

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	6~7
7	指令功能及硬件接口与编程案例	8~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。

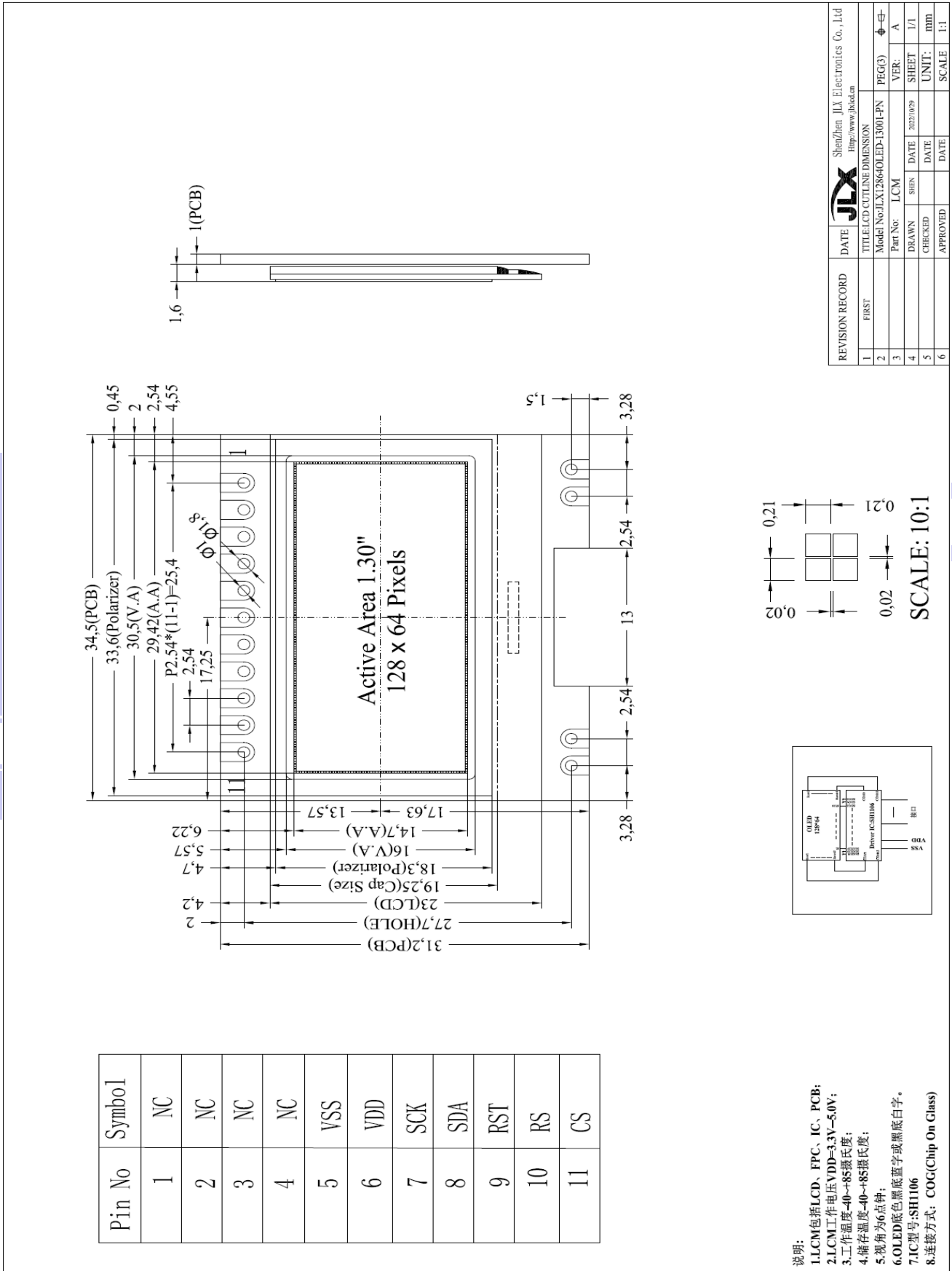
2.7 工作温度宽：-40℃~+85℃；

2.8 储存温度宽：-40℃~+85℃；

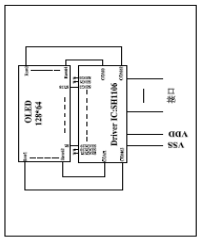
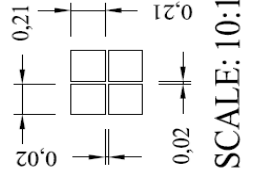


3. 外形尺寸及接口引脚功能

3.1 外形图



REVISION RECORD		DATE		TITLE/LCD CUTLINE DIMENSION		Model No: JLX12864OLED-13001-PN		Part No: LCM		VER: A		PEG(3)	
1	FIRST												
2													
3													
4													
5													
6													



说明:
 1.LCM包括LCD、FPC、IC、PCB;
 2.LCM工作电压VDD=3.3V~5.0V;
 3.工作温度-40~+85摄氏度;
 4.储存温度-40~+85摄氏度;
 5.视角为6点钟;
 6.OLED底色黑色/底蓝字或黑底白字。
 7.IC型号:SH1106
 8.连接方式: COG(Chip On Glass)

