

JLX12864OLED-13001-PN

中文使用说明书

目 录

序号	内 容 标 题	页码
1	概述	2
2	特点	2
3	外形及接口引脚功能	3~4
4	基本原理	4
5	技术参数	5
6	时序特性	5~6
7	指令功能及硬件接口与编程案例	7~页末

1. 概述

晶联讯电子专注于 OLED 屏及液晶模块的研发、制造。所生产 JLX12864OLED-13001-PN 型 OLED 模块由于使用方便、显示清晰，广泛应用于各种人机交流面板。

JLX12864OLED-13001-PN 可以显示 128 列*64 行点阵单色图片，或显示 16*16 点阵的汉字 8 个*4 行，或显示 8*16 点阵的英文、数字、符号 16 个*4 行。或显示 5*8 点阵的英文、数字、符号 21 个*8 行。

2. JLX12864OLED-13001-PN 图像型点阵 OLED 模块的特性

2.1 结构牢：焊接式 FPC。

2.2 IC 采用 SH1106, 功能强大，稳定性好

2.3 功耗低。

2.4 显示内容：

- 128*64 点阵单色图片；

- 可选用 16*16 点阵或其他点阵的图片来自编汉字，按照 16*16 点阵汉字来计算可显示 8 字/行*4 行。

2.5 指令功能强：可组合成各种输入、显示、移位方式以满足不同的要求；

2.6 接口简单方便：采用 4 线 SPI 串行接口。

2.7 工作温度宽：-20℃ - 70℃；

2.8 储存温度宽：-30℃ - 80℃；



3. 外形尺寸及接口引脚功能

3.1 外形图

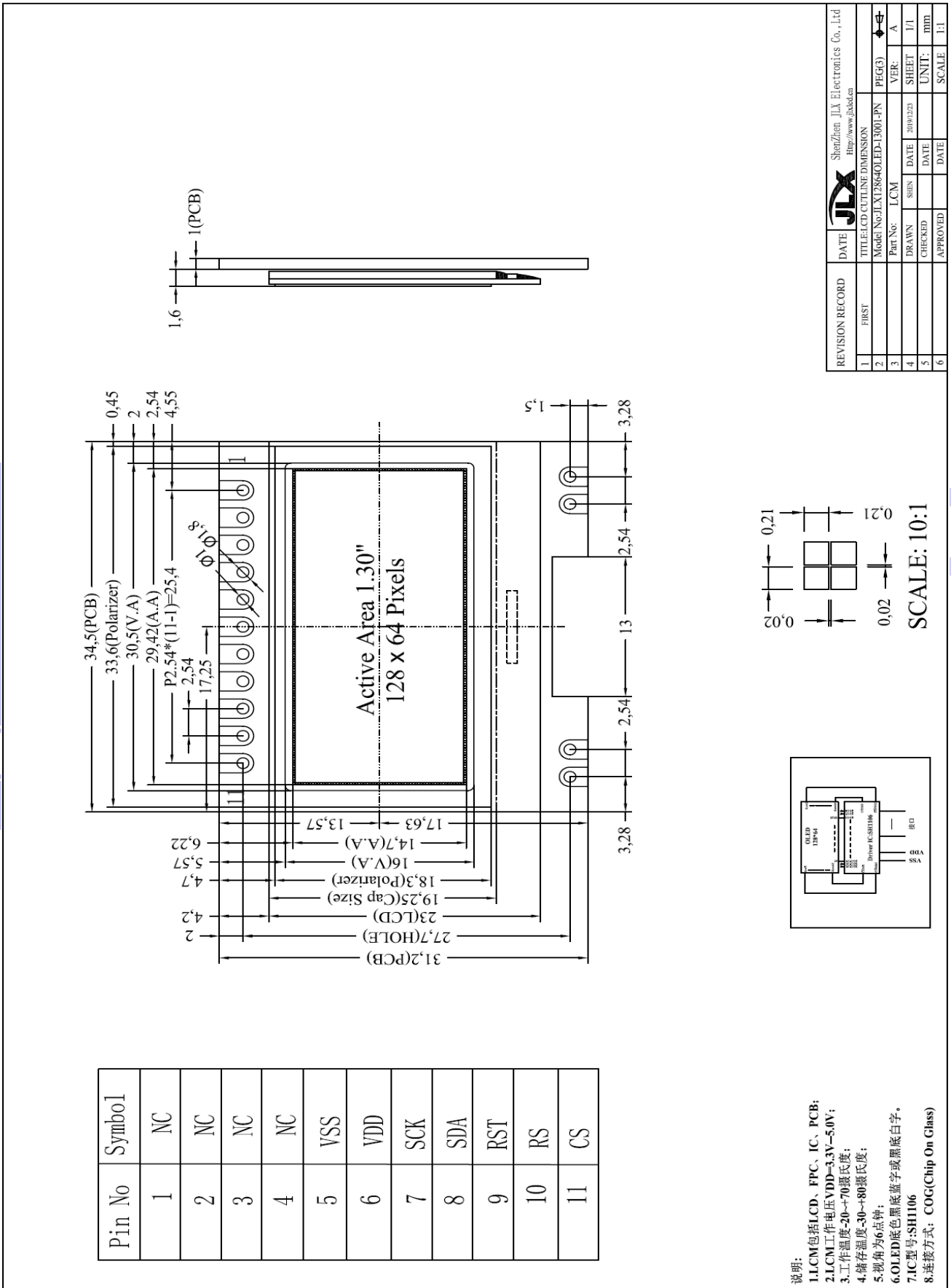


图 1. OLED 模块外形尺寸

模块的接口引脚功能

引线号	符号	名称	功能
1	NC	NC	
2	NC	NC	
3	NC	NC	
4	NC	NC	
5	VSS	接地	0V
6	VDD	电源电路	2.7V~5.0V
7	SCK	I/O	串行时钟
8	SDA	I/O	串行数据
9	RST	复位	低电平复位，复位完成后，回到高电平，OLED 模块开始工作
10	RS	寄存选择信号	H: 数据存储器 0: 指令存储 (IC 资料上缩写为“A0”)
11	CS	片选	低电平片选

