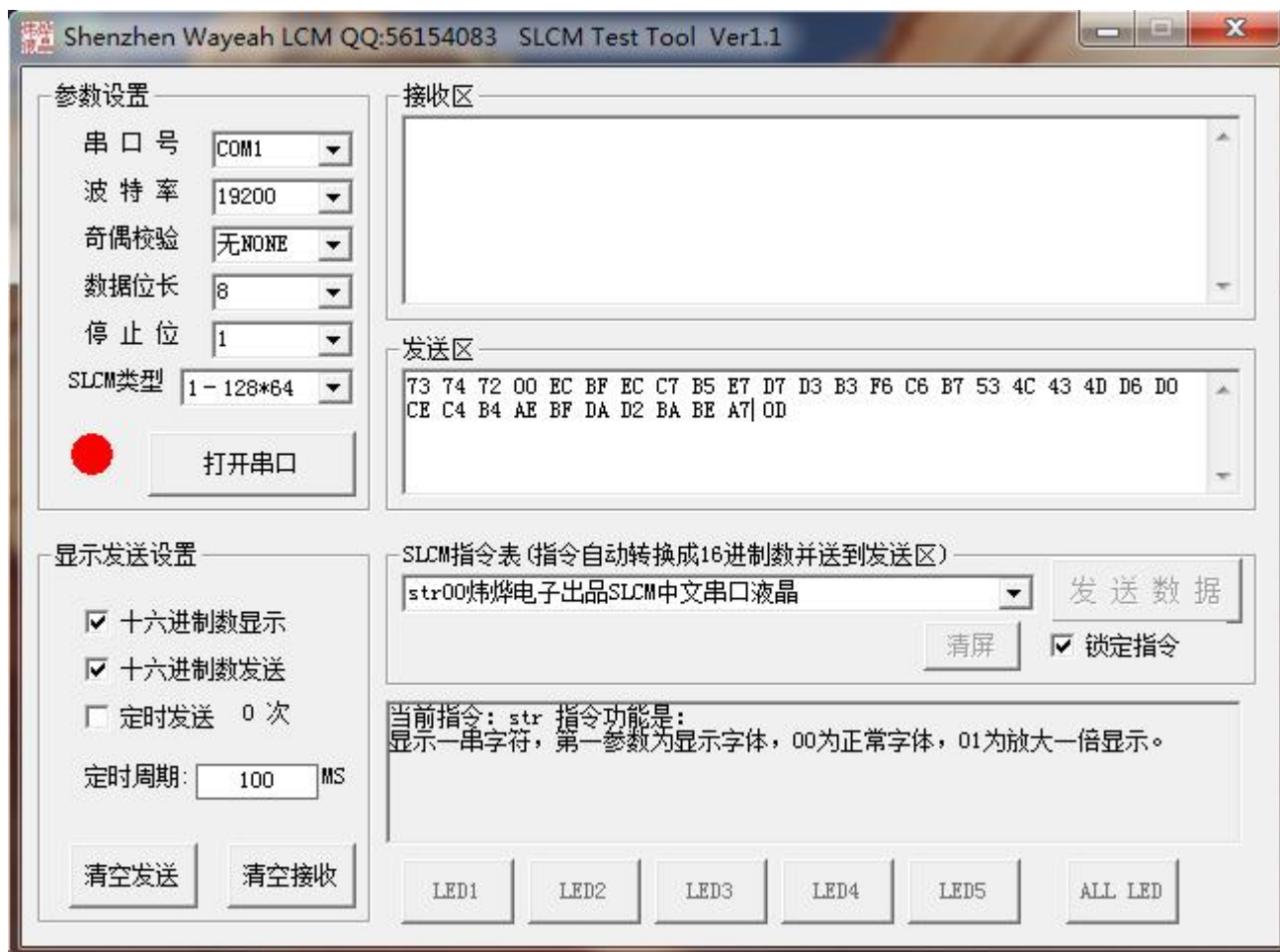
串口液晶的尺寸图:

(请参考另外的 PDF 文件)

串口液晶调试软件使用方法: (串口液晶调试工具.exe)

软件已经配置好了串口液晶的全部指令，点击相应的指令后马上转换成 16 进制数并推送到数据发送区，只要接上串口线，然后液晶上电，点击“发送数据”按钮就可以发送相应指令了。见下图

