JLX12864G-928 使用说明书

目 录

1．概述

晶联讯电子专注于液晶屏及液晶模块的研发、制造。所生产 JLX12864G-928 型液晶模块由于使用方便、显示清晰，广泛应用于各种人机交流面板。

JLX12864G-928 可以显示 128 列*64 行点阵单色图片，或显示 8 个/行*4 行 16*16 点阵的汉字，或显示 16 个/行*8 行 8*8 点阵的英文、数字、符号。

2．JLX12864G-928 图像型点阵液晶模块的特性

2.1 结构牢：背光带有挡墙，焊接式 FPC。

2.2 IC 采用矽创公司 ST7567,功能强大，稳定性好

2.3 功耗低:1～100mW（关掉背光：0.3mA@3.3V,打开背光不大于 100mW）；

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。按照 12*12 点阵汉字来计算可显示 10 字/行*4 行。

2.5 指令功能强:可软件调对比度、正显/反显转换、行列扫描方向可改（可旋转 180 度使用）。”;

2.6 接口简单方便:采用 4 线 SPI 串口。

2.7 工作温度宽:-10℃ - 60℃，储存温度:-20℃ - 70℃;

3. 外形尺寸及接口引脚功能

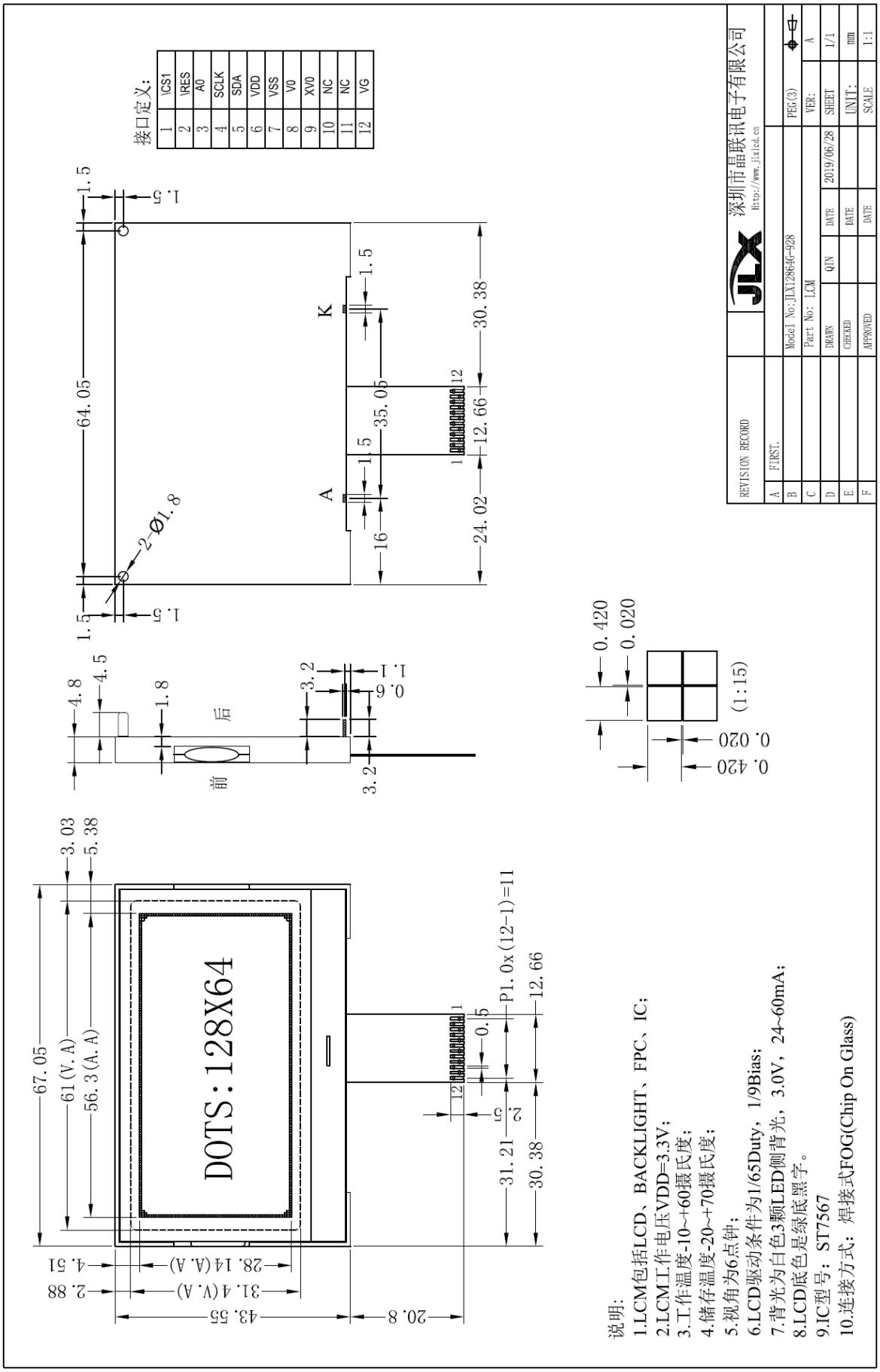


图 1. 外形尺寸

模块的接口引脚功能

引 线 号	符 号	名 称	功 能
1	CS	片选	低电平片选
2	RES	复位	低电平复位，复位完成后，回到高电平，液晶模块开始工作
3	RS(A0)	寄存器选择信号	H:数据寄存器 0:指令寄存器
4	SCLK	I/O	串行时钟(SCLK)
5	SDA	I/O	串行数据(SDA)
6	VDD	供电电源正极	供电电源正极
7	VSS	接地	0V
8	V0	倍压电路	
9	XV0	倍压电路	
10	NC	空脚	
11	NC	空脚	
12	VG	偏置电压	

表 1：模块的接口引脚功能

4. 基本原理

4.1 液晶屏（LCD）

在 **LCD** 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）.

