

# 字库串口液晶模块

FOR 128\*64 ~ 320\*240 点阵

(咨询 QQ: 56154083 电话: 13714759758 刘工)

## 功能简介:

1. 驱动 128\*64~320\*240 点阵的液晶显示模块, 单 5V 供电, LED 背光电源外接;
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
4. RS232 电平/TTL 电平串口或者直接 USB 接口(默认为普通 IO 串口)
5. 可在电脑上用串口助手调试(RS232 电平状态下)

电源及控制接口: (两种接口在出厂时已设定为其中一种用户不可自设定)

### PWR (DC5V)

+	-
+5V	0V

### COM(UART 接口)

1	2	3
GND	RX	TX

### PWR (DC5V)

+	-
+5V	0V

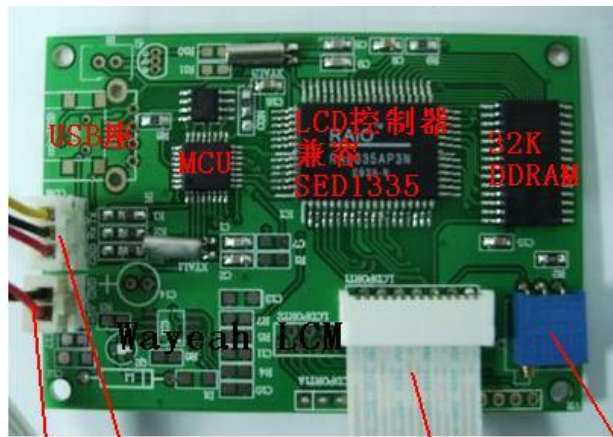
### COM(普通 IO 接口)

1	2	3
SAD	SCK	MCK

说明: UART 接口和普通 IO 串口在控制板上是通过三个 0 欧电阻跳选的, 在产品出厂时只可选择其中一种使用, 所以在你决定使用前需要预先通知我们。

见下图说明:

SLCM1335 串行接口及电源接口(独立控制板)



上: 三线串口 IO  
下: 电源 (5V)

LCD 接口  
(FFC, PITCH=1.25)

电位器  
调对比度

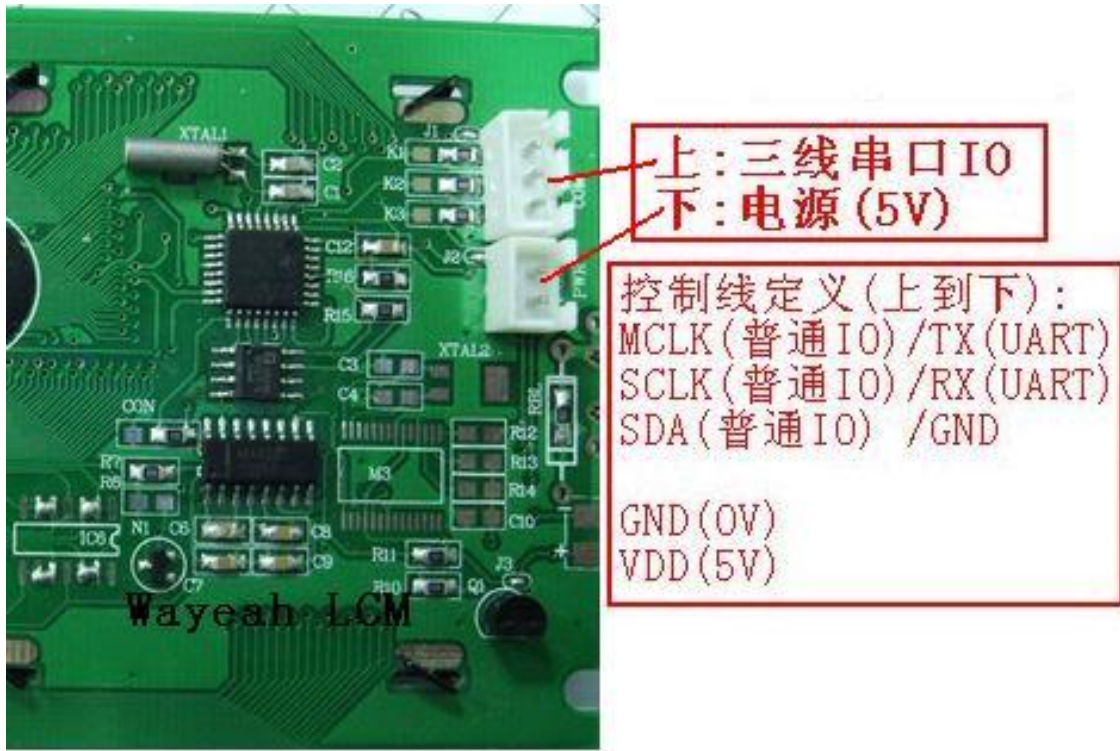
控制线定义(上到下):  
MCLK(普通 IO)/TX(UART)  
SCLK(普通 IO)/RX(UART)  
SDA(普通 IO)/GND