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

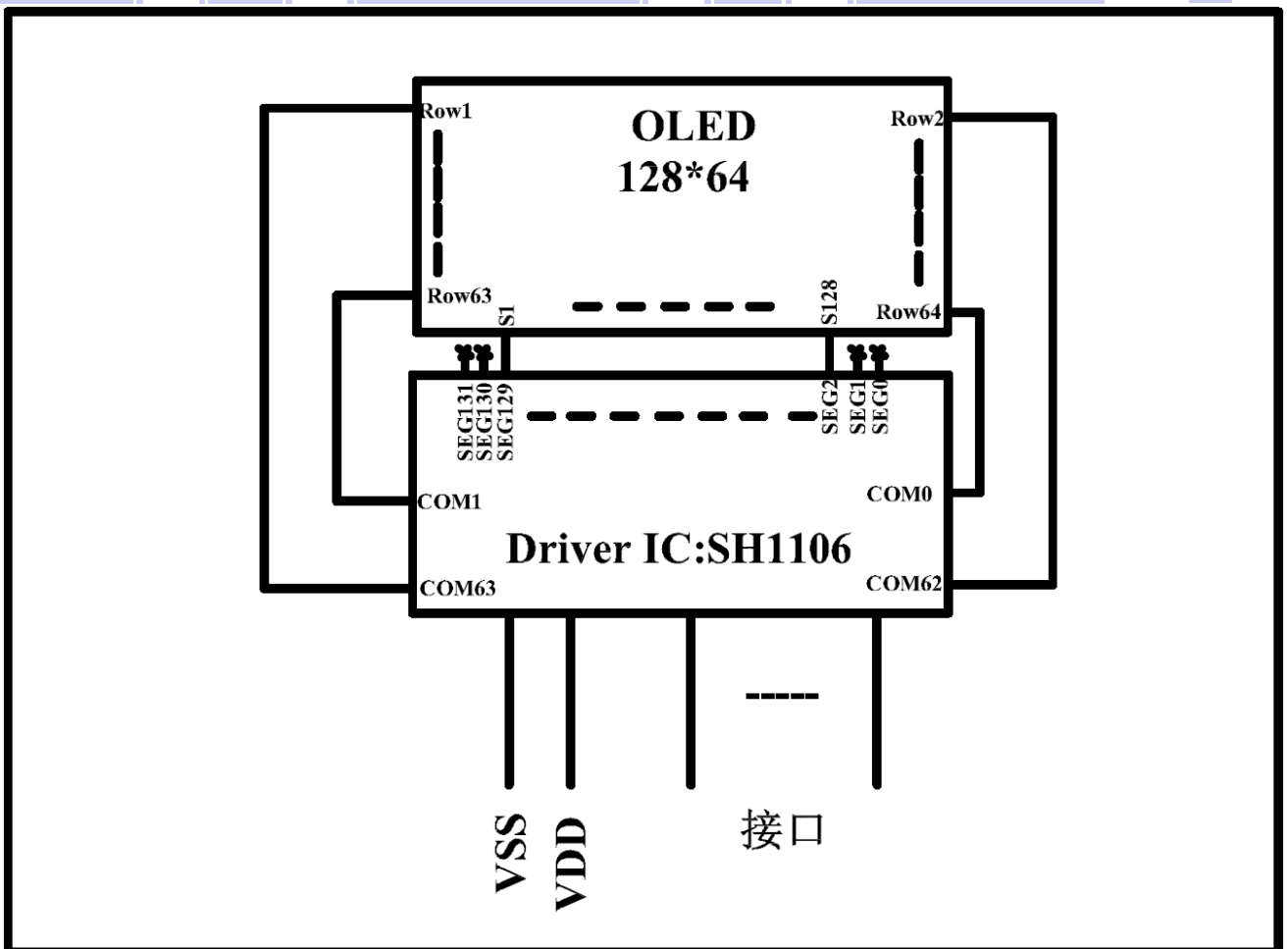


图 2. JLX12864OLED-13001

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）

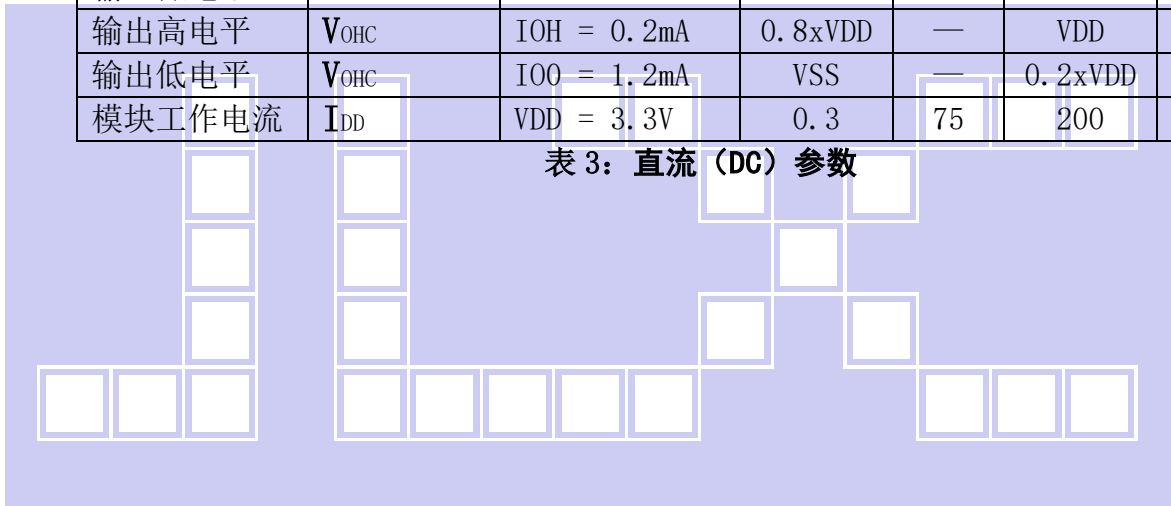
名称	符号	标准值			单位
		最小	典型	最大	
电路电源	VDD - VSS	2.7	3.3	5.2	V
OLED 驱动电压	VCC	7.0	—	14	V
静电电压		—	—	100	V
工作温度		-40		+85	°C
储存温度		-40		+85	°C

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		2.7	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	75	200	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)