4.2 内部电路框图：

图 2 是 JLX12864G-928 图像点阵型模块的电路框图：

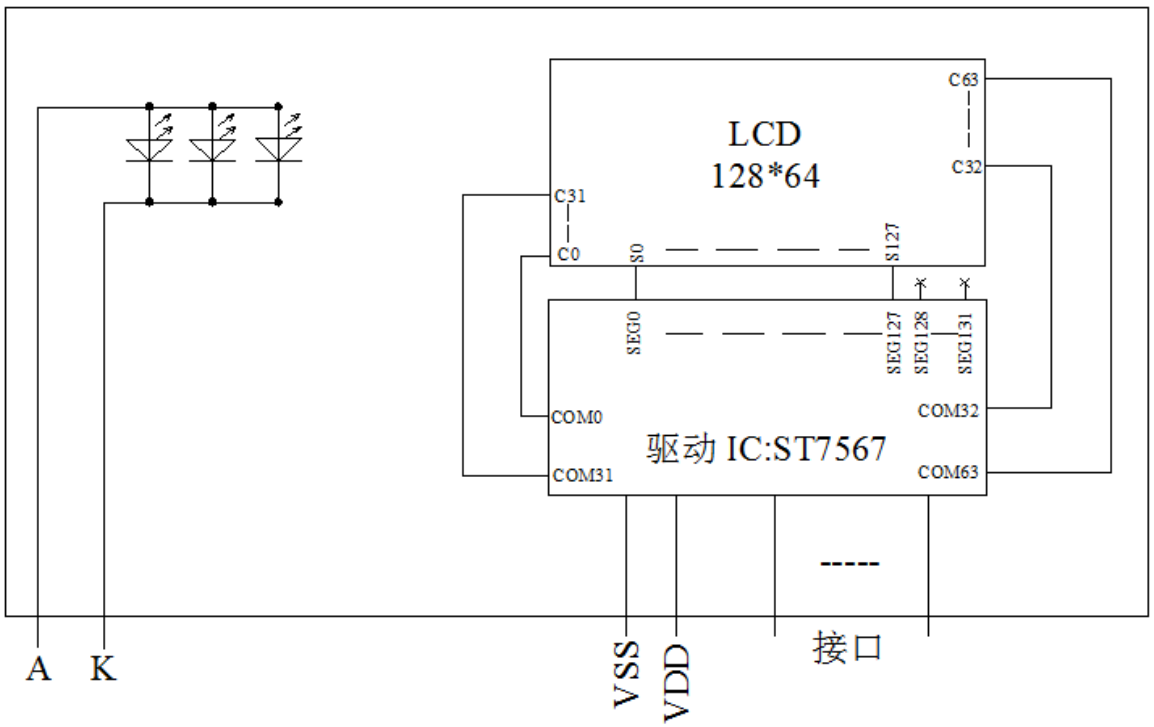


图 2:JLX12864G-928 图像点阵型液晶模块的电路框图

4.2 背光参数

该型号液晶模块带 LED 背光源。它的性能参数如下:
背光板选择绿色。
正常工作电流为：24∽45mA（LED 灯数共 3 颗）;
工作电压：3.0V;

5. 技术参数

5.1 最大极限参数（超过极限参数则会损坏液晶模块）

名称	符号	标准值			单位
		最小	典型	最大	
电路电源	VDD - VSS	-0.3		3.6	V
LCD 驱动电压	V0、VOUT	-0.3		13.5	V
LCD 驱动电压	V1\V2\V3\V4	-0.3		V0	V
工作温度		-10		+60	℃
储存温度		-20		+70	℃

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	3.6	V
背光工作电压	VLED		2.9	3.0	3.1	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} = -0.5mA	0.8xVDD	-	VDD	V
输出低电平	V_{OHC}	I_{OL} = -0.5mA	VSS	-	0.2xVDD	V
模块工作电流	I_{DD}	VDD = 3.3V	-		0.3	mA
背光工作电流	I_{LED}	V_{LED}=3.0V	24	45	60	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 ST7567（Writing Data from CPU to ST7567）