GND(0V)  
VDD(5V)

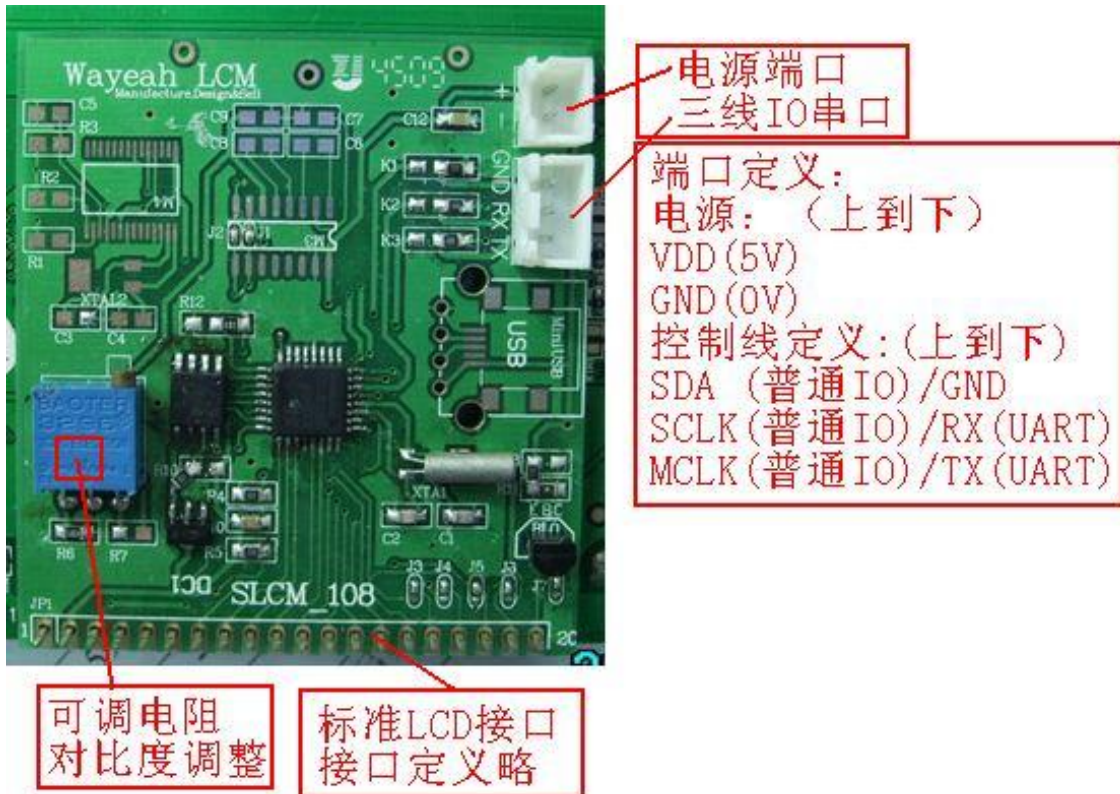
14PIN LCD 接口:(右到左)