表 1：模块的接口引脚功能

4. 基本原理

4.1 OLED 屏 (OLED)

在 OLED 上排列着 128×64 点阵, 128 个列信号与驱动 IC 相连, 64 个行信号也与驱动 IC 相连, IC 邦定在 LCD 玻璃上 (这种加工工艺叫 COG)。

电路框图

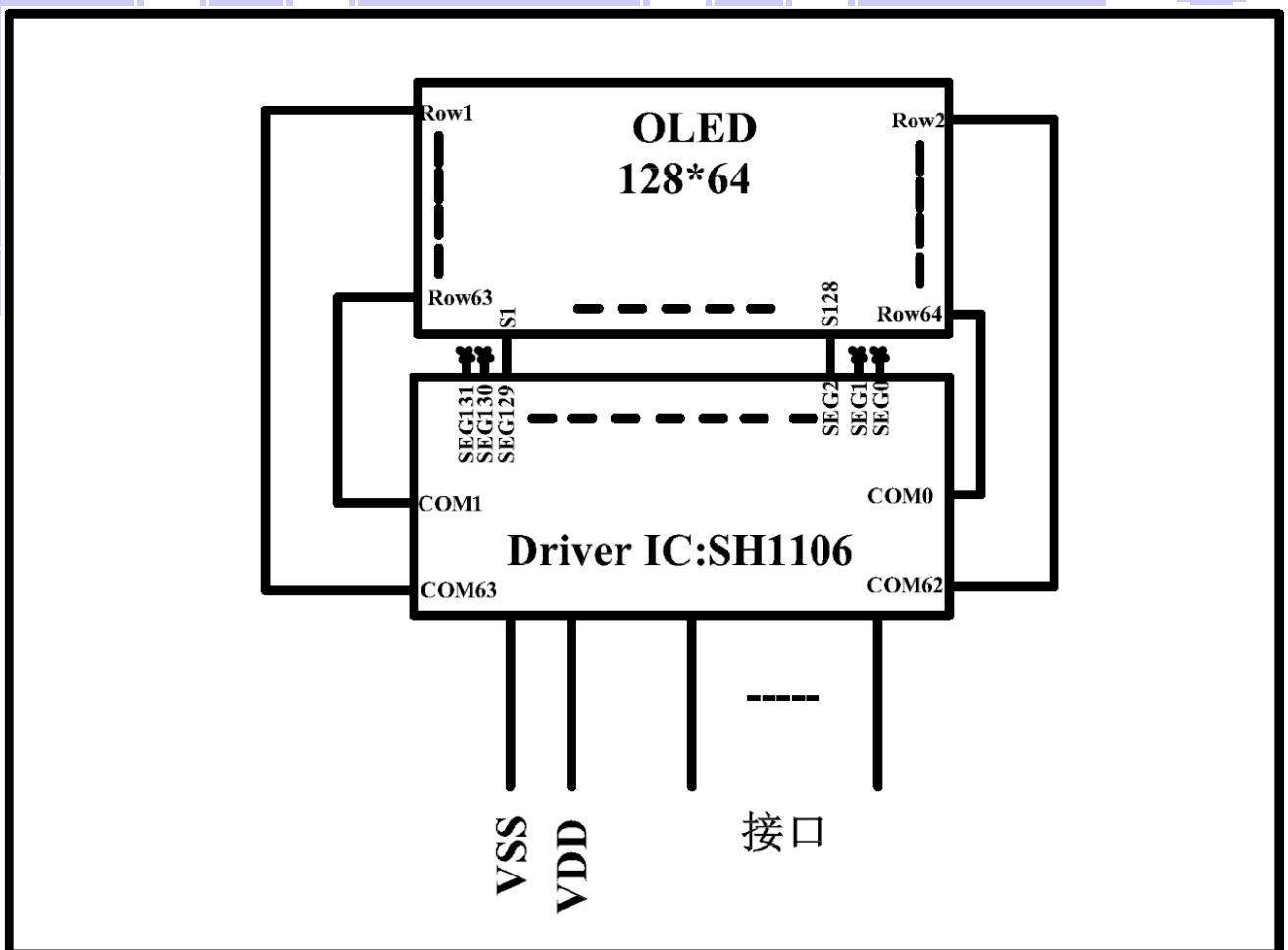


图 2. JLX12864OLED-13001

5. 技术参数

5.1 最大极限参数 (超过极限参数则会损坏 OLED 模块)

名称	符号	标准值			单位
		最小	典型	最大	
电路电源	VDD - VSS	2.4	3.3	5.2	V
OLED 驱动电压	VCC	7.0	—	14	V
静电电压		—	—	100	V
工作温度		-20		+70	°C
储存温度		-30		+80	°C

表 2: 最大极限参数

5.2 直流 (DC) 参数

名称	符号	测试条件	标准值			单位
			MIN	TYPE	MAX	
工作电压	VDD		2.4	3.3	5.2	V
输入高电平	V _{IHC}		0.8xVDD	—	VDD	V
输入低电平	V _{ILC}		VSS	—	0.2xVDD	V
输出高电平	V _{OHC}	I _{OH} = 0.2mA	0.8xVDD	—	VDD	V
输出低电平	V _{OHC}	I _{OO} = 1.2mA	VSS	—	0.2xVDD	V
模块工作电流	I _{DD}	VDD = 3.3V	—		0.3	mA