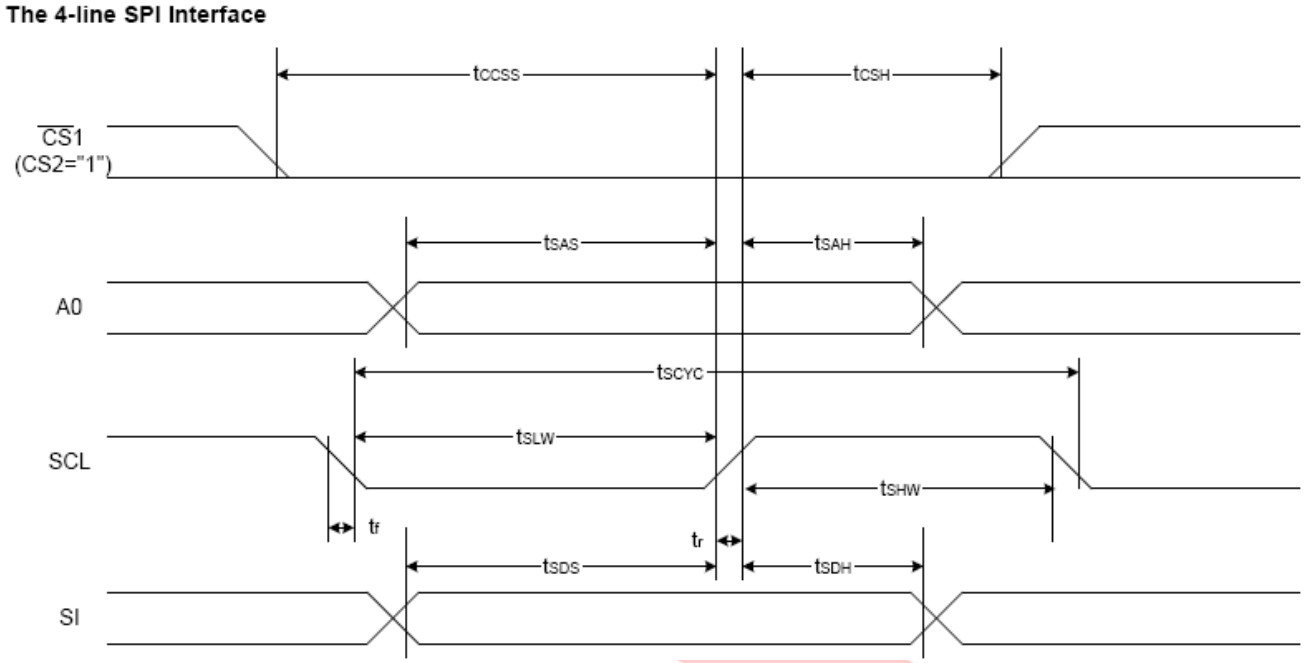


图 4. 从 CPU 写到 ST7567（Writing Data from CPU to ST7567）

6.2 串行接口：时序要求（AC 参数）：

写数据到 ST7567 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）	Tscyc	引脚：SCK	50	--	25	ns
保持SCK高电平脉宽 （SCK “H” pulse width）	Tshw	引脚：SCK	25			ns
保持SCK低电平脉宽 （SCK “L” pulse width）	TSLW	引脚：SCK	25			ns
地址建立时间 (Address setup time)	TSAS	引脚： RS	20	--	--	ns
地址保持时间 (Address hold time)	Tsah	引脚：RS	10	--	--	ns
数据建立时间 (Data setup time)	Tsds	引脚：SI	20	--	--	ns
数据保持时间 (Data hold time)	TSDH	引脚：SI	10	--	--	ns
片选信号建立时间 （CS-SCL time）	Tcss	引脚：CS	20			ns

片选信号保持时间（CS-SCL time）	T_{csh}	引脚：CS	40			ns

VDD =3.0V±5%,Ta = 25℃

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

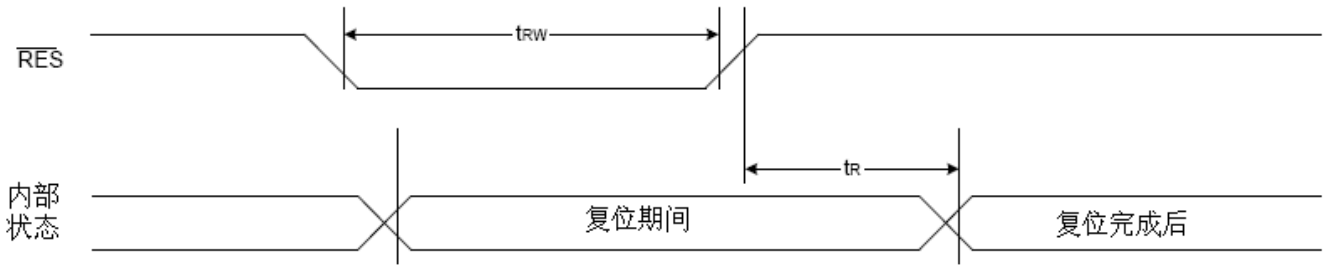


图 7：**电源启动后复位的时序**

表 6：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		--	--	3.0	us
复位保持低电平的时间	t_{RW}	引脚：RES	3.0	--	--	us

7. 指令功能：

7.1 指令表

指 令 表 表 8.

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，针对该液晶屏一般设置为 0x60
(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，本液晶屏没有这一行图标，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。比如液晶模块的第 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
(4) 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	并口时：读驱动 IC 的当前状态,串口时不能用此指令
(6)写显示数据到液晶屏 (Display data write)	1	8 位显示数据								从 CPU 写数据到液晶屏，每一位对应一个点阵，1 个字节对应 8 个竖置的点阵
(7)读液晶屏的显示数据 (Display data read)	1	8 位显示数据								并口时：读已经显示到液晶屏上的点阵数据。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：常规：列地址从左到右， 0xA1：反转：列地址从右到左
(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)LCD 偏压比设置 (LCD bias set)	0	1	0	1	0	0	0	1	0 1	设置偏压比： 0XA2：BIAS=1/9 (常用) 0XA3：BIAS=1/7
(12) 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43-44页
(13) 退出上述“读-改-写”指令(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 43-44 页
(14) 软件复位（Reset）	0	1	1	1	0	0	0	1	0	0XE2 :软件复位。