1 DO	8 CL1
2 D1	9 CL2
3 D2	10 VDD
4 D3	11 VSS
5 DISOFF	12 VEE
6 YD	13 VO
7 FRM	14 NC

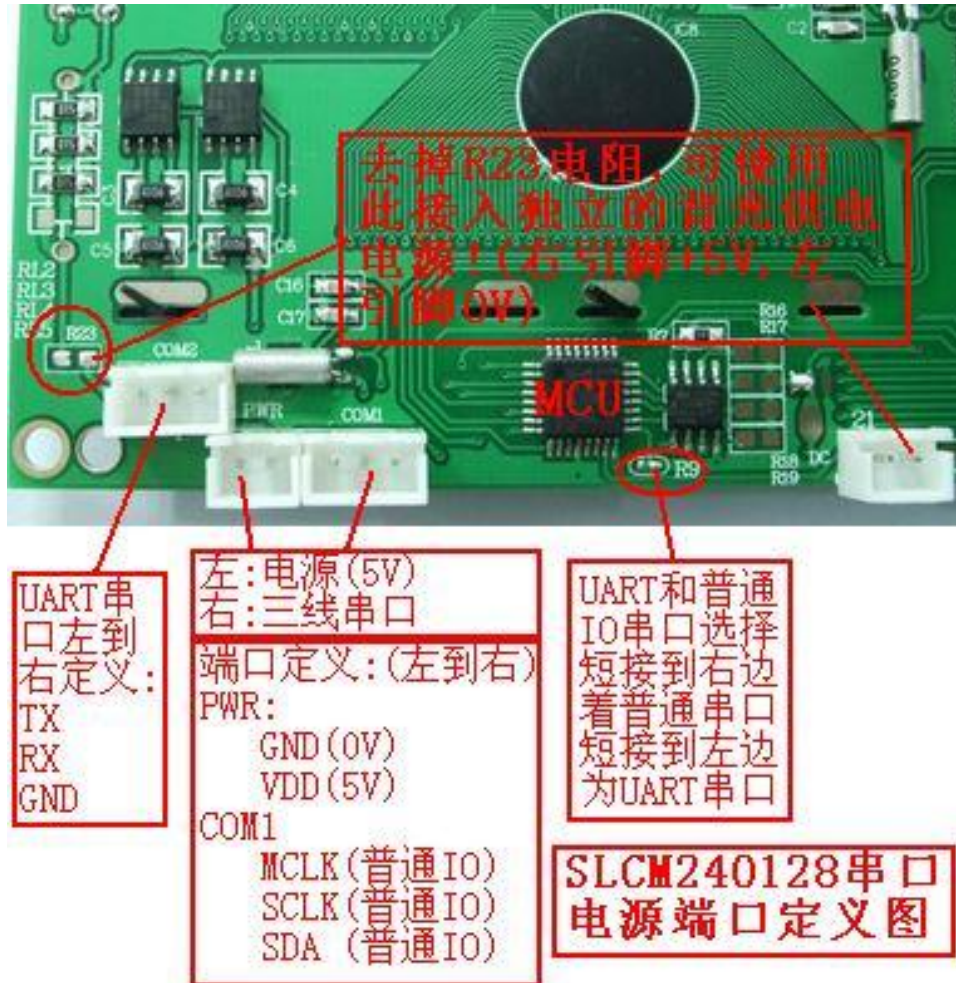
SLCM24064 的串行接口及电源接口(液晶和控制一体板):



SLCM12864 和 SLCM19264 的串行接口及电源接口(独立控制板)



SLCM240128 串行接口及电源定义图(液晶和控制一体):