表 3: 直流 (DC) 参数

6. 读写时序特性

6.1 串行接口:

从 CPU 写到 SH1106 (Writing Data from CPU to SH1106)

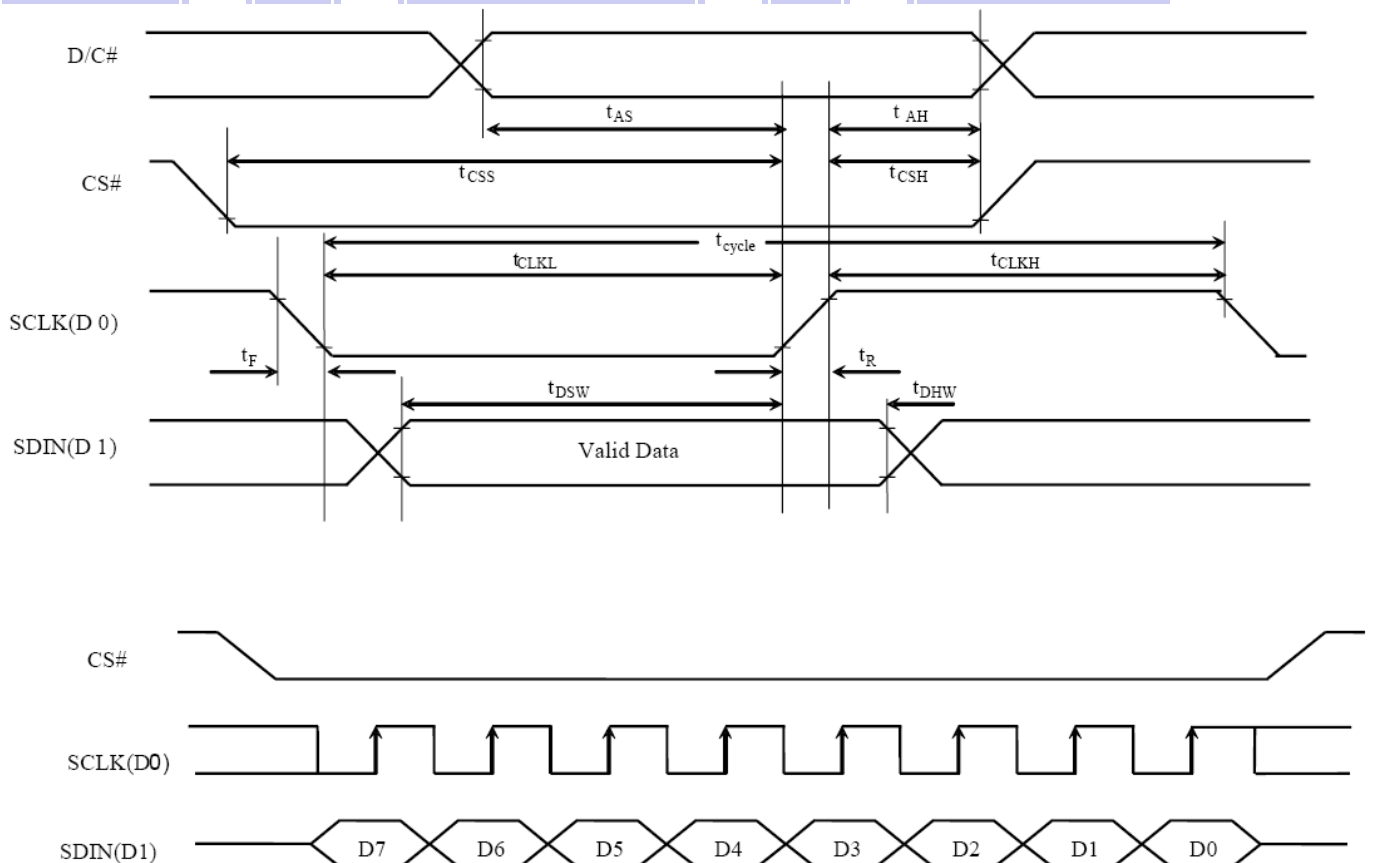


图 3. 从 CPU 写到 SH1106 (Writing Data from CPU to SH1106)

6.2 串行接口: 时序要求 (AC 参数):

写数据到 SH1106 的时序要求:

表 4.

项目	符号	测试条件	极限值			单位
			MIN	TYPE	MAX	
4线 SPI串口时钟周期 (4-line SPI Clock Period)	T_{scyc}	引脚: SCK	100	—	—	ns
保持SCK高电平脉宽 (SCK "H" pulse width)	T_{shw}	引脚: SCK	20	—	—	ns
保持SCK低电平脉宽 (SCK "L" pulse width)	T_{SLW}	引脚: SCK	20	—	—	ns
地址建立时间 (Address setup time)	T_{sAS}	引脚: RS	15	—	—	ns
地址保持时间 (Address hold time)	T_{sah}	引脚: RS	15	—	—	ns
数据建立时间 (Data setup time)	T_{sds}	引脚: SI	15	—	—	ns
数据保持时间 (Data hold time)	T_{SDH}	引脚: SI	15	—	—	ns
片选信号建立时间 (CS-SCL time)	T_{css}	引脚: CS	20	—	—	ns
片选信号保持时间 (CS-SCL time)	T_{csh}	引脚: CS	10	—	—	ns

* (VDD = 1.65V~3.3V, Ta = 25°C)

6.3 电源启动后复位的时序要求 (RESET CONDITION AFTER POWER UP):