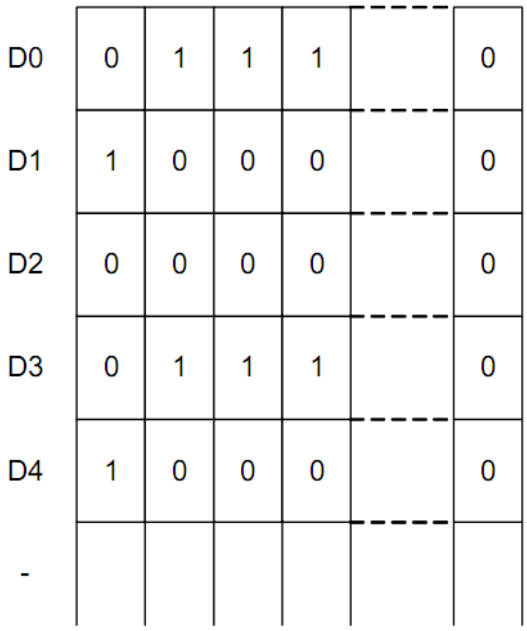
(15) 行扫描顺序选择 (Common output mode select)			1	1	0	0	0 1	0	0	0	行扫描顺序选择： **0XC0**:普通扫描顺序：从上到下 **0XC8**:反转扫描顺序：从下到上
(16) 电源控制 (Power control set)			0	0	1	0	1	**电压操作模式选择，共 3 位**			选择内部电压供应操作模式： D2、D1、D0 位分别对应内部升压是否打开(1 为打开，0 为不打开)，电压调整电路是否打开(1 为打开，0 为不打开)，电压跟随器是否打开(1 为打开，0 为不打开)。 通常是 **0x2C,0x2E,0x2F** 三条指令按顺序紧接着写，表示依次打开内部升压、电压调整电路、电压跟随器。也可以单单写 **0x2F**，一次性打开三部分电路。
(17) 选择内部电阻比例		0	0	0	1	0	0	**内部电压值电阻设置**			选择内部电阻比例（Rb/Ra）:可以理解为**粗调**对比度值。可设置范围为：**0x20～0x27，**数值越大对比度越浓，越小越淡
(18)	内部设置液晶电压模式	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：**0x00～0x3F,**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**						
(19)静态图标显示：开/关		0	1	0	1	0	1	1	0	0 1	静态图标的开关设置： **0xAC**: 关, **0xAD**: 开。 此指令在进入及退出睡眠模式时起作用
(20) 升压倍数选择 (Booster ratio set)		0	1	1	1	1	1	0	0	0	选择升压倍数： 00：2 倍，3 倍，4 倍 01：5 倍 11：6 倍。本模块外部已设置升压倍数为 4 倍，不必使用此指令
			0	0	0	0	0	0	2 位数设置升压倍数		
(21) 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 规格书第 47 页“POWER SAVE”
(22)空指令（ NOP）		0	1	1	1	0	0	0	1	1	空操作
(23) 测试（Test）		0	1	1	1	1	*	*	*	*	内部测试用，千万别用！

请详细参考 IC 资料”ST7567.PDF”

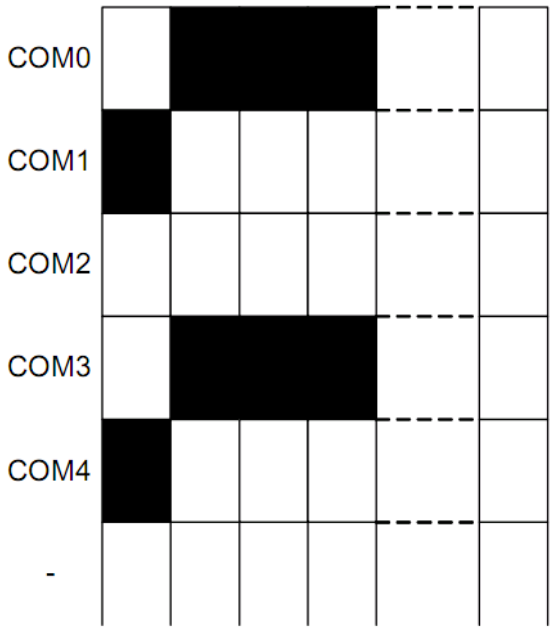
7.3 点阵与 DD RAM(显示数据存储器)地址的对应关系

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：数据是从下向上排列的。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：