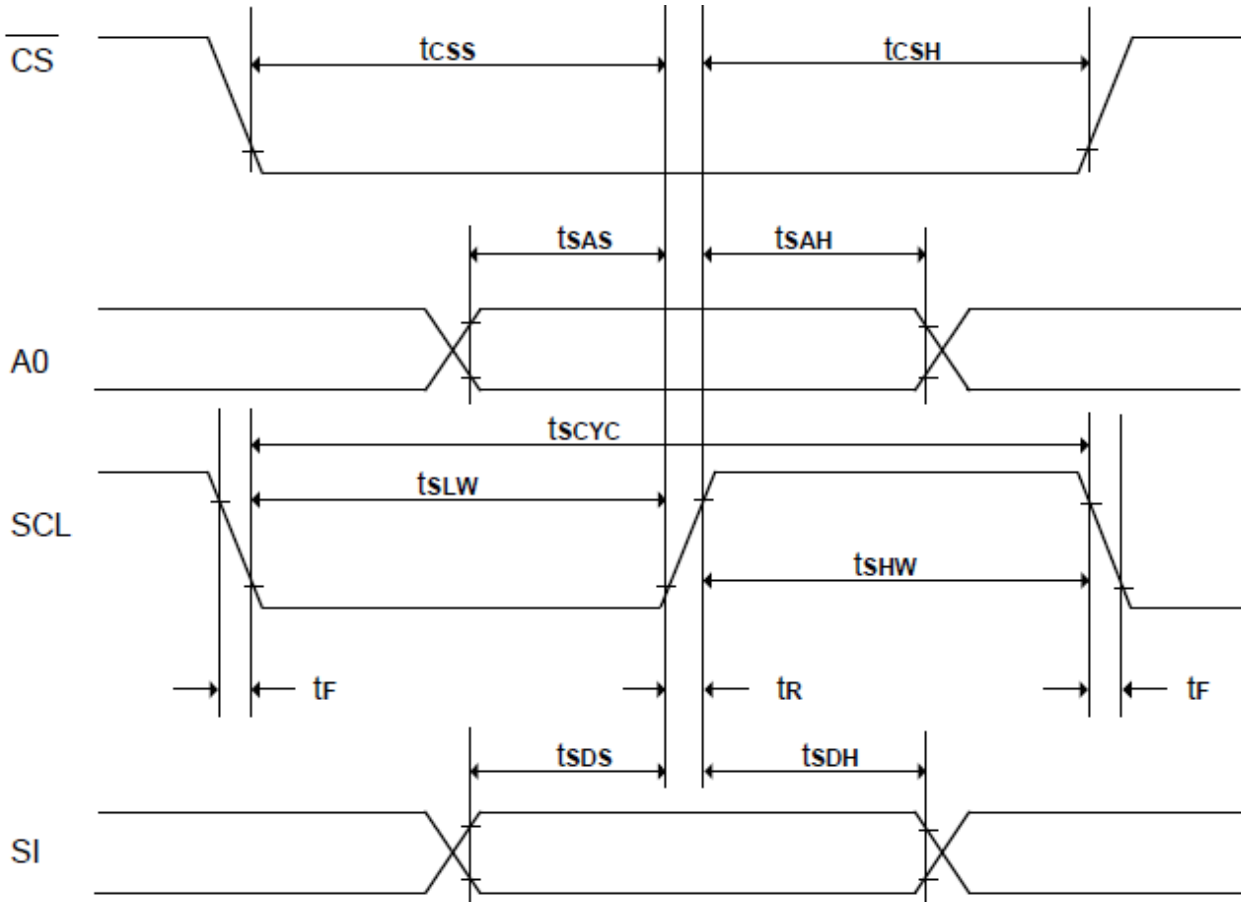


图 3. 从 CPU 写到 SH1106 (Writing Data from CPU to SH1106)

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}		250	—	—	ns
地址建立时间 (Address setup time)	T_{sAS}		150	—	—	ns
地址保持时间 (Address hold time)	T_{sAH}		150	—	—	ns
数据建立时间 (Data setup time)	T_{sDS}		100	—	—	ns
数据保持时间 (Data hold time)	T_{SDH}		100	—	—	ns
片选信号建立时间 (CS-SCL time)	T_{cSS}		120	—	—	ns
片选信号保持时间 (CS-SCL time)	T_{cSH}		60	—	—	ns

保持SCK高电平脉宽 (SCK "H" pulse width)	T_{shw}		100	—	—	ns
保持SCK低电平脉宽 (SCK "L" pulse width)	T_{slw}		100	—	—	ns
上升时间 (Rise time)	T_R				15	ns
下降时间 (Fall time)	T_F				15	ns

* (VDD = 2.4V~3.5V, Ta = 25°C)

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

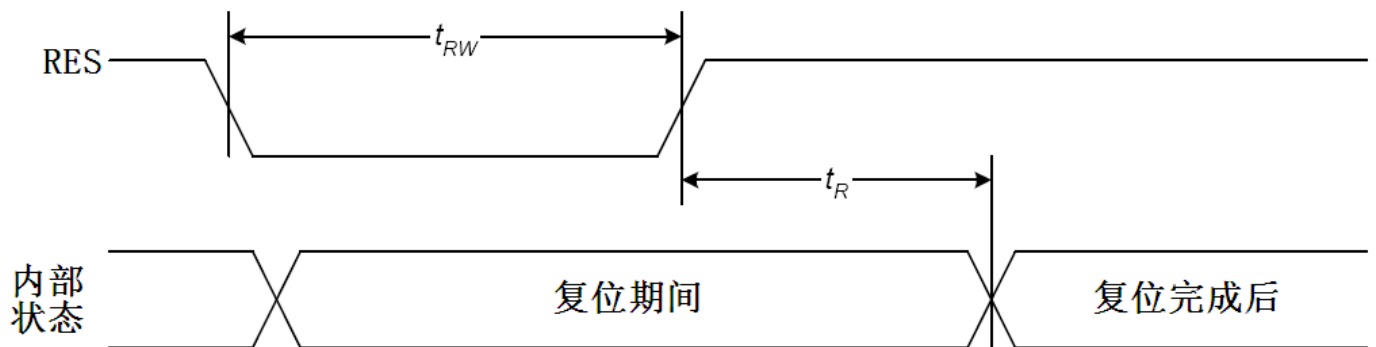


图 4. 电源启动后复位的时序

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		500	—	—	ms
复位保持低电平的时间	t_{RW}	引脚: RES	500	—	—	ms