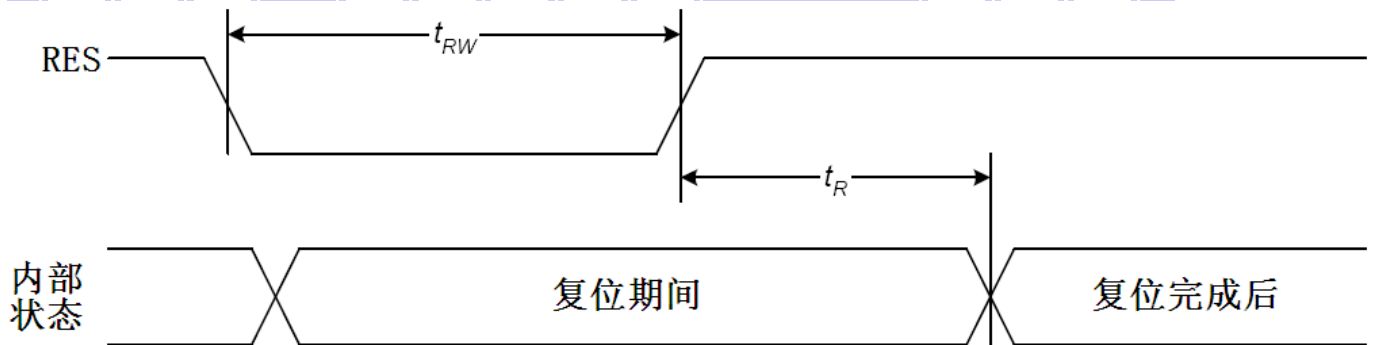


图 4: 电源启动后复位的时序

表 5: 电源启动后复位的时序要求

项目	符号	测试条件	极限值			单位
			MIN	TYPE	MAX	
复位时间	t_R		—	—	1.0	us
复位保持低电平的时间	t_{RW}	引脚: RES	5.0	—	—	us

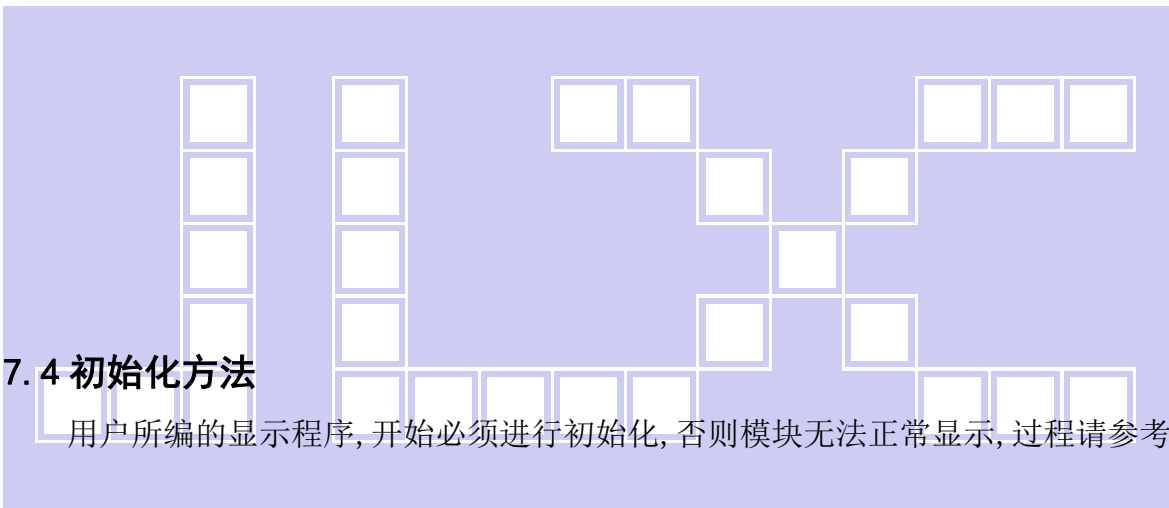
7. 指令功能:

7.1 指令表

表 6

指令名称		指令码								说明	
		RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1		DB0
(1) 显示开/关 (display on/off)		0	1	0	1	0	1	1	1	0 1	显示开/关: 0xAE : 关, 0xAF : 开
(2) 显示初始行设置 (Display start line set)		0	0	1	显示初始行地址, 共 6 位						设置显示存储器的显示初始行, 可设置值为 0x40~0x7F , 分别代表第 0~63 行, 针对该 OLED 屏一般设置为 0x40
(3) 页地址设置 (Page address set)		0	1	0	1	1	显示页地址, 共 4 位				设置页地址。每 8 行为一个页, 64 行分为 8 个页, 可设置值为: 0xB0~0xB8 分别对应第一页到第九页, 第九页是一个单独的一行图标, 本 OLED 屏没有这一行图标, 所以设置值为 0xB0~0xB7 分别对应第一页~第八页。
(4)	列地址高4位设置	0	0	0	0	1	列地址的高 4 位				高 4 位与低 4 位共同组成列地址, 指定 128 列中的其中一列。比如 OLED 模块的第 100 列地址十六进制为 0x64 , 那么此指令由 2 个字节来表达: 0x16, 0x04
	列地址低4位设置		0	0	0	0	列地址的低 4 位				
(5) 读状态 (Status read)		0	状态				0	0	0	0	串口时: 读驱动IC的当前状态, 串口时不能用此指令。
(6) 写显示数据到 OLED 屏 (Display data write)		1	8 位显示数据								从 CPU 写数据到 OLED 屏, 每一位对应一个点阵, 1 个字节对应 8 个竖置的点阵
(7) 读 OLED 屏的显示数据 (Display data read)		1	8 位显示数据								串口时: 读已经显示到 OLED 屏上的点阵数据。 串口时不能用此指令。
(8) 显示列地址增减 (ADC select)			1	0	1	0	0	0	0	0 1	显示列地址增减: 0xA0 : 反转: 列地址从右到左, 0xA1 : 常规: 列地址从左到右
(9) 显示正显/反显 (Display normal/reverse)		0	1	0	1	0	0	1	1	0 1	显示正显/反显: 0xA6 : 常规: 正显 0xA7 : 反显
(10) 显示全部点阵 (Display all points)		0	1	0	1	0	0	1	0	0 1	显示全部点阵: 0xA4 : 常规 0xA5 : 显示全部点阵
(11) 行扫描顺序选择 (Common output mode select)			1	1	0	0	0 1	0	0	0	行扫描顺序选择: 0xC0 : 普通扫描顺序: 从上到下 0xC8 : 反转扫描顺序: 从下到上
(12) OLED 振荡频率设置 (Oscillator Frequency)		0	1	1	0	1	0	1	0	1	设置振荡频率: 范围: 0000-1111 , 参考指令: 0Xd5 0X80
(13) 电源控制 (Power control set)		0	1	0	0	0	1	1	0	1	设置升压: 0X8d 0X14