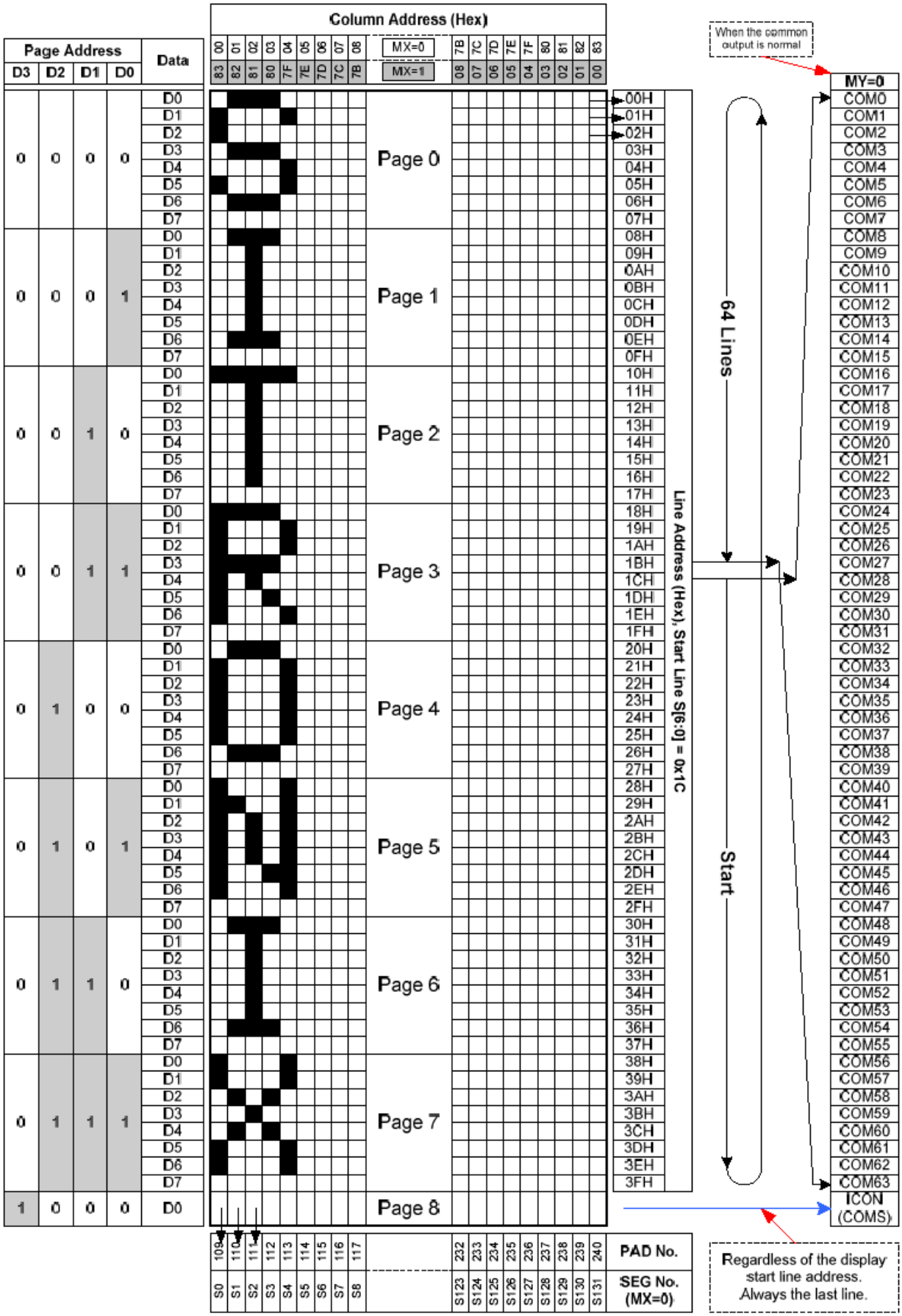


Display data RAM
（显示数据存储器）



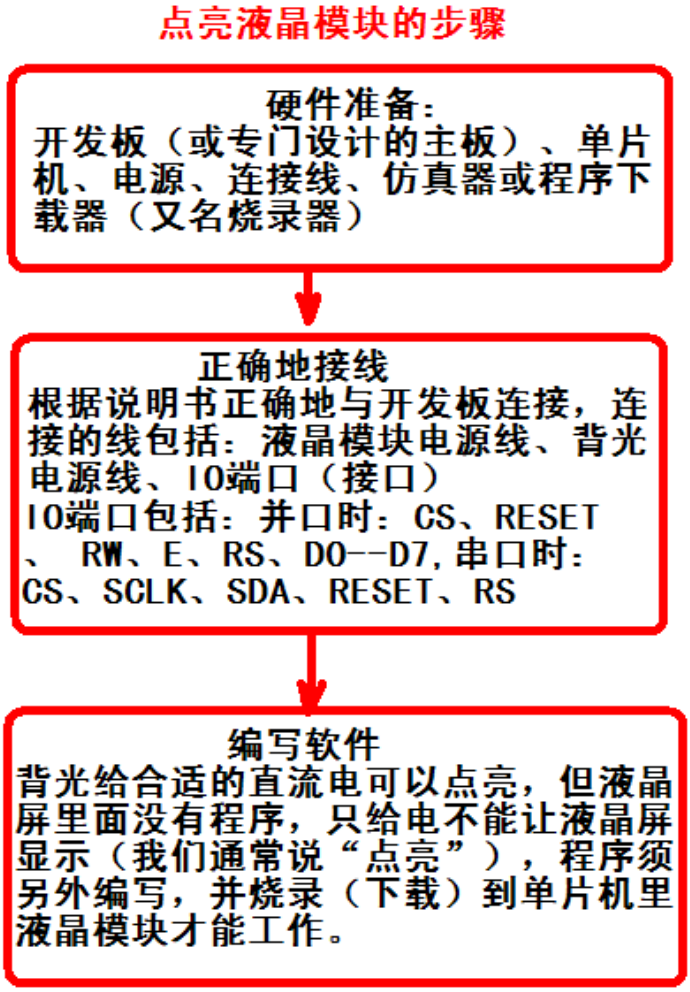
Liquid crystal display
（液晶屏）

DB7--DB0 的排列方向：数据是从下向上排列的。最低位 D0 是在最上面，最高位 D7 是在最下面。下图摘自 ST7567 IC 资料，可通过“ST7567.PDF”获取最佳效果。


Column Address (Hex)
MX=0
MX=1
Page Address
D3 D2 D1 D0
Data
Page 0
Page 1
Page 2
Page 3
Page 4
Page 5
Page 6
Page 7
Page 8
Line Address (Hex), Start Line S[6:0] = 0x1C
64 Lines
Start
When the common output is normal
MY=0
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
ICON (COMS)
PAD No.
SEG No. (MX=0)
Regardless of the display start line address. Always the last line.

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



7.5 程序举例:

液晶模块与 MPU(以 8051 系列单片机为例)接口图如下:

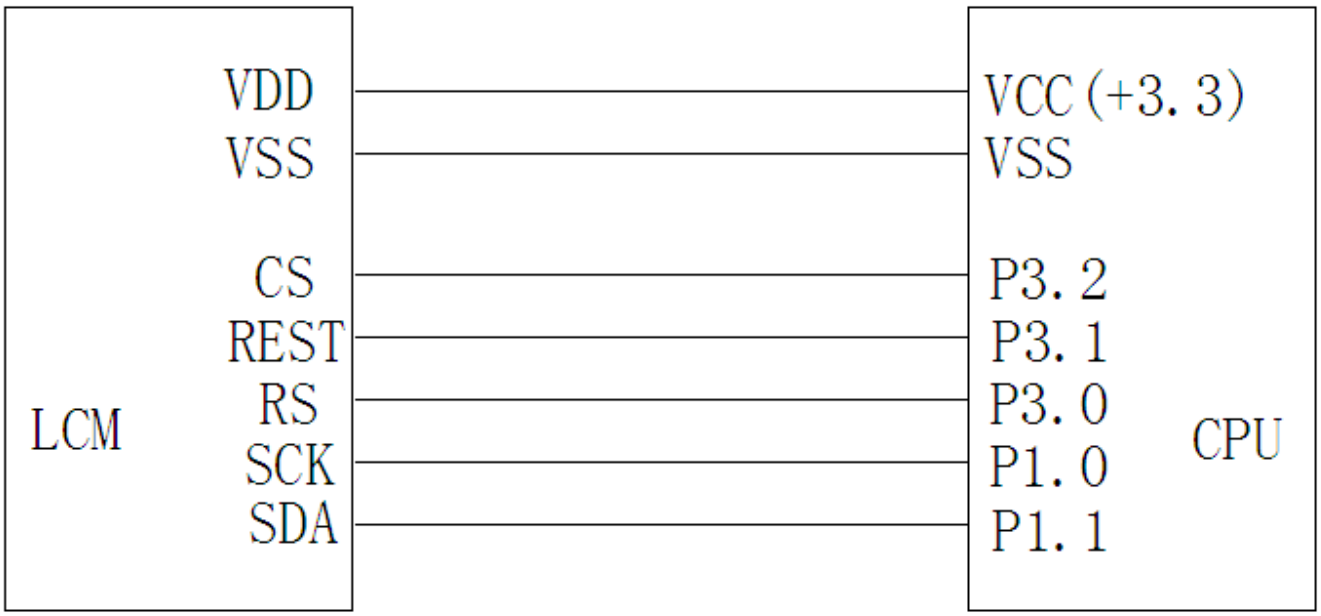
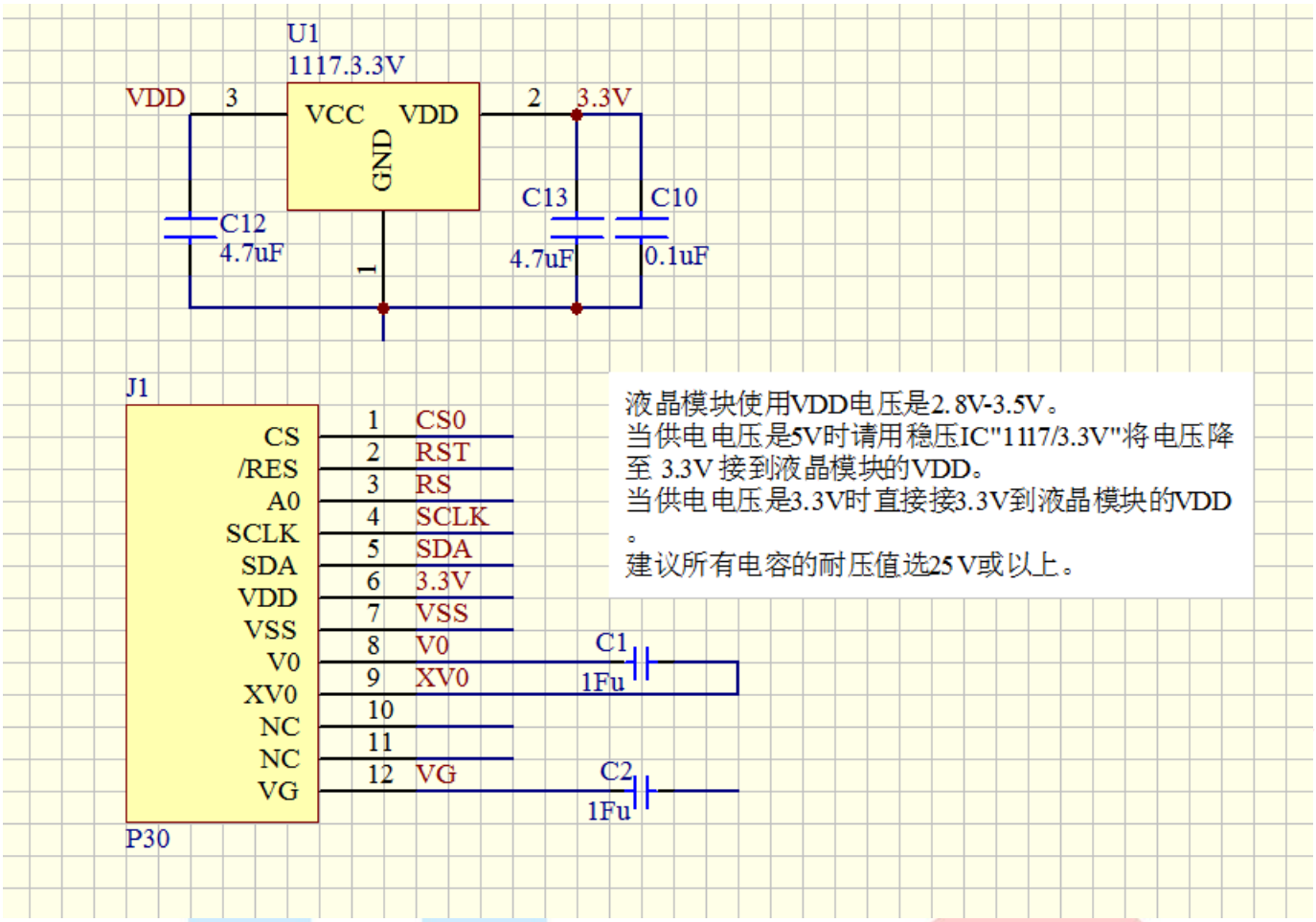
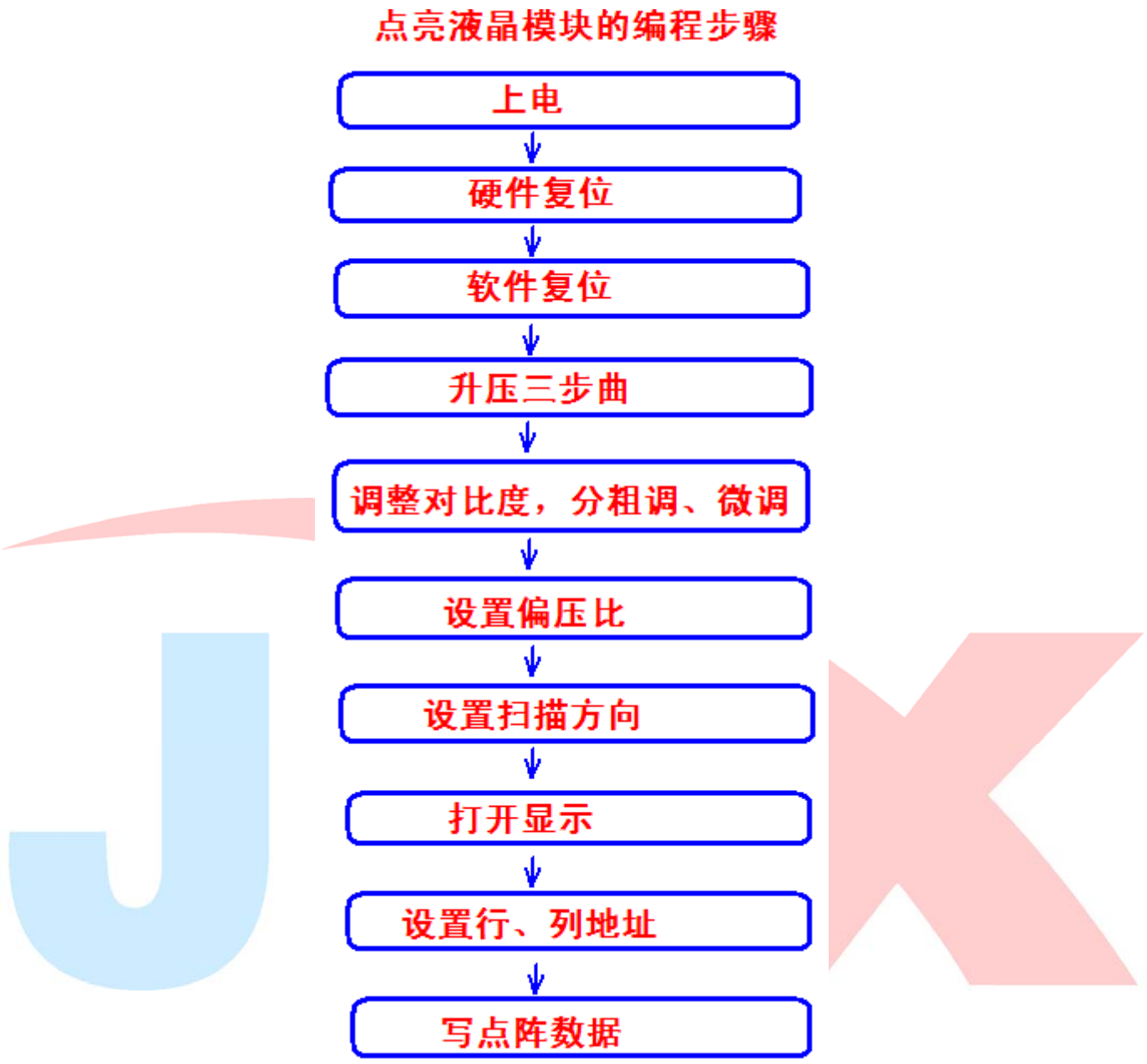


