JLX12832G-03703 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 JLX12832G-03703 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。

JLX12832G-03703 可以显示 128 列*32 行点阵单色图片，或显示 16*16 点阵的汉字 8 个*2 行，或显示 8*16 点阵的英文、数字、符号 16 个*2 行。或显示 5*8 点阵的英文、数字、符号 21 个*4 行。

2．JLX12832G-03703 图像型点阵液晶模块的特性

2.1 结构牢：背光带有挡墙，插式 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*32 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*2 行。按照 12*12 点阵汉字来计算可显示 10 字/行*2 行。

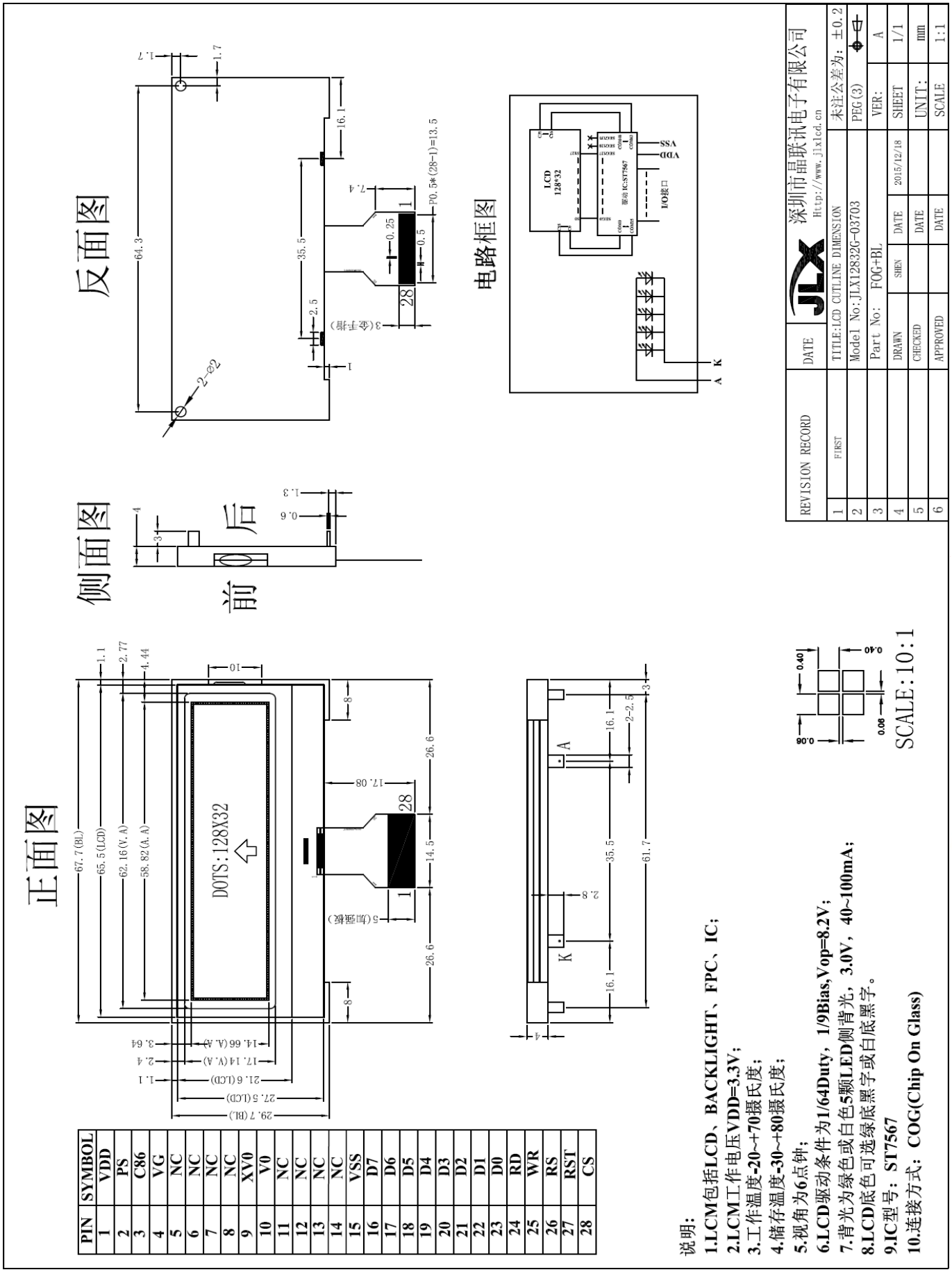
2.5 指令功能强:可软件调对比度、正显/反显转换、行列扫描方向可改（可旋转 180 度使用）。

2.6 接口简单方便:可选并行或采用 4 线 SPI 串口。

2.7 工作温度:-20℃ – 70℃;

2.8 储存温度:-30℃ – 80℃;

3. 外形尺寸及接口引脚功能



PIN	SYMBOL
1	VDD
2	PS
3	C86
4	VG
5	NC
6	NC
7	NC
8	XV0
9	V0
10	NC
11	NC
12	NC
13	NC
14	NC
15	VSS
16	D7
17	D6
18	D5
19	D4
20	D3
21	D2
22	D1
23	D0
24	RD
25	WR
26	RS
27	RST
28	CS