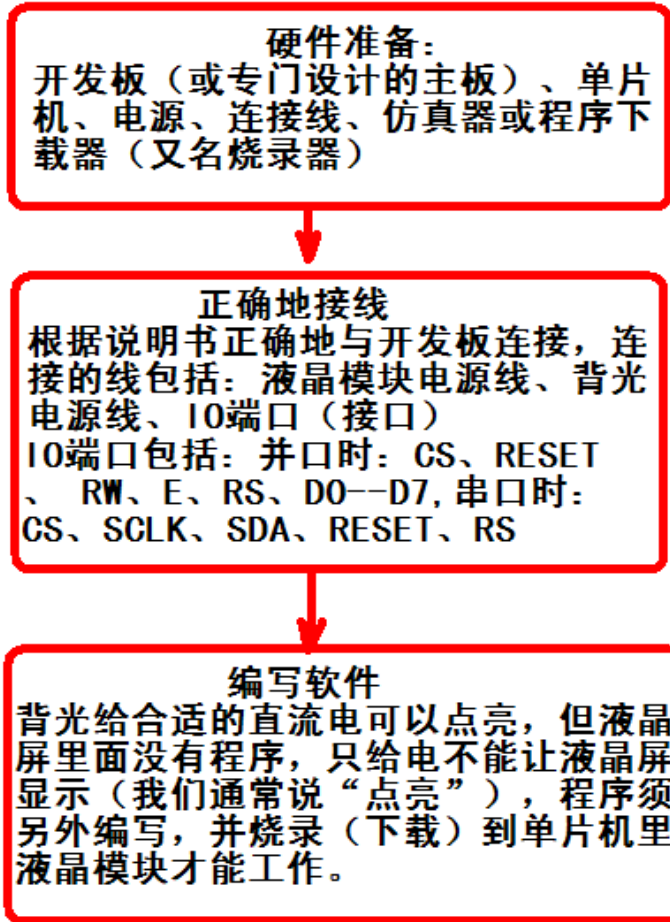
(14)	内部设置OLED电压模式	0	1	0	0	0	0	0	0	1	设置内部电阻微调，可以理解为 微调 对比度值，此两个指令需紧接着使用。上面一条指令 0x81 是不改的，下面一条指令可设置范围为： 0x00~0xFF ,数值越大对比度越浓，越小越淡
	设置的电压值		0	0	6位电压值数据，0~63共64级						
(15)	静态图标显示： 开/关	0	1	0	1	0	1	1	1	0 1	静态图标的开关设置： 0xAE : 关, 0xAF : 开。 此指令在进入及退出睡眠模式时起作用
(16)	省电模式 (Power save)										省电模式，此非一条指令，是由“(10)显示全部点阵”、(19)静态图标显示：开/关等指令合成一个“省电功能”。详细看 IC 规格书 “POWER SAVE”部分
(17)	空指令 (NOP)	0	1	1	1	0	0	0	1	1	空操作



7.4 初始化方法

用户所编的显示程序,开始必须进行初始化,否则模块无法正常显示,过程请参考程序

点亮液晶模块的步骤



7.5 程序举例:

OLED 模块与 MPU(以 8051 系列单片机为例)接口图如下:

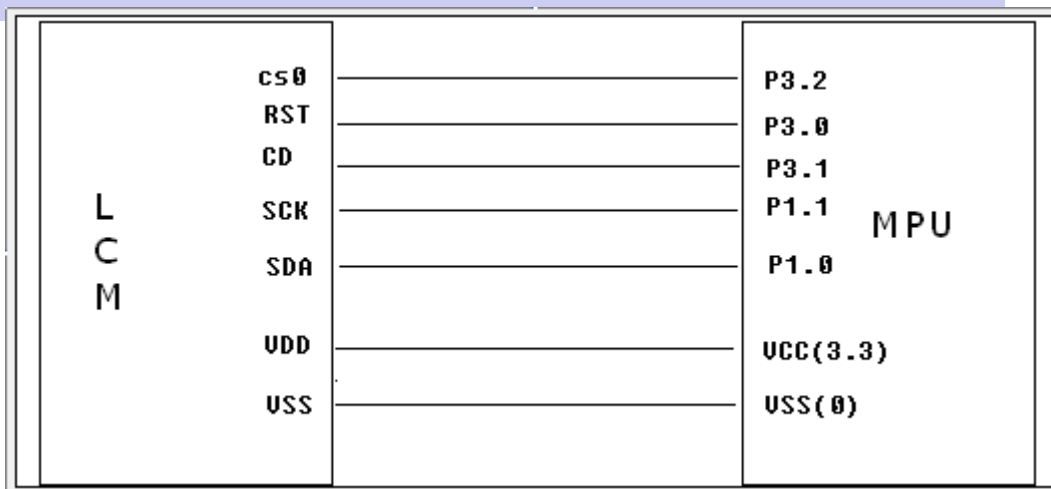


图 5.串行接口

7.5.1 程序:

```

// OLED 演示程序
// OLED 模块型号: JLX12864OLED-13001, 串行接口!
// 驱动 IC 是:SH1106
// 资料(源程序、驱动手册、使用说明书等)销售统一发
  