图 9. 串行接口

串行电路图


U1
1117.3.3V
VDD
3
VCC
VDD
GND
2
3.3V
C12
4.7uF
1
C13
4.7uF
C10
0.1uF
J1
CS
/RES
A0
SCLK
SDA
VDD
VSS
V0
XV0
NC
NC
VG
1 CS0
2 RST
3 RS
4 SCLK
5 SDA
6 3.3V
7 VSS
8 V0
9 XV0
10
11
12 VG
C1
1Fu
C2
1Fu
P30
液晶模块使用VDD电压是2.8V-3.5V。
当供电电压是5V时请用稳压IC"1117/3.3V"将电压降至 3.3V 接到液晶模块的VDD。
当供电电压是3.3V时直接接3.3V到液晶模块的VDD。
建议所有电容的耐压值选25 V或以上。

7.51、程序



以下为串行方式的例程序：

```
/* Test program for JLX12864G-928, 串行接口
   驱动 IC 是:ST7567(or compatible)
   晶联讯电子：网址  http://www.jlxlcd.cn; http://www.jlxlcd.com.cn
*/
#include <reg51.H>
#include <intrins.h>
#include <Ctype.h>
#include <math.h>
#include <chinese.h>

sbit cs1=P3^2;          /*接口定义*/
sbit reset=P3^1;   /*接口定义*/
sbit rs=P3^0;      /*接口定义*/
sbit sclk=P1^0;         /*接口定义*/
sbit sid=P1^1;          /*接口定义。另外 P1.0~1.7 对应 DB0~DB7*/
sbit key=P2^0;          /*按键接口，P2.0 口与 GND 之间接一个按键*/


/*延时*/
```

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

/*短延时*/
void delay_us(int i)
{
     int j,k;
     for(j=0;j<i;j++)
          for(k=0;k<1;k++);
}
void waitkey()
{
repeat:  if(key==1)    goto repeat;
         else delay(2000);
}

//=============transfer command to LCM===============
void transfer_command(int data1)
{
     char i;
     cs1=0;
     rs=0;
     for(i=0;i<8;i++)
     {
          sclk=0;
          if(data1&0x80) sid=1;
          else sid=0;
          sclk=1;
          delay_us(1);
          data1<<=1;
     }
     cs1=1;
}

//------------transfer data to LCM---------------
void transfer_data(int data1)
{
     char i;
     cs1=0;
     rs=1;
     for(i=0;i<8;i++)
     {
          sclk=0;
          //delay_us(1);
          if(data1&0x80) sid=1;
          else sid=0;
          sclk=1;
          //delay_us(1);
          data1<<=1;
     }
     cs1=1;
}