说明：

1.LCM包括LCD、BACKLIGHT、FPC、IC；
2.LCM工作电压VDD=3.3V；
3.工作温度-20~+70摄氏度；
4.储存温度-30~+80摄氏度；
5.视角为6点钟；
6.LCD驱动条件为1/64Duty，1/9Bias,Vop=8.2V；
7.背光为绿色或白色5颗LED侧背光，3.0V，40~100mA；
8.LCD底色可选绿底黑字或白底黑字。
9.IC型号：ST7567
10.连接方式：COG(Chip On Glass)

图 1. 外形尺寸

模块的接口引脚功能

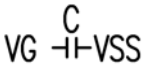
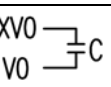
引 线 号	符 号	名 称	功 能
1	VDD	供电电源正极	供电电源正极
2	PS	选串并控制接口	接 VDD:选择并行接口， 接 VSS:选择串行接口
3	C86	选择 6800 或 8080	并行接口时：H:6800 系统,L:8080 系统。 串行接口时：接 VDD
4	VG	升压输出	
5	NC	空脚	空脚
6	NC	空脚	空脚
7	NC	空脚	空脚
8	NC	空脚	空脚
9	XV0	偏置电压	
10	V0	偏置电压	
11	NC	空脚	空脚
12	NC	空脚	空脚
13	NC	空脚	空脚
14	NC	空脚	空脚
15	VSS	接地	0V
16	D7（SDA）	I/O	并行接口时：数据总线 DB7 串行接口时：串行数据(SDA)
17	D6(SCLK)	I/O	并行接口时：数据总线 DB6 串行接口时：串行时钟(SCLK)
18	D5	I/O	并行接口时：数据总线 DB0~DB5 串行接口时：空脚
19	D4	I/O	
20	D3	I/O	
21	D2	I/O	
22	D1	I/O	
23	D0	I/O	
24	RD(E)	使能信号,或“读”	并行接口时并且选择 6800 时序时：使能信号,高电平有效. 并行接口时并且选择 8080 时序时：读数据,低电平有效. 串行接口时：空
25	WR(R/W)	读/写,或“写”	并行接口时并且选择 6800 时序时：H:读数据 L:写数据 并行接口时并且选择 8080 时序时：写数据,低电平有效. 串行接口时：空
26	RS	寄存器选择信号	H:数据寄存器 0:指令寄存器
27	RST	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
28	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 液晶屏（LCD）

在 LCD 上排列着 128×32 点阵,128 个列信号与驱动 IC 相连，32 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：