```

```
#include <reg52.H>

//=====

sbit lcd_sclk =P1^1; //接口定义:lcd_sclk 就是 LCD 的 SCLK //SCLK 接到 “D0” 脚
sbit lcd_sda =P1^0; //接口定义:lcd_sda 就是 LCD 的 SDA //SDIN 接到 “D1” 脚
sbit lcd_reset=P3^0; //接口定义:lcd_reset 就是 LCD 的 RESET
sbit lcd_dc =P3^1; //接口定义:lcd_dc 就是 LCD 的 D/C (RS)

sbit lcd_cs1=P3^2; //接口定义:lcd_cs1 就是 LCD 的 CS
sbit key=P2^0; //定义一个按键: P2.0 口与 GND 之间接一个按键

#define uchar unsigned char
#define uint unsigned int
#define ulong unsigned long

uchar code jiong1[]={/*— 文字: 囧 —*/
/*— 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 —*/
0x00, 0xFE, 0x82, 0x42, 0xA2, 0x9E, 0x8A, 0x82, 0x86, 0x8A, 0xB2, 0x62, 0x02, 0xFE, 0x00, 0x00,
0x00, 0x7F, 0x40, 0x40, 0x7F, 0x40, 0x40, 0x40, 0x40, 0x40, 0x7F, 0x40, 0x40, 0x7F, 0x00, 0x00};

uchar code lei1[]={/*— 文字: 晶 —*/
/*— 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 —*/
0x80, 0x80, 0x80, 0xBF, 0xA5, 0xA5, 0xA5, 0x3F, 0xA5, 0xA5, 0xA5, 0xBF, 0x80, 0x80, 0x80, 0x00,
0x7F, 0x24, 0x24, 0x3F, 0x24, 0x24, 0x7F, 0x00, 0x7F, 0x24, 0x24, 0x3F, 0x24, 0x24, 0x7F, 0x00};

//延时
void delay(int i)
{
    int j,k;
    for(j=0;j<i;j++)
        for(k=0;k<110;k++);
}

//等待按键: P2.0 口与 GND 之间接一个按键
void waitkey()
{
    repeat:    if(key==1) goto repeat;
              else delay(1200);
}

//写指令到 OLED 显示模块
void transfer_command(int data1)
{
    uchar i;
    lcd_cs1=0;
    lcd_dc= 0;
    for(i=0;i<8;i++)
    {
```



```

        lcd_sclk = 0;
        if (data1 & 0x80)  lcd_sda = 1;
        else                lcd_sda = 0;
        lcd_sclk = 1;
        data1 <<= 1;
    }
    lcd_cs1=1;
}

```

//写数据到 OLED 显示模块

```
void transfer_data(int data1)
```

```

{
    uchar i;
    lcd_cs1=0;
    lcd_dc= 1;
    for(i=0;i<8;i++)
    {

```

```

        lcd_sclk = 0;
        if (data1 & 0x80)  lcd_sda = 1;
        else                lcd_sda = 0;
        lcd_sclk = 1;
        data1 <<= 1;
    }
    lcd_cs1=1;
}

```

//OLED 显示模块初始化

```
void initial_lcd()
```

```

{
    delay(200);
    lcd_cs1=0;
    Rom_CS = 1;
    lcd_reset=0;        //低电平复位
    delay(200);
    lcd_reset=1;        //复位完毕
    delay(200);
    transfer_command(0xae); //关显示
    transfer_command(0xd5); //晶振频率
    transfer_command(0x80);
    transfer_command(0xa8); //duty 设置
    transfer_command(0x3f); //duty=1/64
    transfer_command(0xd3); //显示偏移
    transfer_command(0x00);
    transfer_command(0x40); //起始行
    transfer_command(0x8d); //升压允许
    transfer_command(0x14);
    transfer_command(0x20); //page address mode
}

```



```

transfer_command(0x02);
transfer_command(0xc8); //行扫描顺序: 从上到下
transfer_command(0xa1); //列扫描顺序: 从左到右
transfer_command(0xda); //sequential configuration
transfer_command(0x12);
transfer_command(0x81); //微调对比度, 本指令的 0x81 不要改动, 改下面的值
transfer_command(0xcf); //微调对比度的值, 可设置范围 0x00~0xff
transfer_command(0xd9); //Set Pre-Charge Period
transfer_command(0xf1);
transfer_command(0xdb); //Set VCOMH Deselect Level
transfer_command(0x40);
transfer_command(0xaf); //开显示
}

void lcd_address(uchar page, uchar column)
{
    column=column+1; //或白屏 -1 //我们平常所说的第 1 列, 在 LCD 驱动 IC 里是第 0 列。所以在这里减去 1.
    page=page-1;
    transfer_command(0xb0+page); //设置页地址。每页是 8 行。一个画面的 64 行被分成 8 个页。我们平常所说的第 1 页, 在 LCD 驱动
    IC 里是第 0 页, 所以在这里减去 1
    transfer_command(((column>>4)&0x0f)+0x10); //设置列地址的高 4 位
    transfer_command(column&0x0f); //设置列地址的低 4 位
}

//全屏清屏
void clear_screen()
{
    unsigned char i, j;
    for(j=0; j<8; j++)
    {
        lcd_address(1+j, 1);
        for(i=0; i<128; i++)
        {
            transfer_data(0x00);
        }
    }
}

/*显示 16x16 点阵图像、汉字、生僻字或 16x16 点阵的其他图标*/
void display_graphic_16x16(uint page, uint column, uchar *dp)
{
    uint i, j;
    lcd_cs1=0;
    Rom_CS = 1;
    for(j=0; j<2; j++)
    {
        lcd_address(page+j, column);
        for (i=0; i<16; i++)
        {
            transfer_data(*dp); //写数据到 LCD, 每写完一个 8 位的数据后列地址自动加 1*/
        }
    }
}