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)列地址低4位设置	0	0	0	0	0	列地址的低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																			
(2)列地址高4位设置		0	0	0	1	列地址的高4位																								
(3)设定升压峰值 (Set Pump voltage value)	0	0	0	1	1	0	0	A1	A0	<table border="1"> <thead> <tr> <th>A1</th> <th>A0</th> <th>升压峰值</th> <th>指令</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>6.4V</td> <td>0x30</td> </tr> <tr> <td>0</td> <td>1</td> <td>7.4V</td> <td>0x31</td> </tr> <tr> <td>1</td> <td>0</td> <td>8.0V</td> <td>0x32</td> </tr> <tr> <td>1</td> <td>1</td> <td>9.0V</td> <td>0x33</td> </tr> </tbody> </table>	A1	A0	升压峰值	指令	0	0	6.4V	0x30	0	1	7.4V	0x31	1	0	8.0V	0x32	1	1	9.0V	0x33
A1	A0	升压峰值	指令																											
0	0	6.4V	0x30																											
0	1	7.4V	0x31																											
1	0	8.0V	0x32																											
1	1	9.0V	0x33																											
(4)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																			
(5) 设置对比度	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																				
	0	8位电压值数据,0~255共256级																												
(6) 显示列地址增减 (ADC select)	0	1	0	1	0	0	0	0	ADC	显示列地址增减: 0 0xA0: 常规:列地址从右到左, 1 0xA1: 反转:列地址从左到右																				
(7) 设置常规/打开全部点阵 (Set Entire Display OFF/ON)	0	1	0	1	0	0	1	0	D	设置常规显示/打开全部点阵 0 0xA4: 常规显示,写什么内容显示什么 1 0xA5: 全部点阵点亮,之前的显示会被覆盖																				
(8)显示正显/反显 (Display normal/reverse)	0	1	0	1	0	0	1	1	D	显示正显/反显: 0 0xA6: 常规:正显 1 0xA7: 反显																				
(9)设置显示行数 (Set Multiplex Ration)	0	1	0	1	0	1	0	0	0	0xA8: 设置显示行数																				
	0	*	*	共6位,0~63共64级							设置范围:00~3f 针对本型号为0x3f,64行																			
(10) 设置内部/外部升压 Set DC-DC OFF/ON: (Double Bytes Command)	0	1	0	1	0	1	1	0	1	0x8d: 设置内部/外部升压																				
	0	1	0	0	0	1	0	1	D	0x8A: 使用外部升压 0x8B: 使用内部升压 1																				
(11) 显示开/关 (display on/off)	0	1	0	1	0	1	1	1	0	显示开/关: 1 0xAE: 关, 0xAF: 开																				
(12)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:0xB0~0xB7分别对应第一页到第八页。																			
(13) 行扫描顺序选择 (Common output mode)	0	1	1	0	0	D	0	0	0	行扫描顺序选择: 0x0C: 普通扫描顺序: 从下到上																				



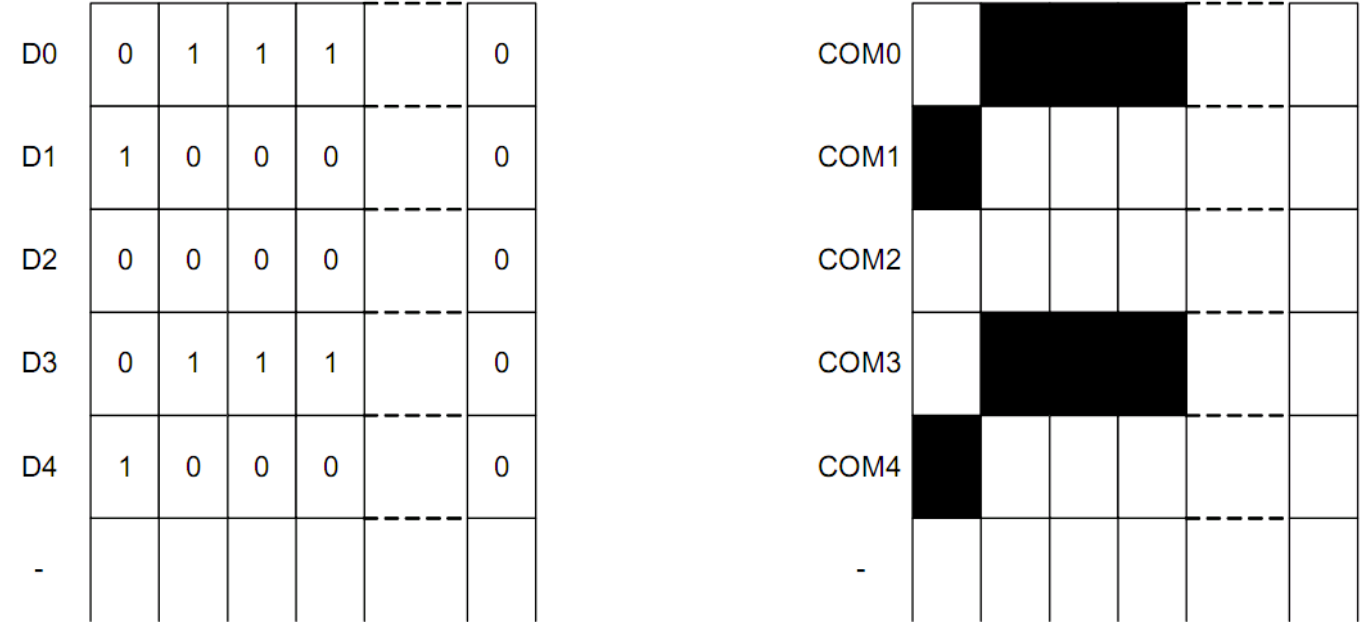
select)						1				0XC8: 反转扫描顺序: 从上到下
(14) 设置显示行偏移 (Set Display Offset: (Double Bytes Command))	0	1	1	0	1	0	0	1	1	0XD3: 设置显示行偏移
	0	*	*	共 6 位, 0~63 共 64 级						0X00: 默认, 范围: 00—3f
(15) OLED 振荡频率设置 (Oscillator Frequency(Double Bytes Command))	0	1	1	0	1	0	1	0	1	0Xd5: 振荡频率设置
	0	共 8 位, 0~255 共 256 级								0X80: 默认值, 范围: 00—ff
(16)设置预充电周期 (Set Dis-charge/Pre-charge Period: (Double Bytes Command))	0	1	1	0	1	1	0	0	1	预充电周期模式设置: 0XD9:
	0	共 8 位, 0~255 共 256 级								设置预充电时间: 0X1f: 默认值, 范围: 00—ff
(17)设置COM硬件配置 (Set Common pads hardware configuration: (Double Bytes Command))	0	1	1	0	1	1	0	1	0	设置COM硬件配置 0XDA:
	0	0	0	0	D 0 1	0	0	1	0	设置COM配置模式 0X02: 0X12: 本款型号配置
(18)设置VCOM (Set VCOM Deselect Level: (Double Bytes Command))	0	1	1	0	1	1	0	1	1	设置VCOM 0XDB:
	0	共 8 位, 0~255 共 256 级								0X40: 本款型号配置 范围: 00—ff
(19)读-改-写 (Read-Modify-Write)	0	1	1	1	0	0	0	0	0	0XE0: “读-改-写”开始。 列地址的增加: 写入时: 列地址+1 读出时: 列地址不加 详情请参考IC资料第28页
(20)退出上述“读-改- 写”(End)	0	1	1	1	0	1	1	1	0	0XEE: 上述“读-改-写”指令结束 详情请参考IC资料第28页
(21)空指令 (NOP)	0	1	1	1	0	0	0	1	1	0XE3: 空操作
(22) 写显示 (Write Display Data)	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
(23) 读状态 (Status read)	0	BUSY	ON/OFF	*	*	*	0	0	0	暂不可用
(24) 读OLED屏的显示 数据 (Read Display Data)	1	8 位显示数据								并口时: 读已经显示到OLED屏上的点阵数 据。