电路框图

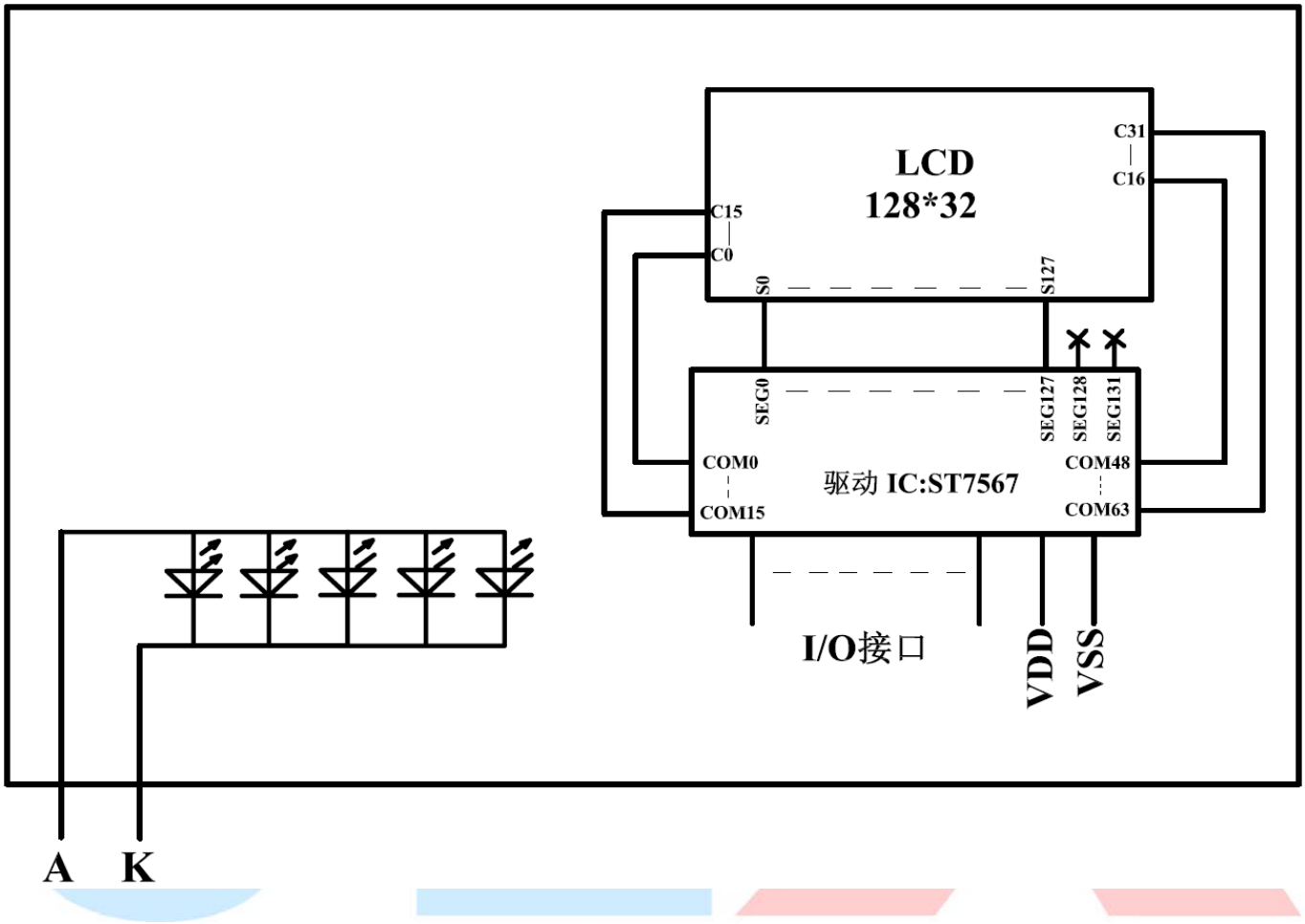


图 2:JLX12832G-03703 图像点阵型液晶模块的电路框图

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:
背光板可选择绿色、白色。
正常工作电流为：40∽100mA（LED 灯数共 5 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		-20		+70	℃
储存温度		-30		+80	℃

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	-	3.6	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	40	75	100	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）

System Bus Timing for 4-Line Serial Interface

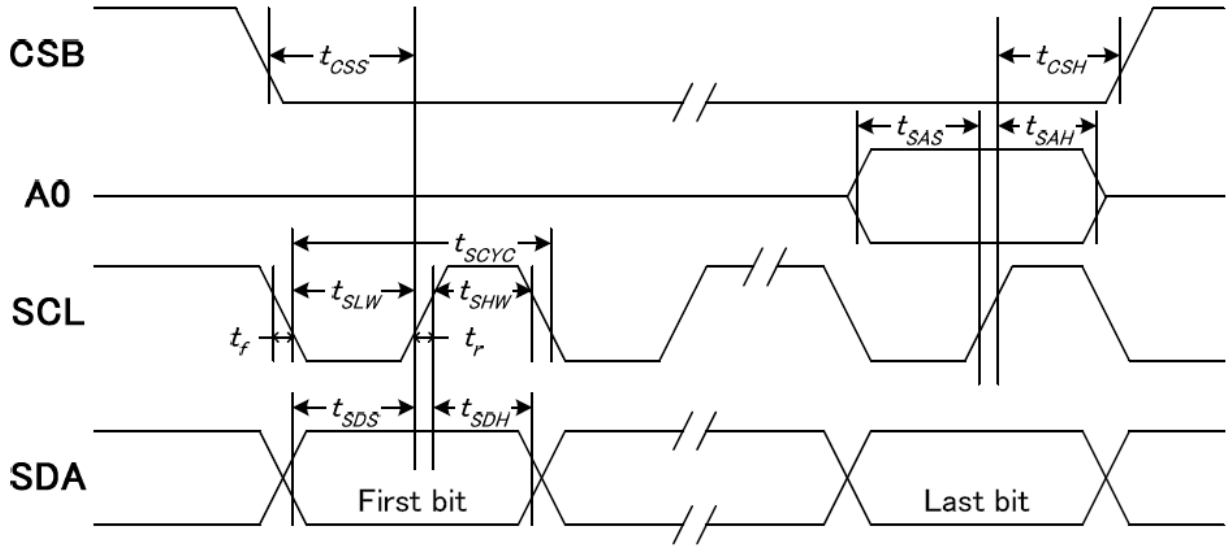


图 4. **从 CPU 写到 ST7567（Writing Data from CPU to ST7565R）**

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：CS0	20			ns
片选信号保持时间 （CS-SCL time）	Tcsh	引脚：CS0	40			ns