/*LCD 模块初始化*/
void initial_lcd()
{
     reset=0;         /*低电平复位*/
     delay(200);
     reset=1;           /*复位完毕*/
     delay(50);
     transfer_command(0xe2); /*软复位*/
     delay(5);
     transfer_command(0x2c);  /*升压步聚 1*/
     delay(5);
     transfer_command(0x2e);  /*升压步聚 2*/
     delay(5);
     transfer_command(0x2f);  /*升压步聚 3*/
     delay(5);

     transfer_command(0x25);  /*粗调对比度，可设置范围 0x20～0x27*/
     transfer_command(0x81);  /*微调对比度*/
     transfer_command(0x1e);  /*微调对比度的值，可设置范围 0x00～0x3f*/
     transfer_command(0xa2);  /*1/9 偏压比（bias）*/

     transfer_command(0xc8);  /*行扫描顺序：从上到下*/
```

```
    transfer_command(0xa0);  /*列扫描顺序：从左到右*/

    transfer_command(0x40);  /*起始行：第一行开始*/
    transfer_command(0xaf);  /*开显示*/
}

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
}

/*
void lcd_address(uchar page,uchar column)
{
    column=column+3;                              //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(i=0;i<9;i++)
    {
        lcd_address(1+i,1);
        for(j=0;j<132;j++)
        {
            transfer_data(0x00);
        }
    }
}

//===显示测试画面：例如全显示，隔行显示，隔列显示，雪花显示=====
void test_display(uchar data1,uchar data2)
{
    int i,j;
    for(j=0;j<8;j++)
    {
        lcd_address(j+1,0);
        for(i=0;i<128;i++)
        {
            transfer_data(data1);
            transfer_data(data2);
        }
    }
}

void display_graphic(uchar *dp)
{
    uchar i,j;
    for(j=0;j<8;j++)
    {
        lcd_address(j+1,1);
        for(i=0;i<128;i++)
        {
            transfer_data(*dp);
            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<31;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 reverse,uchar *dp)
{
    uchar i,j;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j,column);
        for (i=0;i<16;i++)
        {
            if(reverse==1)
            {
                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*/
            }
            else
                transfer_data(*dp);
            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++;
        }
    }
}

//显示一串 8x16 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）
void display_string_8x16(uint page,uint column,uchar reverse,uchar *text)
{
    uint i=0,j,k,n,data1;
    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++)
                {
                    if(reverse==1)    data1=~ascii_table_8x16[j][k+8*n];
                    else data1=ascii_table_8x16[j][k+8*n];
                    transfer_data(data1);
                }
            }
            i++;
            column+=8;

        }
        else
        i++;
    }
}

//显示一串 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）
void display_string_5x8(uint page,uint column,uchar reverse,uchar *text)
{
    uchar i=0,j,k,data1;
    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)     data1=~ascii_table_5x8[j][k];
                    else data1=ascii_table_5x8[j][k];
                    transfer_data(data1);
                }
                if(reverse==1)     transfer_data(0xff);
                else transfer_data(0x00);
                i++;
                column+=6;
            }
            else
            i++;
        }
}



void sleep()
{
    transfer_command(0xac);/*静态图标关闭*/
    transfer_command(0x00);/*静态图标寄存器设置：关闭。此指令与上述指令一起完成静态图标关闭*/
    transfer_command(0xae);/*显示：关*/
    transfer_command(0xa5);/*全屏显示：开*/
}
void wake()
{
    transfer_command(0xa4);/*全屏显示：关。进入正常模式*/
    transfer_command(0xad);/*静态图标开启*/
    transfer_command(0x03);/*静态图标寄存器设置：开。此指令与上述指令一起完成静态图标开启*/
    transfer_command(0xaf);/*显示：开*/
}

void main(void)
{
    initial_lcd();
    while(1)
    {
        clear_screen();
        display_graphic(bmp1);
        waitkey();

        clear_screen();
        display_graphic(bmp2);
        waitkey();

        clear_screen();                                    //clear all dots
        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();
        display_graphic_32x32(1,33,cheng1);           //在第 1 页，第 49 列显示单个汉字"成"*/
        display_graphic_32x32(1,65,gong);               //在第 1 页，第 49 列显示单个汉字"成"*/
        display_graphic_16x16(5,1,1,zhuang1);           //在第 5 页，第 1 列显示单个汉字"状"
        display_graphic_16x16(5,(1+16),1,tai1);       //在第 5 页，第 17 列显示单个汉字"态"
        display_string_8x16(5,33,0,":");              //在第 1 页，第 1 列显示字符串
        display_graphic_16x16(5,41,0,shi1);           //在第 5 页，第 41 列显示单个汉字"使"
        display_graphic_16x16(5,(1+16*3+8),0,yong1);  //在第 5 页，第 49 列显示单个汉字"用"
        display_string_8x16(5,89,0,"00:00");          //显示 8x16 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）
        waitkey();

        clear_screen();                                    //clear all dots
        display_string_8x16(1,1,0,"0123456789abcdef");     //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_8x16(3,1,0,"~`!@#$%^&*()_-+=");     //同上
        display_string_5x8(5,1,1," ! #$%&'()*+,-./01234");
        display_string_5x8(6,1,0,"56789:;<=>?@ABCDEFGHI");
```

```
        display_string_5x8(7,1,0,"JKLMNOPQRSTUVWXYZ[\]^");
        display_string_5x8(8,1,0,"_`abcdefghijklmnopqrs");
        waitkey();
        test_display(0xff,0xff);
        waitkey();
        test_display(0xaa,0x55);
        waitkey();
        test_display(0x55,0xaa);
        waitkey();
        clear_screen();
        display_graphic(bmp3);
        waitkey();
        clear_screen();
        display_graphic(bmp4);
        waitkey();

    }
}
```