刚打开的时候是这个样子的: (软件如有更新，界面可能会有差别。更新后不一定主动通知用户)



操作方法:

- 一、选择你电脑上真正的串口号，别的用默认值
- 二、SLCM 类型对应有三个选项，分别对应到三种串口液晶的指令集。
 - 1-128*64 对应着：128*64、192*64、240*64、240*128 四种分辨率
 - 2-320*240 对应着：320*240 一种分辨率
 - 3-160*32 对应着：122*32、160*32、256*32 三种分辨率

三、然后点“打开串口”按钮。

然后在 SLCM 指令表下面选择一条指令，在中间的发送区就会出现对应的 16 进制编码了，再点“发送数据”按钮，发送区的数据就发送到串口液晶上，串口液晶就会根据指令码作出反应！

选择了一条指令后，在下方会显示对该指令的简要说明，也可以在在选择框里修改指令参数！

“发送数据”按钮下面的勾选项是对指令选择框里的内容进行锁定，只有以指令码开头的输入才会被推送到发送区。去掉勾选后就是一个纯粹的字符串转换十六进制格式功能

串口调试就是这么简单！

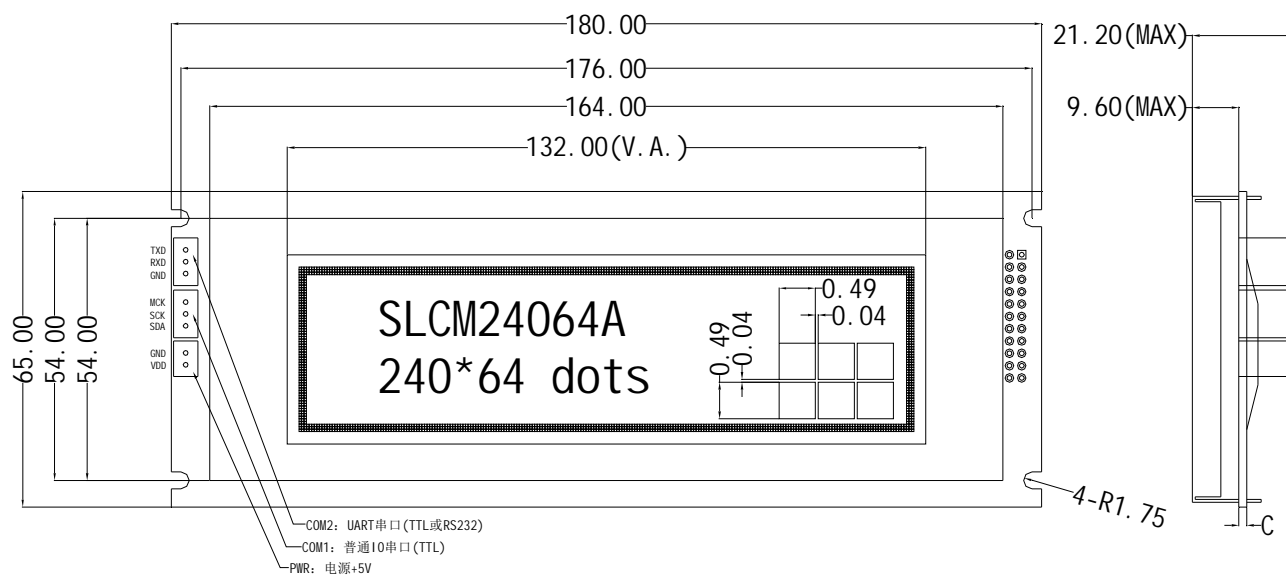
输入说明:

指令的参数输入请以 16 进制的形式输入，例如要在第 2 行的第 10 个位置进行显示，则输入：“loc0A0100”。loc 用来指定显示坐标，10 的十六进制数是 0A，第二行是 01（第一行是 00），最后的 00 则是选择显示光标块的开关。（关于指令的详细说明请参考本文件上面的表格）

各种串口液晶的安装尺寸图：（单位均为 mm）

SLCM12864D（最小尺寸：54*50）

SLCM24064A (尺寸为: 180*65)



本文结束