请详细参考 IC 资料”SH1106.PDF”。

7.2 点阵与 DD RAM 地址的对应关系

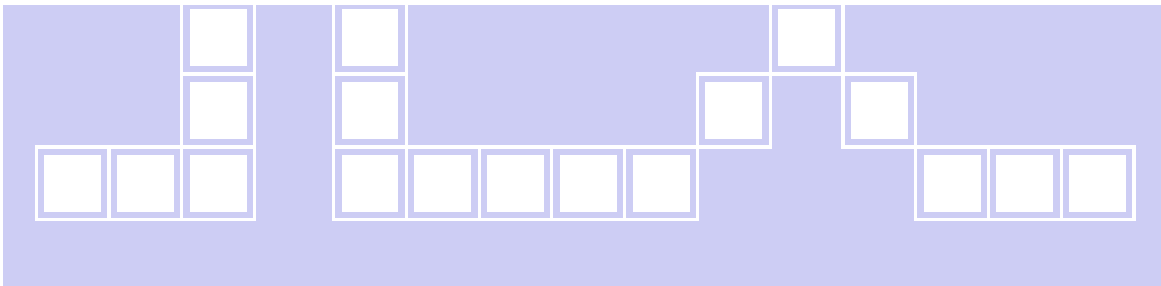
请留意页的定义：PAGE, 与平时所讲的“页”并不是一个意思，在此表示 **8 个行就是一个“页”**，一个 128*64 点阵的屏分为 8 个“页”，从第 0 “页”到第 7 “页”。

DB7—DB0 的排列方向：数据是从上向下排列的。最低位 D0 是在最上面，最高位 D7 是在最下面。每一位 (bit) 数据对应一个点阵，通常“1”代表点亮该点阵，“0”代表关掉该点阵。 如下图所示：



Display data RAM
(显示数据存储)

Liquid crystal display
(OLED)



Page Address				Data								Line Address											
D3	D2	D1	D0	D0	PAGE 0	00H	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7									
				D1											01H	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15
				D2											02H	COM16	COM17	COM18	COM19	COM20	COM21	COM22	COM23
				D3											03H	COM24	COM25	COM26	COM27	COM28	COM29	COM30	COM31
				D4											04H	COM32	COM33	COM34	COM35	COM36	COM37	COM38	COM39
				D5											05H	COM40	COM41	COM42	COM43	COM44	COM45	COM46	COM47
				D6											06H	COM48	COM49	COM50	COM51	COM52	COM53	COM54	COM55
				D7											07H	COM56	COM57	COM58	COM59	COM60	COM61	COM62	COM63
0	0	0	0	D0	PAGE1	08H	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15									
				D1											09H	COM16	COM17	COM18	COM19	COM20	COM21	COM22	COM23
				D2											0AH	COM24	COM25	COM26	COM27	COM28	COM29	COM30	COM31
				D3											0BH	COM32	COM33	COM34	COM35	COM36	COM37	COM38	COM39
				D4											0CH	COM40	COM41	COM42	COM43	COM44	COM45	COM46	COM47
				D5											0DH	COM48	COM49	COM50	COM51	COM52	COM53	COM54	COM55
				D6											0EH	COM56	COM57	COM58	COM59	COM60	COM61	COM62	COM63
				D7											0FH	COM64	COM65	COM66	COM67	COM68	COM69	COM70	COM71
0	0	1	0	D0	PAGE2	10H	COM10	COM11	COM12	COM13	COM14	COM15	COM16	COM17									
				D1											11H	COM18	COM19	COM20	COM21	COM22	COM23	COM24	
				D2											12H	COM24	COM25	COM26	COM27	COM28	COM29	COM30	COM31
				D3											13H	COM32	COM33	COM34	COM35	COM36	COM37	COM38	COM39
				D4											14H	COM40	COM41	COM42	COM43	COM44	COM45	COM46	COM47
				D5											15H	COM48	COM49	COM50	COM51	COM52	COM53	COM54	COM55
				D6											16H	COM56	COM57	COM58	COM59	COM60	COM61	COM62	COM63
				D7											17H	COM64	COM65	COM66	COM67	COM68	COM69	COM70	COM71
0	0	1	0	D0	PAGE3	18H	COM18	COM19	COM20	COM21	COM22	COM23	COM24	COM25									
				D1											19H	COM26	COM27	COM28	COM29	COM30	COM31	COM32	
				D2											1AH	COM32	COM33	COM34	COM35	COM36	COM37	COM38	COM39
				D3											1BH	COM40	COM41	COM42	COM43	COM44	COM45	COM46	COM47
				D4											1CH	COM48	COM49	COM50	COM51	COM52	COM53	COM54	COM55
				D5											1DH	COM56	COM57	COM58	COM59	COM60	COM61	COM62	COM63
				D6											1EH	COM64	COM65	COM66	COM67	COM68	COM69	COM70	COM71
				D7											1FH	COM72	COM73	COM74	COM75	COM76	COM77	COM78	COM79
0	0	1	1	D0	PAGE4	20H	COM20	COM21	COM22	COM23	COM24	COM25	COM26	COM27									
				D1											21H	COM28	COM29	COM30	COM31	COM32	COM33	COM34	
				D2											22H	COM34	COM35	COM36	COM37	COM38	COM39	COM40	COM41
				D3											23H	COM42	COM43	COM44	COM45	COM46	COM47	COM48	COM49
				D4											24H	COM50	COM51	COM52	COM53	COM54	COM55	COM56	COM57
				D5											25H	COM58	COM59	COM60	COM61	COM62	COM63	COM64	COM65
				D6											26H	COM66	COM67	COM68	COM69	COM70	COM71	COM72	COM73
				D7											27H	COM74	COM75	COM76	COM77	COM78	COM79	COM80	COM81
0	1	0	0	D0	PAGE5	28H	COM28	COM29	COM30	COM31	COM32	COM33	COM34	COM35									
				D1											29H	COM36	COM37	COM38	COM39	COM40	COM41	COM42	
				D2											2AH	COM42	COM43	COM44	COM45	COM46	COM47	COM48	COM49
				D3											2BH	COM50	COM51	COM52	COM53	COM54	COM55	COM56	COM57
				D4											2CH	COM58	COM59	COM60	COM61	COM62	COM63	COM64	COM65
				D5											2DH	COM66	COM67	COM68	COM69	COM70	COM71	COM72	COM73
				D6											2EH	COM74	COM75	COM76	COM77	COM78	COM79	COM80	COM81
				D7											2FH	COM82	COM83	COM84	COM85	COM86	COM87	COM88	COM89
0	1	0	1	D0	PAGE6	30H	COM30	COM31	COM32	COM33	COM34	COM35	COM36	COM37									
				D1											31H	COM38	COM39	COM40	COM41	COM42	COM43	COM44	
				D2											32H	COM44	COM45	COM46	COM47	COM48	COM49	COM50	COM51
				D3											33H	COM52	COM53	COM54	COM55	COM56	COM57	COM58	COM59
				D4											34H	COM60	COM61	COM62	COM63	COM64	COM65	COM66	COM67
				D5											35H	COM68	COM69	COM70	COM71	COM72	COM73	COM74	COM75
				D6											36H	COM76	COM77	COM78	COM79	COM80	COM81	COM82	COM83
				D7											37H	COM84	COM85	COM86	COM87	COM88	COM89	COM90	COM91
0	1	1	1	D0	PAGE7	38H	COM38	COM39	COM40	COM41	COM42	COM43	COM44	COM45									
				D1											39H	COM46	COM47	COM48	COM49	COM50	COM51	COM52	
				D2											3AH	COM52	COM53	COM54	COM55	COM56	COM57	COM58	COM59
				D3											3BH	COM60	COM61	COM62	COM63	COM64	COM65	COM66	COM67
				D4											3CH	COM68	COM69	COM70	COM71	COM72	COM73	COM74	COM75
				D5											3DH	COM76	COM77	COM78	COM79	COM80	COM81	COM82	COM83
				D6											3EH	COM84	COM85	COM86	COM87	COM88	COM89	COM90	COM91
				D7											3FH	COM92	COM93	COM94	COM95	COM96	COM97	COM98	COM99
Column Address				ADC				81H	82H	83H													
				D0="1"	D0="0"																		
LCD OUT				SEG0	00H																		
				SEG1	01H																		
				SEG2	02H																		
LCD OUT				SEG129	02H																		
				SEG130	01H																		
				SEG131	00H																		