VDD =3.0V±5%, Ta = 25℃

从 CPU 写到 ST7567（Writing Data from CPU to ST7567）

System Bus Timing for 6800 Series MPU

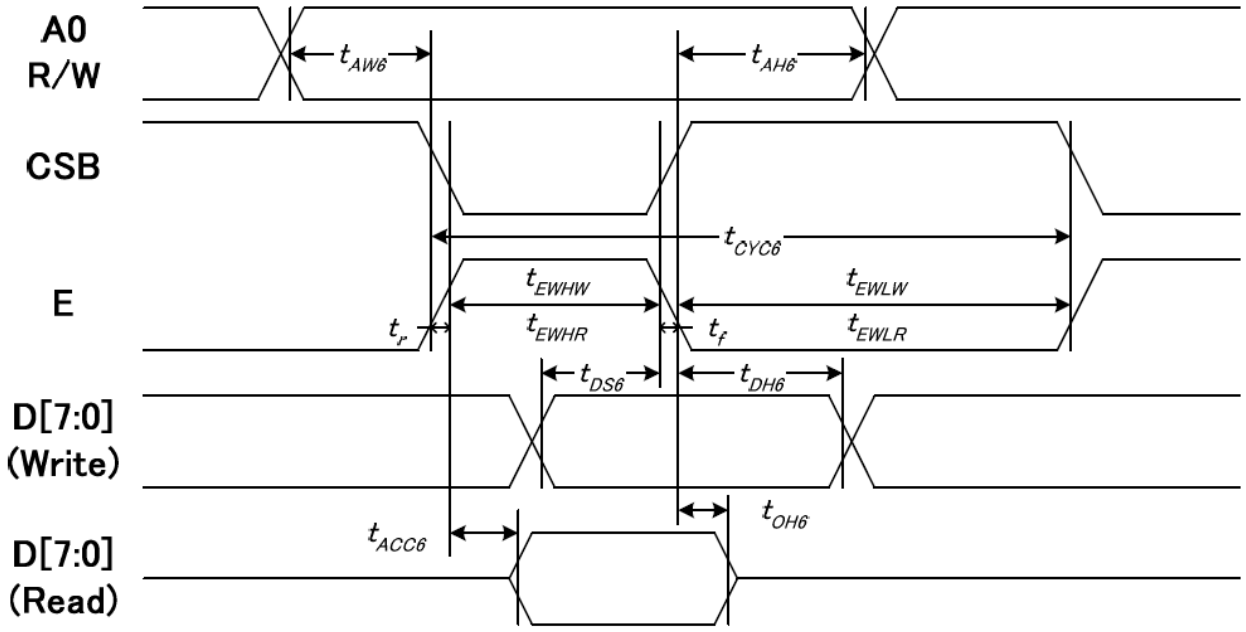


图 4. **从 CPU 写到 ST7567（Writing Data from CPU to ST7565R）**

System Bus Timing for 8080 Series MPU

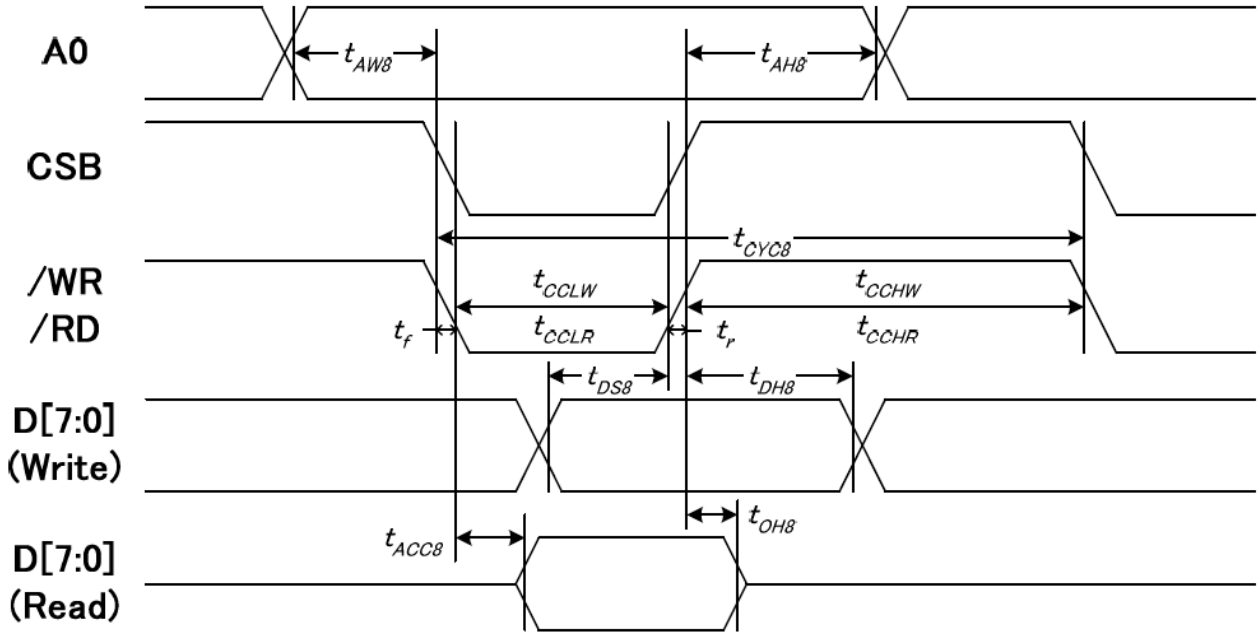


图 5. 从 CPU 写到 ST7567（Writing Data from CPU to ST7565R）

6.3 并行接口：时序要求（AC 参数）：

写数据到 ST7567 的时序要求：（6800 系列 MPU） **表 4.**

项 目	符 号	测试条件	极限值			单位
			MIN	TYPE	MAX	
地址保持时间	A0	tAH6	10	—	—	ns
地址建立时间		tAW6	0	—	—	
系统循环时间		tCYC6	240	—	—	
使能“低”脉冲（写）	WR	tEWLW	80	—	—	
使能“高”脉冲（写）		tEWHW	80	—	—	
使能“低”脉冲（读）	RD	tEWLR	80	—	—	
使能“高”脉冲（读）		tEWHR	140	—	—	
写数据建立时间	D7-D0	tDS6	40	—	—	
写数据保持时间		tDH6	10	—	—	
读时间		tACC6	—	—	70	
读输出来允许时间		tOH6	5	—	50	

VDD=3.3V ，Ta=25℃

写数据到 ST7567 的时序要求：（8080 系列 MPU） **表 5.**

项 目	符 号	测试条件	极限值			单位
			MIN	TYPE	MAX	
地址保持时间	A0	tAH8	0	—	—	ns
地址建立时间		tAW8	10	—	—	
系统循环时间		tCYC8	240	—	—	
使能“低”脉冲（写）	WR	tCCLW	80	—	—	
使能“高”脉冲（写）		tCCHW	80	—	—	
使能“低”脉冲（读）	RD	tCCLR	140	—	—	