```



```

        dp++;
    }
}
lcd_cs1=1;
}
//显示 8x16 点阵图像、ASCII, 或 8x16 点阵的自造字符、其他图标
void display_graphic_8x16(uchar page,uchar column,uchar *dp)
{
    uchar i,j;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j,column);
        for (i=0;i<8;i++)
        {
            transfer_data(*dp);                //写数据到LCD, 每写完一个 8 位的数据后列地址自动加 1
            dp++;
        }
    }
}

```

```

/*显示 5*7 点阵图像、ASCII, 或 5x7 点阵的自造字符、其他图标*/
void display_graphic_5x7(uint page,uchar column,uchar *dp)
{
    uint col_cnt;
    uchar page_address;
    uchar column_address_L,column_address_H;
    page_address = 0xb0+page-1;
    lcd_cs1=0;
    column=column+1;
    column_address_L=(column&0x0f);
    column_address_H=((column>>4)&0x0f)+0x10;
    transfer_command(page_address);        /*Set Page Address*/
    transfer_command(column_address_H);    /*Set MSB of column Address*/
    transfer_command(column_address_L);    /*Set LSB of column Address*/

    for (col_cnt=0;col_cnt<6;col_cnt++)
    {
        transfer_data(*dp);
        dp++;
    }
    lcd_cs1=1;
}

```

/****送指令到晶联讯字库 IC****/

```

void send_command_to_ROM(uchar datu )
{
    uchar i;
    for(i=0;i<8;i++ )
    {

```



```

    if(datu&0x80)
        Rom_IN = 1;
    else
        Rom_IN = 0;
        datu = datu<<1;
        Rom_SCK=0;
        Rom_SCK=1;
}
}
/****从晶联讯字库 IC 中取汉字或字符数据（1 个字节）****/
static uchar get_data_from_ROM( )
{
    uchar i;
    uchar ret_data=0;
    Rom_SCK=1;
    for(i=0;i<8;i++)
    {
        Rom_OUT=1;
        Rom_SCK=0;
        ret_data=ret_data<<1;
        if( Rom_OUT )
            ret_data=ret_data+1;
        else
            ret_data=ret_data+0;
        Rom_SCK=1;
    }
    return(ret_data);
}
/*从相关地址（addrHigh: 地址高字节,addrMid: 地址中字节,addrLow: 地址低字节）中连续读出 DataLen 个字节的数据到 pBuff 的地址*/
/*连续读取*/
void get_n_bytes_data_from_ROM(uchar addrHigh,uchar addrMid,uchar addrLow,uchar *pBuff,uchar DataLen )
{
    uchar i;
    Rom_CS = 0;
    lcd_cs1=1;
    Rom_SCK=0;
    send_command_to_ROM(0x03);
    send_command_to_ROM(addrHigh);
    send_command_to_ROM(addrMid);
    send_command_to_ROM(addrLow);
    for(i = 0; i < DataLen; i++)
        *(pBuff+i) =get_data_from_ROM();
    Rom_CS = 1;
}
/*****/
ulong fontaddr=0;
void display_GB2312_string(uchar y,uchar x,uchar *text)