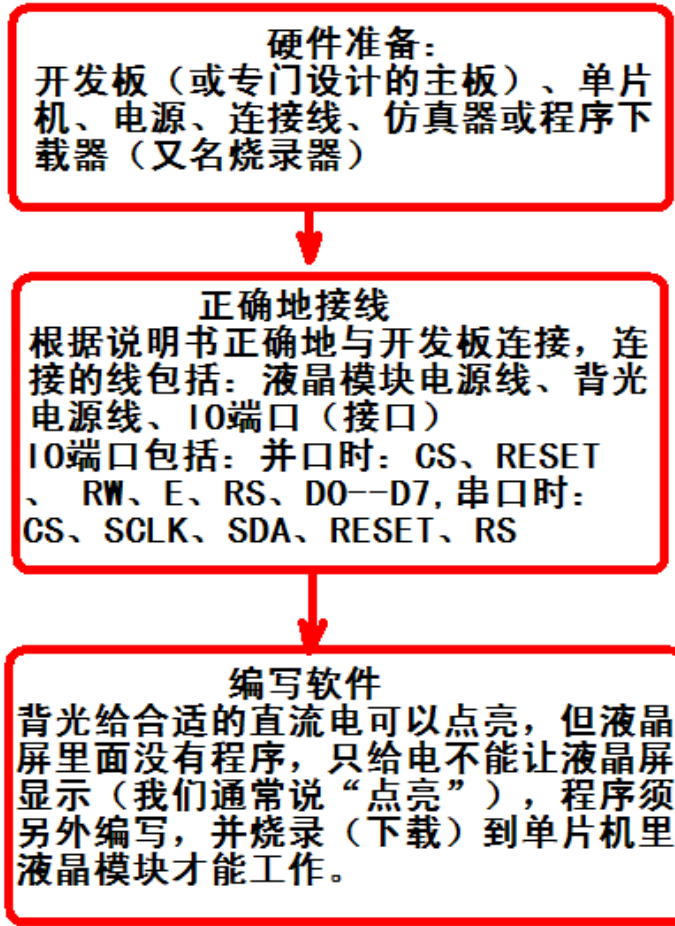


OUTPUT
COM0
COM1
COM2
COM3
COM4
COM5
COM6
COM7
COM8
COM9
COM10
COM11
COM12
COM13
COM14
COM15
COM16
COM17
COM18
COM19
COM20
COM21
COM22
COM23
COM24
COM25
COM26
COM27
COM28
COM29
COM30
COM31
COM32
COM33
COM34
COM35
COM36
COM37
COM38
COM39
COM40
COM41
COM42
COM43
COM44
COM45
COM46
COM47
COM48
COM49
COM50
COM51
COM52
COM53
COM54
COM55
COM56
COM57
COM58
COM59
COM60
COM61
COM62
COM63

7.3 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



7.4 程序举例:

OLED 模块与 MPU (以 8051 系列单片机为例) 接口图如下:

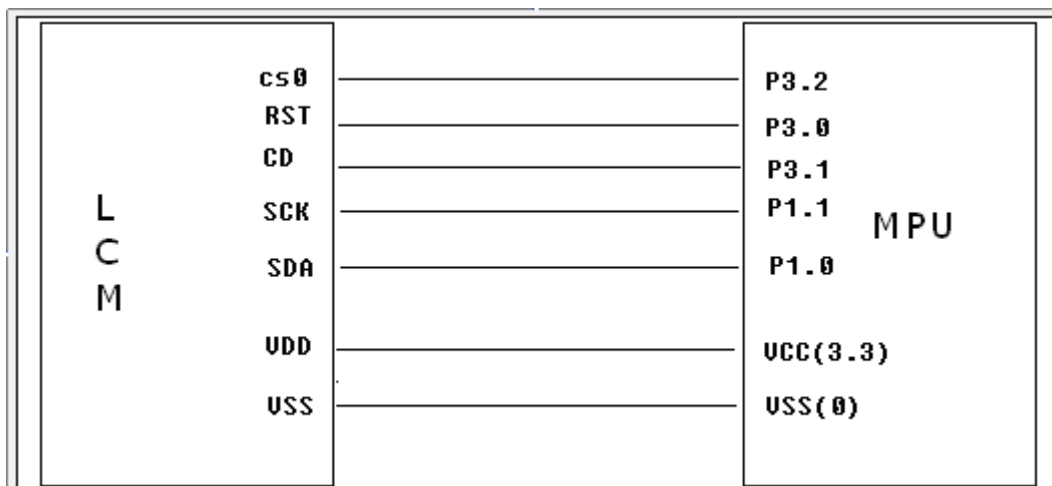


图 5.串行接口

7.4.1 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

//延时
void delay_ms(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_ms(1500);
}