使能“高”脉冲（读）		tCCHR	80	—	—	
写数据建立时间	D7-D0	tDS8	40	—	—	
写数据保持时间		tDH8	20	—	—	
读时间		tACC8	—	—	70	
读输出来允许时间		tOH8	5	—	50	

VDD=3.3V ，Ta=25℃

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

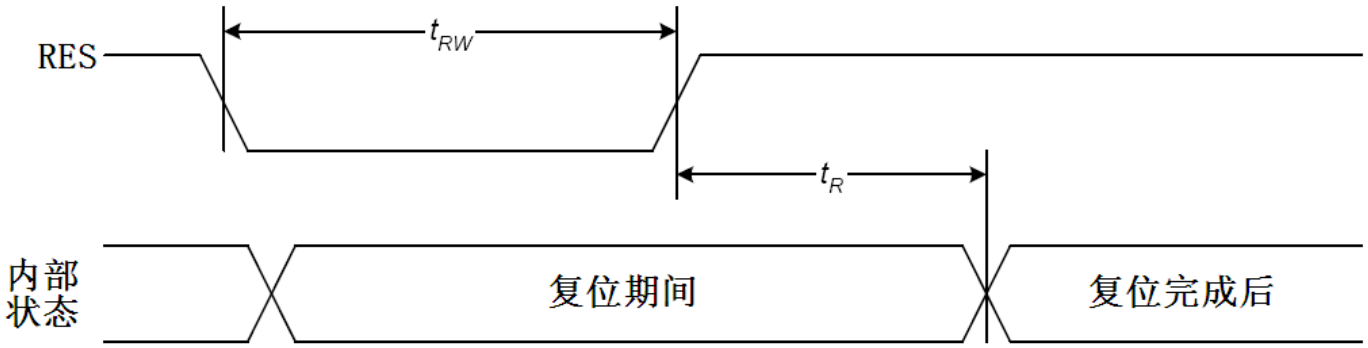


图 7：**电源启动后复位的时序**

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	tR		—	—	1.0	us
复位保持低电平的时间	tRW	引脚：RES	1.0	—	—	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

	列地址低4位设置		0	0	0	0	列地址的低 4 位				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
(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。**本液晶模块使用串行接口，不具备此功能**。
(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。 **本液晶模块使用串行接口，不具备此功能**。
(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。 **本液晶模块使用串行接口，不具备此功能**。 **详情请参考IC资料**
(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 **本液晶模块使用串行接口，不具备此功能**。 **详情请参考 IC 资料**
(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

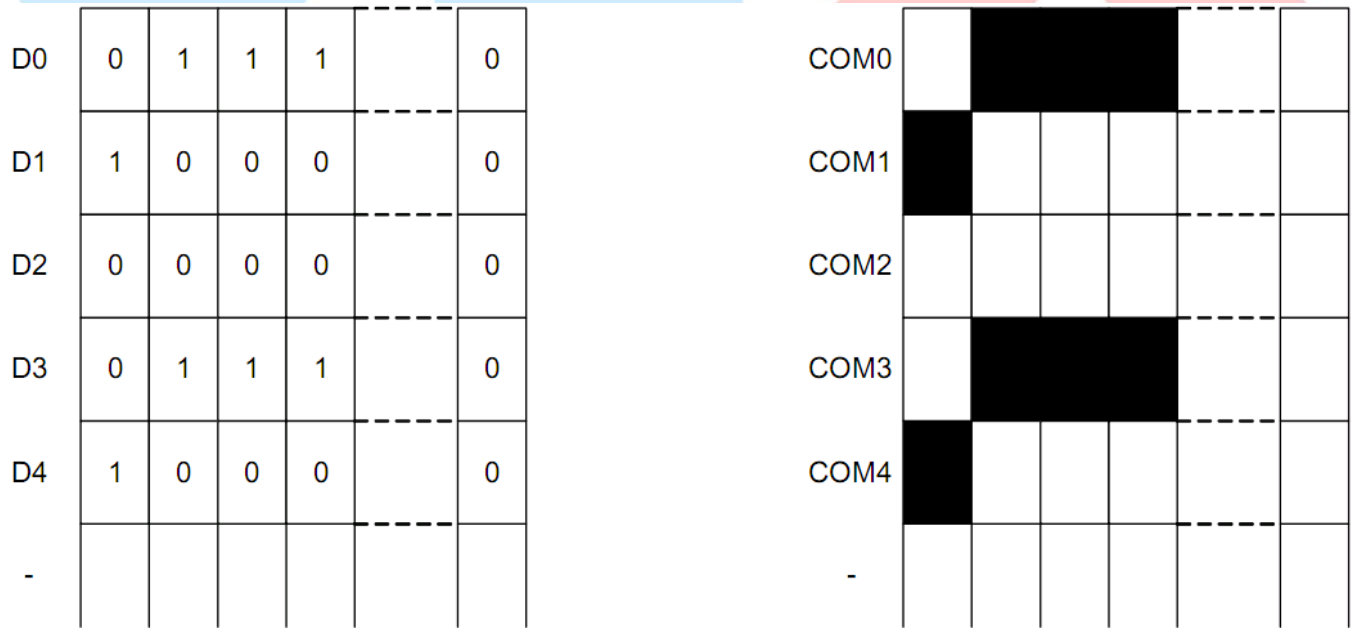
(18)	内部设置液晶电压模式	0	1	0	0	0	0	0	0	1	设置内部电阻微调，可以理解为**微调**对比度值，此两个指令需紧接着使用。上面一条指令 0x81 是不改的，下面一条指令可设置范围为：**0x00～0x3F,**数值越大对比度越浓，越小越淡
	设置的电压值		0	0	6 位电压值数据，0～63 共 64 级						
(19)静态图标显示：开/关		0	1	0	1	0	1	1	0	0 1	静态图标的开关设置： **0xAC**：关，**0xAD**：开。 此指令在进入及退出睡眠模式时起作用
(20) 升压倍数选择 (Booster ratio set)		0	1	1	1	1	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 “POWER SAVE”部分
(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_V1.7.PDF”的第 21～28 页。