```



```

{
    uchar i= 0;
    uchar addrHigh, addrMid, addrLow ;
    uchar fontbuf[32];
    while((text[i]>0x00))
    {
        if(((text[i]>=0xb0) &&(text[i]<=0xf7))&&(text[i+1]>=0xa1))
        {
            /*国标简体 (GB2312) 汉字在晶联讯字库 IC 中的地址由以下公式来计算: */
            /*Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 846)*32+ BaseAdd;BaseAdd=0*/
            /*由于担心 8 位单片机有乘法溢出问题, 所以分三部取地址*/
            fontaddr = (text[i]- 0xb0)*94;
            fontaddr += (text[i+1]-0xa1)+846;
            fontaddr = (ulong) (fontaddr*32);
            addrHigh = (fontaddr&0xff0000)>>16; /*地址的高 8 位, 共 24 位*/
            addrMid = (fontaddr&0xff00)>>8; /*地址的中 8 位, 共 24 位*/
            addrLow = fontaddr&0xff; /*地址的低 8 位, 共 24 位*/
            get_n_bytes_data_from_ROM(addrHigh, addrMid, addrLow, fontbuf, 32 );/*取 32 个字节的数据, 存到"fontbuf[32]"*/
            display_graphic_16x16(y, x, fontbuf);/*显示汉字到 LCD 上, y 为页地址, x 为列地址, fontbuf[]为数据*/
            i+=2;
            x+=16;
        }
        else if(((text[i]>=0xa1) &&(text[i]<=0xa3))&&(text[i+1]>=0xa1))
        {
            /*国标简体 (GB2312) 15x16 点的字符在晶联讯字库 IC 中的地址由以下公式来计算: */
            /*Address = ((MSB - 0xa1) * 94 + (LSB - 0xA1))*32+ BaseAdd;BaseAdd=0*/
            /*由于担心 8 位单片机有乘法溢出问题, 所以分三部取地址*/
            fontaddr = (text[i]- 0xa1)*94;
            fontaddr += (text[i+1]-0xa1);
            fontaddr = (ulong) (fontaddr*32);
            addrHigh = (fontaddr&0xff0000)>>16; /*地址的高 8 位, 共 24 位*/
            addrMid = (fontaddr&0xff00)>>8; /*地址的中 8 位, 共 24 位*/
            addrLow = fontaddr&0xff; /*地址的低 8 位, 共 24 位*/
            get_n_bytes_data_from_ROM(addrHigh, addrMid, addrLow, fontbuf, 32 );/*取 32 个字节的数据, 存到"fontbuf[32]"*/
            display_graphic_16x16(y, x, fontbuf);/*显示汉字到 LCD 上, y 为页地址, x 为列地址, fontbuf[]为数据*/
            i+=2;
            x+=16;
        }
    }
    else if((text[i]>=0x20) &&(text[i]<=0x7e))
    {
        unsigned char fontbuf[16];
        fontaddr = (text[i]- 0x20);
        fontaddr = (unsigned long) (fontaddr*16);
        fontaddr = (unsigned long) (fontaddr+0x3cf80);
        addrHigh = (fontaddr&0xff0000)>>16;
        addrMid = (fontaddr&0xff00)>>8;
        addrLow = fontaddr&0xff;
    }
}
    
```



```

        get_n_bytes_data_from_ROM(addrHigh, addrMid, addrLow, fontbuf, 16 );/*取 16 个字节的数
        据, 存到"fontbuf[32]"*/
        display_graphic_8x16(y, x, fontbuf);/*显示 8x16 的 ASCII 字到 LCD 上, y 为页地址, x 为列地址, fontbuf[]为数据*/
        i+=1;
        x+=8;
    }
    else
        i++;
}
}

void display_string_5x7(uchar y, uchar x, uchar *text)
{
    unsigned char i= 0;
    unsigned char addrHigh, addrMid, addrLow ;
    while((text[i]>0x00))
    {
        if((text[i]>=0x20) &&(text[i]<=0x7e))
        {
            unsigned char fontbuf[8];
            fontaddr = (text[i]- 0x20);
            fontaddr = (unsigned long) (fontaddr*8);
            fontaddr = (unsigned long) (fontaddr+0x3bfc0);
            addrHigh = (fontaddr&0xff0000)>>16;
            addrMid = (fontaddr&0xff00)>>8;
            addrLow = fontaddr&0xff;
            get_n_bytes_data_from_ROM(addrHigh, addrMid, addrLow, fontbuf, 8);/*取 8 个字节的数
            据, 存到"fontbuf[32]"*/
            display_graphic_5x7(y, x, fontbuf);/*显示 5x7 的 ASCII 字到 LCD 上, y 为页地址, x 为列地址, fontbuf[]为数据*/
            i+=1;
            x+=6;
        }
        else
            i++;
    }
}

void main(void)
{
    while(1)
    {
        initial_lcd();                //初始化
        Rom_CS=1;
        lcd_cs1=0;
        clear_screen();    //clear all dots
        display_GB2312_string(1, 1, "JLX-OLED-13001 ");    /*在第 1 页, 第 1 列, 显示一串 16x16 点阵汉字或 8x16 的 ASCII 字*/
        display_GB2312_string(3, 1, "16X16 简体汉字库,"); /*显示一串 16x16 点阵汉字或 8x16 的 ASCII 字. 以下雷同*/
        display_GB2312_string(5, 1, "或 8X16 点阵 ASCII,");
        display_GB2312_string(7, 1, "或 5X7 点阵 ASCII 码");
        waitkey();
    }
}

```




```

clear_screen();
display_GB2312_string(1,1,"晶联讯成立于二零");
display_GB2312_string(3,1,"零四年十一月七日");
display_GB2312_string(5,1,"主要生产液晶模块");
display_GB2312_string(7,1,"品质至上真诚服务");
waitkey();
display_GB2312_string(1,1,"GB2312 简体字库及");
display_GB2312_string(3,1,"有图型功能, 可自");
display_GB2312_string(5,1,"编大字或图像或生");
display_GB2312_string(7,1,"僻字, 例如:   ");
display_graphic_16x16(7,97,jiong1);           /*在第7页, 第81列显示单个自编生僻汉字"囧"*/
display_graphic_16x16(7,113,lei1);           /*显示单个自编生僻汉字"晶"*/
waitkey();
clear_screen();
display_GB2312_string(1,1,"<!@#%^&*()-+]/"); /*在第1页, 第1列, 显示一串16x16点阵汉字或8*16的ASCII字*/
display_string_5x7(3,1,"<!@#%^&*()-+]/;.:?["); /*在第3页, 第1列, 显示一串5x7点阵的ASCII字*/
display_string_5x7(4,1,"JLX electronics Co., "); /*显示一串5x7点阵的ASCII字*/
display_string_5x7(5,1,"Ltd. established at "); /*显示一串5x7点阵的ASCII字*/
display_string_5x7(6,1,"year 2004.Focus LCM. "); /*显示一串5x7点阵的ASCII字*/
display_string_5x7(7,1,"TEL:0755-29784961-809"); /*显示一串5x7点阵的ASCII字*/
display_string_5x7(8,1,"FAX:0755-29784964 "); /*显示一串5x7点阵的ASCII字*/
waitkey();
}
}

```