### SLCM 选购型号说明:

SLCM xxxxx AA BB C DD

SLCM-串口液晶标志

xxxxxx-5~6 位数字, 表示液晶的分辨率 12864/19264/24064/240128/320240

AA-两位字母表示屏和背光颜色 蓝屏 B 黄绿屏 G 黑屏 F /白光 W 蓝光 B 黄光 Y

BB-两位数字表示字库点阵 01 表示 12\*12/24\*24 点阵, 02 表示 16\*16/32\*32 点阵

C-接口方式 A-UART B-USB C-三IO

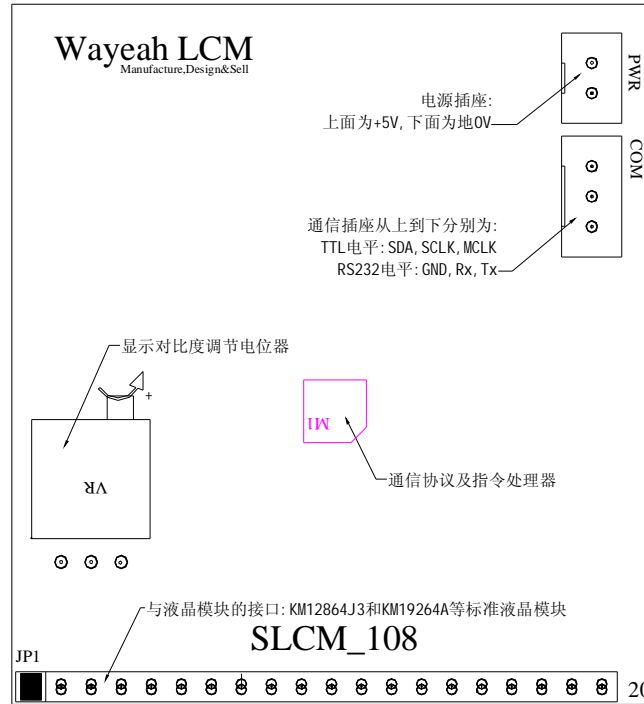
DD-简繁体字库工作电压: 简体 S 繁体 T 3V: 3.0~3.6 5: 5V

例: SLCM320240BW02CS5 表示 320\*240 分辨率蓝屏 16\*16/32\*32 三线 IO 接口简体 5V

## SLCM 的电源及控制接口:

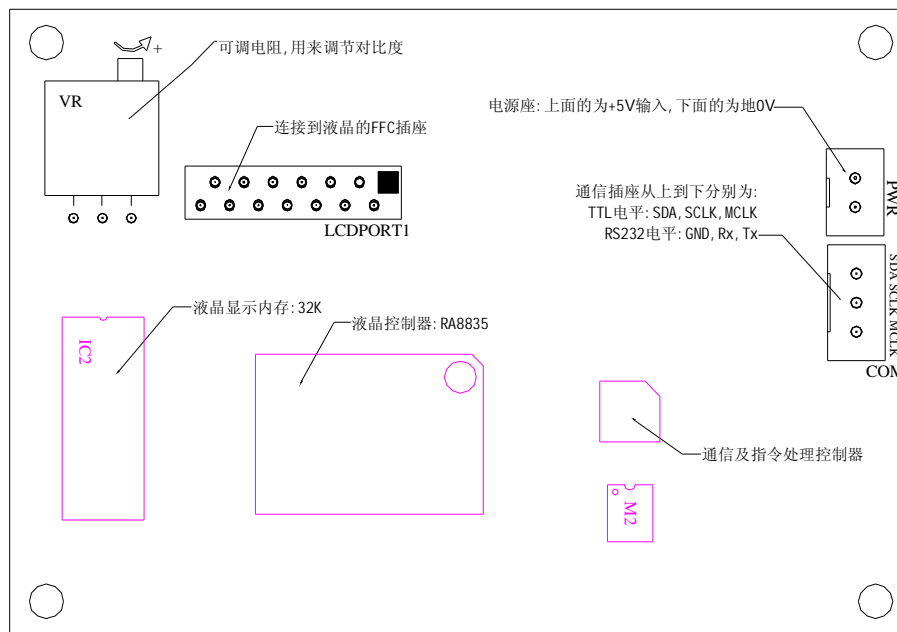
电源接口由 SLCM 板上的一白色两芯插座接入直流 5V 电源, 电源正负极请参照 PCB 上的标识。控制接口由一白色三芯插座接入。

## 128\*64 和 192\*64 点阵液晶控制板接口说明:



串口液晶128\*64及192\*64控制电路

## 320\*240 和 240\*128 点阵液晶控制板(控制器为 RA8835)接口说明:



串口液晶240\*128及320\*240控制电路

## SLCM 指令:

表 1 SLCM 命令说明: (下表命令在某些分辨率下可能无效, 请注意后面的说明)

控制命令	参数	功能说明
???	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
bls	无	开或关背光
brt	a	重新设置波特率(a=1-8),设置新的波特率后,主机也要重新设置 a=1 300      a=2 600 a=3 1200     a=4 2400 a=5 4800     a=6 9600 ;UART/USB 接口有效 a=7 19200    a=8 57600 ;如果 a>8 则不做处理
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)
cir	abcde	画圆.原点(ab,c),半径 d.e=0 消圆, e=1 画圆
cln	abcd	清除从点(a,b)到点(c,d)的一直线
clr	无	清屏.清屏后,显示坐标也回复到起点(左上角)
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 清除以(a,b)为起点的 l 个字符宽,h 点高的窗口内容(适于 240128 320240)
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 反显以(a,b)为起点的 l 个字符宽,h 点高的窗口内容(适于 240128 320240)
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)
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
lin	abcd	从点(a,b)到点(c,d)画一直线
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
sad	a	选择显示区,a=0~6 共 7 个显示区,默认 0(仅 T6963 控制的 240*128 有效)。每个显示区 240*128 个点
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),字符串以'\0'结束。