7.3 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”。

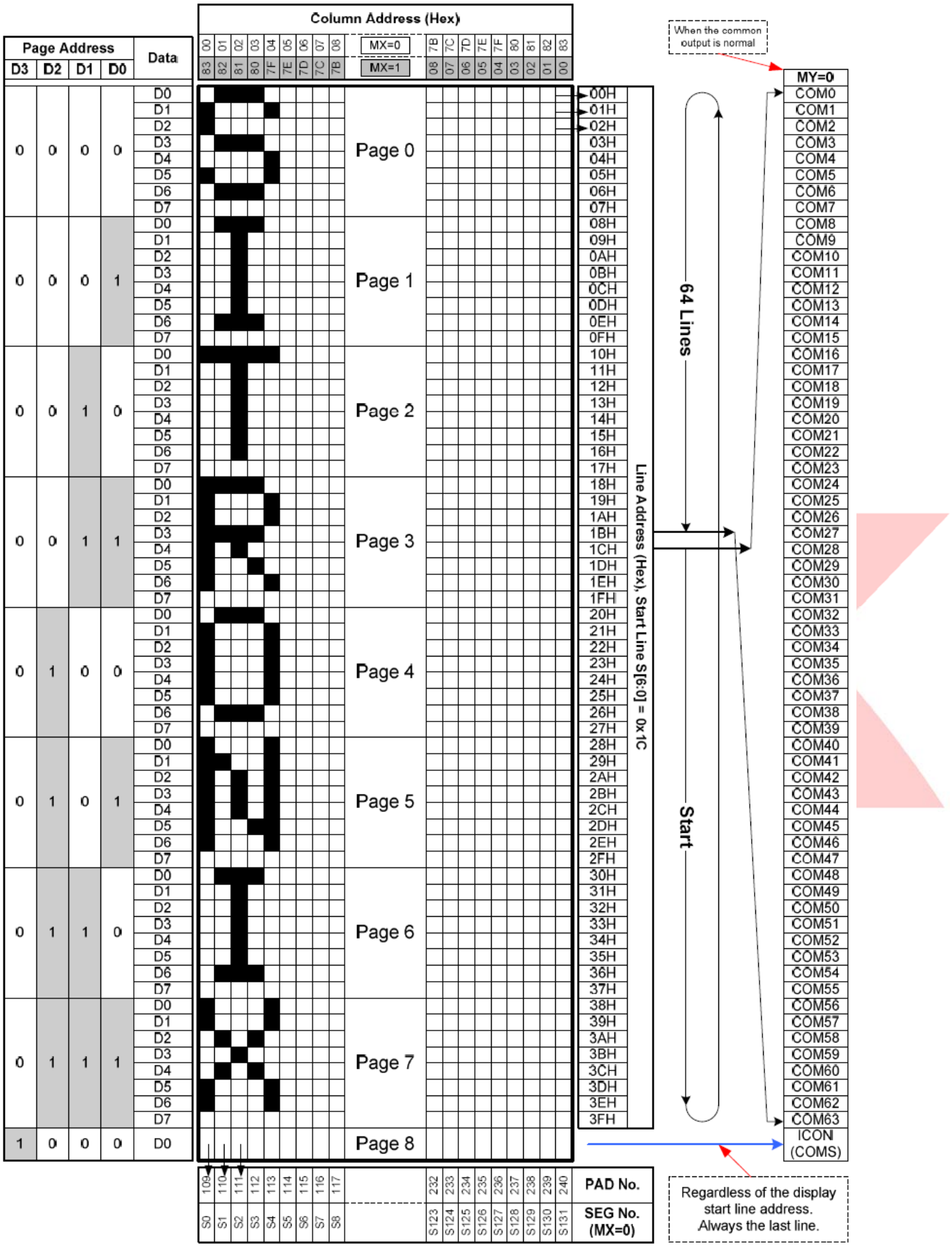
DB7--DB0 的排列方向：数据是从下向上排列的。最低位 D0 是在最上面，最高位 D7 是在最下面。每一位（bit）数据对应一个点阵，通常“1”代表点亮该点阵，“0”代表关掉该点阵. 如下图所示：