//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()

```

```

{
    lcd_reset=0;    //低电平复位
    delay_ms(500);
    lcd_reset=1;    //复位完毕
    delay_ms(200);
    transfer_command(0xAE);    /*display off*/
    transfer_command(0x02);    /*set lower column address*/
    transfer_command(0x10);    /*set higher column address*/
    transfer_command(0x40);    /*set display start line*/
    transfer_command(0xB0);    /*set page address*/
    transfer_command(0x81);    /*contrast control*/
    transfer_command(0xff);
    transfer_command(0xA1);    /*set segment remap*/
    transfer_command(0xA6);    /*normal / reverse*/
    transfer_command(0xA8);    /*multiplex ratio*/
    transfer_command(0x3F);    /*duty = 1/64*/
    transfer_command(0xad);    /*set charge pump enable*/
    transfer_command(0x8b);    /* 0x8B 内供 VCC */
    transfer_command(0x33);    /*0X30—0X33 set VPP 9V */
    transfer_command(0xC8);    /*Com scan direction*/
    transfer_command(0xD3);    /*set display offset*/
    transfer_command(0x00);    /* 0x20 */
    transfer_command(0xD5);    /*set osc division*/
    transfer_command(0x80);
    transfer_command(0xD9);    /*set pre-charge period*/
}

```



```

transfer_command(0x1f); /*0x22*/
transfer_command(0xDA); /*set COM pins*/
transfer_command(0x12);
transfer_command(0xdb); /*set vcomh*/
transfer_command(0x40);
transfer_command(0xAF); /*display ON*/
}
    
```

```
void lcd_address(uchar page,uchar column)
```

```

{
    column=column+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);
        }
    }
}
    
```

```
//full display test
```

```

void full_display(uchar data1,uchar data2)
{
    int i, j;
    for(i=0;i<8;i++)
    {
        lcd_address(i+1,1);
        for(j=0;j<64;j++)
        {
            transfer_data(data1);
            transfer_data(data2);
        }
    }
}
    
```

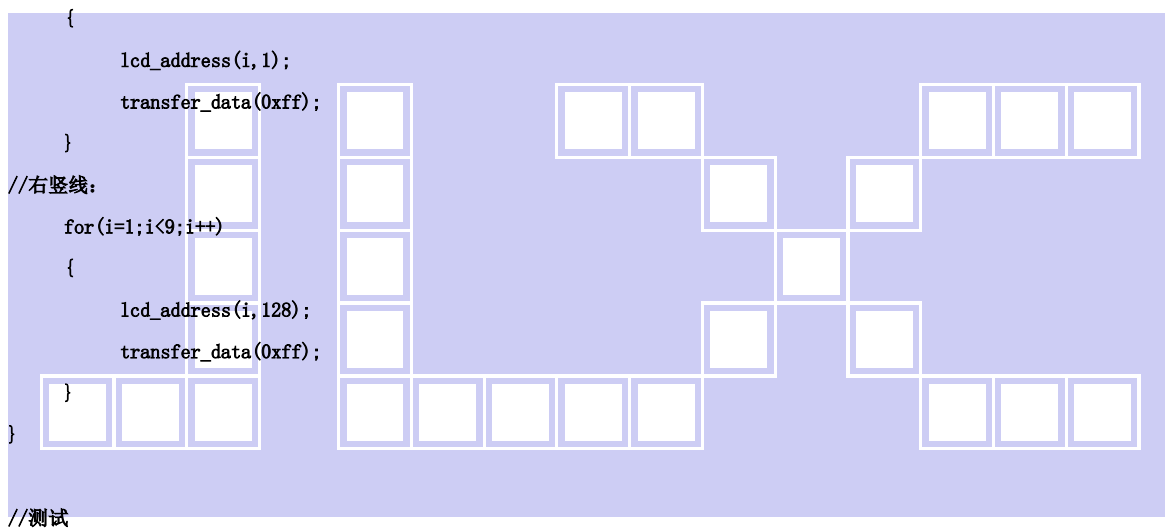
```
//测试外框是否缺划（少行、少列）
```



```

void test_box()
{
    int i;
//上横线:
    for(i=1;i<129;i++)
    {
        lcd_address(1, i);
        transfer_data(0x01);
    }
//下横线:
    for(i=1;i<129;i++)
    {
        lcd_address(8, i);
        transfer_data(0x80);
    }
//左竖线:
    for(i=1;i<9;i++)
    {

```



```

//测试
void test()
{
    test_box();
    waitkey();
    full_display(0xff, 0xff);
    waitkey();
    full_display(0x55, 0x55);
    waitkey();
    full_display(0xaa, 0xaa);
    waitkey();
    full_display(0xff, 0x00);
    waitkey();
    full_display(0x00, 0xff);
    waitkey();
    full_display(0x55, 0xaa);
    waitkey();
    full_display(0xaa, 0x55);

```



```

        waitkey();
    }

//显示 128x64 点阵图像
void display_128x64(uchar *dp)
{
    uint i, j;
    for(j=0; j<8; j++)
    {
        lcd_address(j+1, 1);
        for (i=0; i<128; 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 32x32 点阵图像、汉字、生僻字或 32x32 点阵的其他图标

```

void display_graphic_32x32(uchar page, uchar column, uchar *dp)
{
    uchar i, j;
    for(j=0; j<4; j++)
    {
        lcd_address(page+j, column);
        for (i=0; i<32; 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 16x16 点阵图像、汉字、生僻字或 16x16 点阵的其他图标