\*\*\*显示大字符表示

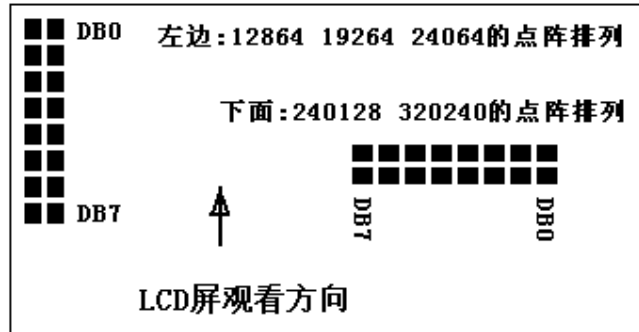
如果当前为 6\*12/12\*12 点阵标准字形,则大字符就是 12\*24/24\*24

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, 可通过命令修改)
3. 在发送命令之前需检测模块为闲态 (TX=0), 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“回车键”的 ASCII 码“0D(H)”, 以表明该命令结束。
6. 如果不想显示开机画面, 则在 100ms 内对 MCLK (TX), SCLK, SDA, 三个端口任何一个进行置 0 即可。
7. 在选择 12\*12/24\*24 点阵时, 无画圆功能, 请注意! 240\*128 和 320\*240 点阵的屏暂无 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:



## 附上测试程序及三线 IO 串口通讯协议

```

#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
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--)
    {
        while(SCLK); //检测从机响应
        SDA=BIT0; //放一个 bit 到 SDA
        MCLK=1;
        DATA=DATA>>1; //准备下一个 bit
        while(!SCLK); //检测从机响应
    }
}

```

```

        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;
}

//显示字符串
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(0x00); //字符串结束标志\0
    send_byte(0x0d); //串口发送结束标志
}

//显示直线
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(!MCLK); //MCLK=0 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++);

        clr_lcd(); //清屏
        for(i=0;i<96;i++) //显示所有的 ascii 字符
        {
            send_byte('c');
            send_byte('h');
            send_byte('r');
            send_byte(0x00);
            send_byte(i+0x20);
            send_byte(0x00); //小字体
            send_byte(0x0d);
        }
        for(n=0;n<60000;n++);

        clr_lcd();

        //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
// send_byte('1'); //不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(2);
// send_byte(2);
// send_byte(0x0d); //定位字符串起始地址
lcd_string(0,"三串口线,UART 串口,带 USB 转换!");
for(n=0;n<60000;n++);

//显示一个串 2
// send_byte('1'); //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(10);
// send_byte(3);
// send_byte(0x0d); //定位字符串起始地址
lcd_string(0,"三 0 线串口,UART");
for(n=0;n<60000;n++);

clr_lcd();
//显示一个串 3
send_byte('1'); //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++);

//close backlight
// send_byte('b');

```

```

//      send_byte('l');
//      send_byte('s');
//      send_byte(0x0d);
//      for(n=0;n<30000;n++);

//open backlight
//      send_byte('b');
//      send_byte('l');
//      send_byte('s');
//      send_byte(0x0d);
//      for(n=0;n<30000;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 300—57600

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

```
{
    unsigned char THTL;
    switch (BaudRate)
    {
```

```

    case 1: THTL = 64; break; //波特率 300
    case 2: THTL = 160; break; //600
    case 3: THTL = 208; break; //1200
    case 4: THTL = 232; break; //2400
    case 5: THTL = 244; break; //4800
    case 6: THTL = 250; break; //9600
    case 7: THTL = 253; break; //19200
    case 8: THTL = 255; break; //57600
    default: 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）
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
}

```

以下为一些显示图片：

128\*64 和 192\*64 点阵液晶显示 12\*12/24\*24 字体





128\*64 点阵液晶显示 16\*16/32\*32 点阵文字



240\*128 液晶显示 16\*16/32\*32 点阵字体





320\*240 上显示 16\*16/32\*32 点阵文字和显示圆