Display data RAM
（显示数据存储器）

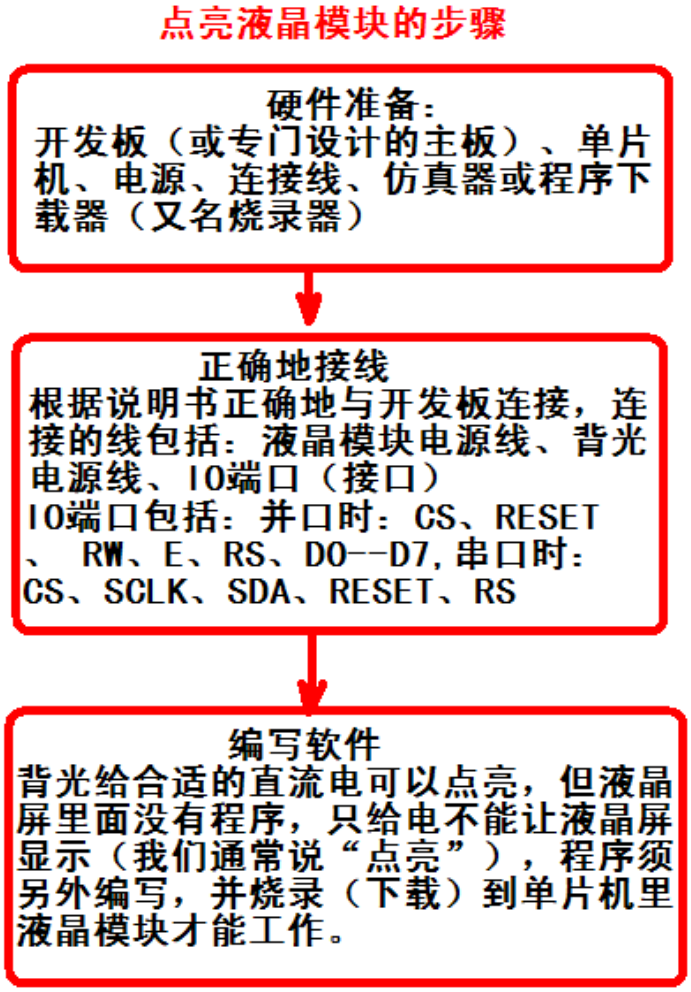
Liquid crystal display
（液晶屏）

下图摘自 ST7567 IC 资料，可通过“ST7567_V1.7.PDF”之第 18 页获取最佳效果。



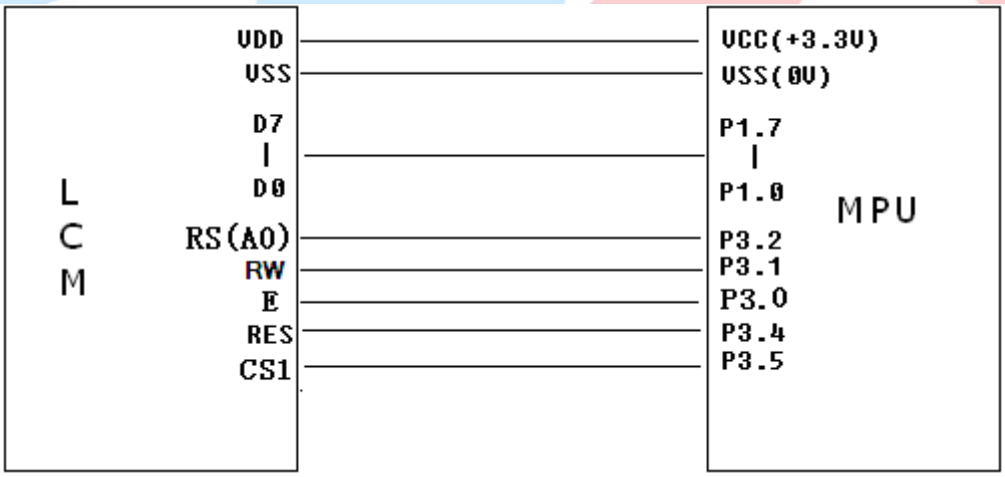
7.4 初始化方法

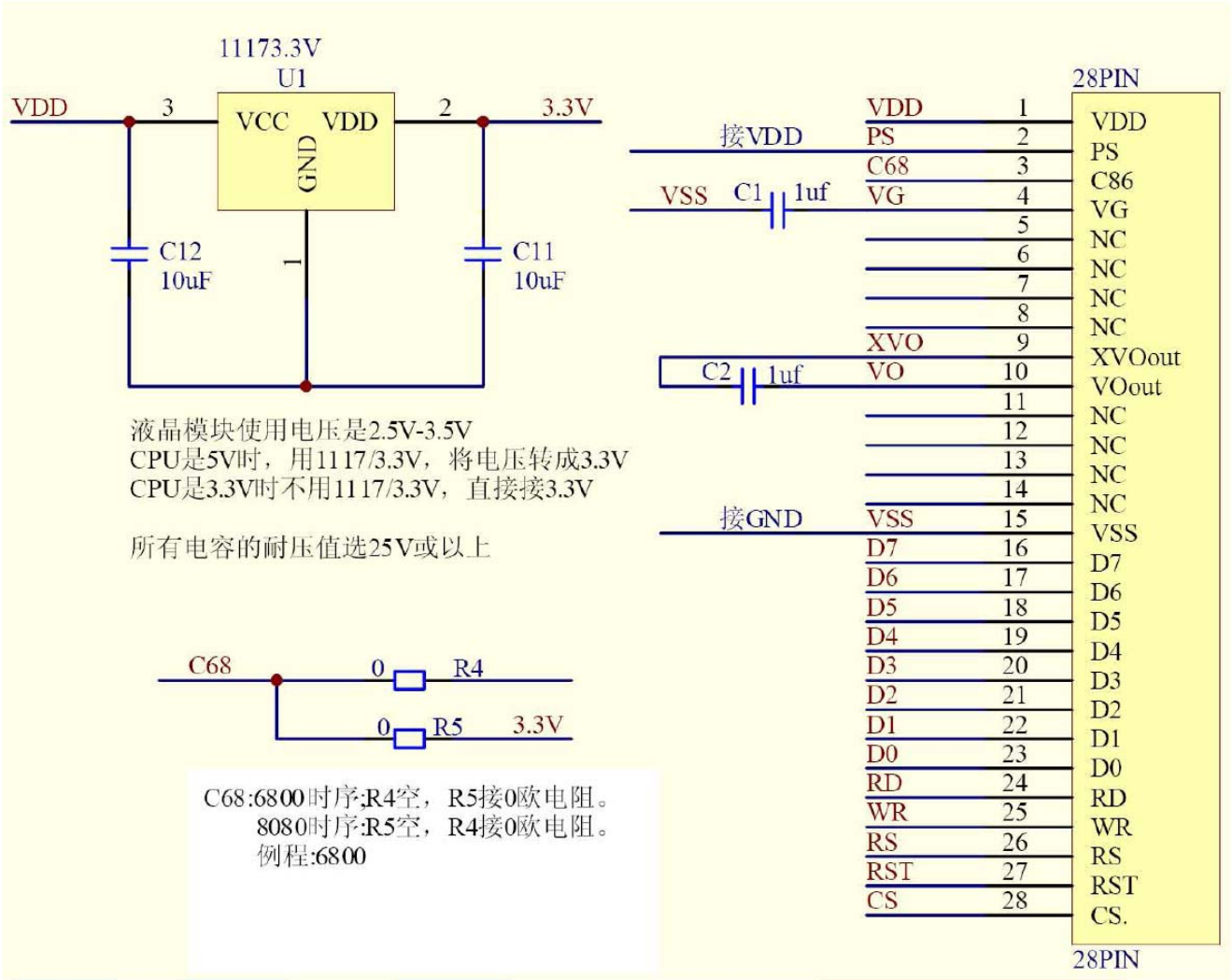
用户所编的显示程序,开始必须进行初始化,否则模块无法正常显示,过程请参考程序



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:并行接口






```
/* JLX12832G-03703-P-FOG 测试程序****/
/* LCD 驱动 IC:ST7567
/* 晶联讯电子：公司网址：http://www.jlxlcd.cn;  阿里巴巴网址：http://www.jlxlcd.com.cn*/
/* 该程序显示 2 行中文如下：*/
/* 全套液晶解决方案*/
/* 质量取胜创建口碑*/

#include <reg51.H>
#include <intrins.h>

sbit lcd_wr=P3^1;     /*接口定义:lcd_rw 就是 LCD 的 wr*/
sbit lcd_rd=P3^0;     /*接口定义:lcd_e 就是 LCD 的 rd*/
sbit lcd_rs=P3^2;     /*接口定义:lcd_rs 就是 LCD 的 rs*/
sbit lcd_cs1=P3^5;    /*接口定义:lcd_cs1 就是 LCD 的 cs1*/
sbit lcd_reset=P3^4; /*接口定义:lcd_reset 就是 LCD 的 reset*/

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

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

void displaygraphic(char *dp);

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:  畾  --*/
/*--  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};

/*写指令到 LCD 模块*/