```

void display_graphic_16x16(uchar page, uchar column, uchar *dp)
{
    uchar i, j;
    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++;
        }
    }
}

```

//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 8x16 的点阵的字符串，括号里的参数分别为（页，列，字符串指针）

```
void display_string_8x16(uint page,uint column,uchar *text)
```

```
{
    uint i=0,j,k,n;
    if(column>123)
    {
        column=1;
        page+=2;
    }
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<=0x7e))
        {
            j=text[i]-0x20;
            for(n=0;n<2;n++)
            {
                lcd_address(page+n,column);
                for(k=0;k<8;k++)
                {
                    transfer_data(ascii_table_8x16[j][k+8*n]); //写数据到 LCD, 每写完 1 字节的数据后列地址自动加 1
                }
            }
            i++;
            column+=8;
        }
        else
            i++;
    }
}
```



//显示 5x8 的点阵的字符串，括号里的参数分别为（页，列，字符串指针）

```
void display_string_5x8(uint page,uint column,uchar reverse,uchar *text)
```

```
{
    uint i=0,j,k,disp_data;
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<=0x7e))
        {
            j=text[i]-0x20;
            lcd_address(page,column);
            for(k=0;k<5;k++)
            {
                if(reverse==1)
                {
                    disp_data=~ascii_table_5x8[j][k];
                }
                else
                {
                    disp_data=ascii_table_5x8[j][k];
                }
                transfer_data(disp_data); //写数据到LCD,每写完1字节的数据后列地址自动加1
            }
            if(reverse==1) transfer_data(0xff); //写入一列空白列,使得5x8的字符与字符之间有一列间隔,更美观
            else transfer_data(0x00); //写入一列空白列,使得5x8的字符与字符之间有一列间隔,更美观
            i++;
            column+=6;
            if(column>123)
            {
                column=1;
                page++;
            }
        }
        else
        {
            i++;
        }
    }
}
```

//写入一组16x16点阵的汉字字符串（字符串表格中需含有此字）

//括号里的参数：（页，列，汉字字符串）

```
void display_string_16x16(uchar page,uchar column,uchar *text)
```

```
{
    uchar i,j,k;
    uint address;

    j = 0;
    while(text[j] != '\0')
    {
```

```

i = 0;
address = 1;
while(Chinese_text_16x16[i] > 0x7e)    // >0x7f 即说明不是 ASCII 码字符
{
    if(Chinese_text_16x16[i] == text[j])
    {
        if(Chinese_text_16x16[i + 1] == text[j + 1])
        {
            address = i * 16;
            break;
        }
    }
    i += 2;
}

```

```

if(column > 113)
{

```

```

    column = 0;
    page += 2;
}

```

```

if(address != 1) // 显示汉字
{

```

```

    for(k=0;k<2;k++)
    {

```

```

        lcd_address(page+k, column);

```

```

        for(i = 0; i < 16; i++)
        {

```

```

            transfer_data(Chinese_code_16x16[address]);

```

```

            address++;
        }
    }
}
j += 2;
}
else //显示空白字符
{

```

```

    for(k=0;k<2;k++)
    {

```

```

        lcd_address(page+k, column);

```

```

        for(i = 0; i < 16; i++)
        {

```

```

            transfer_data(0x00);
        }
    }
}

```

```

j++;

```

```

    }

    column+=16;
}
}

//显示 16x16 点阵的汉字或者 ASCII 码 8x16 点阵的字符混合字符串
//括号里的参数：(页，列，字符串)
void disp_string_8x16_16x16(uchar page,uchar column,uchar *text)
{
    uchar temp[3];
    uchar i = 0;

    while(text[i] != '\0')
    {
        if(text[i] > 0x7e)
        {
            temp[0] = text[i];
            temp[1] = text[i + 1];
            temp[2] = '\0'; //汉字为两个字节
            display_string_16x16(page,column,temp); //显示汉字
            column += 16;
            i += 2;
        }
        else
        {
            temp[0] = text[i];
            temp[1] = '\0'; //字母占一个字节
            display_string_8x16(page, column, temp); //显示字母
            column += 8;
            i++;
        }
    }
}

void main(void)
{
    while(1)
    {
        initial_lcd(); //初始化
        clear_screen(); //清屏

        //演示 32x32 点阵的汉字，16x16 点阵的汉字，8x16 点阵的字符，5x8 点阵的字符
        display_string_5x8(1,1,0,"{(5x8dot ASCII char)}"); //显示字符串，括号里的参数分别为 (PAGE,列,字符串指针)
        display_string_5x8(2,1,0,"[(<~!@#%&*+=?>>)]");
        disp_string_8x16_16x16(3,1,"标准 16x16dot 汉字"); //显示 16x16 点阵汉字串或 8x16 点阵的字符串，括号里的参数分别为 (页,列,字符串指针)
    }
}

```



```

display_graphic_32x32 (5, 1+32*0, jing1); //显示单个 32x32 点阵的汉字，括号里的参数分别为 (PAGE, 列, 字符
指针)

display_graphic_32x32 (5, 1+32*1, lian1);
display_graphic_32x32 (5, 1+32*2, xun1);
disp_string_8x16_16x16(5, 1+32*3, "JLX:");
disp_string_8x16_16x16(7, 1+32*3, "OLED");
waitkey();

//演示显示一页纯英文的 5x8 点阵的菜单界面

clear_screen(); //clear all dots
display_string_5x8(1, 1, 1, "012345678901234567890");
display_string_5x8(1, 1, 1, "MENU"); //显示 5x8 点阵的字符串，括号里的参数分别为 (页, 列, 是否反显, 数据指针)
display_string_5x8(3, 1, 0, "Select>>>>");
display_string_5x8(3, 64, 1, "1. Graphic ");
display_string_5x8(4, 64, 0, "2. Chinese ");
display_string_5x8(5, 64, 0, "3. Movie ");
display_string_5x8(6, 64, 0, "4. Contrast");
display_string_5x8(7, 64, 0, "5. Mirror ");
display_string_5x8(8, 1, 1, "PRE USER DEL NEW");
display_string_5x8(8, 19, 0, " ");
display_string_5x8(8, 65, 0, " ");
display_string_5x8(8, 97, 0, " ");
waitkey();
clear_screen(); //clear all dots
display_128x64 (bmp1);
waitkey();
clear_screen(); //clear all dots
display_128x64 (bmp2);
waitkey();
clear_screen(); //clear all dots
test();
}
}

```



-END-