void transfer_command_lcd(int data1)
{

    lcd_cs1=0;
    lcd_rs=0;
    lcd_rd=0;
    lcd_wr=0;
    P1=data1;
    lcd_rd=1;
    lcd_cs1=1;
    lcd_rd=0;
}

/*写数据到 LCD 模块*/
void transfer_data_lcd(int data1)
{
    lcd_cs1=0;
    lcd_rs=1;
    lcd_rd=0;
    lcd_wr=0;
    P1=data1;
    lcd_rd=1;
    lcd_cs1=1;
    lcd_rd=0;

}
```

```
/*延时*/
void delay(int i)
{
 int j,k;
 for(j=0;j<i;j++)
 for(k=0;k<500;k++);
}


/*等待一个按键，我的主板是用 P2.0 与 GND 之间接一个按键*/
void waitkey()
{
 repeat:
     if (P2&0x01) goto repeat;
     else delay(6);
     if (P2&0x01) goto repeat;
     else
     delay(40);
}

void lcd_address(int page,int column)
{

     column=column;
     transfer_command_lcd(0xb0+page-1);    /*设置页地址*/
     transfer_command_lcd(0x10+(column>>4&0x0f));    /*设置列地址的高 4 位*/
     transfer_command_lcd(column&0x0f);    /*设置列地址的低 4 位*/
}
/*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;
     for(j=0;j<2;j++)
     {
          lcd_address(page,column);
          for (i=0;i<16;i++)
          {
               transfer_data_lcd(*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++;
          }
          page++;
     }
     lcd_cs1=1;
}
```

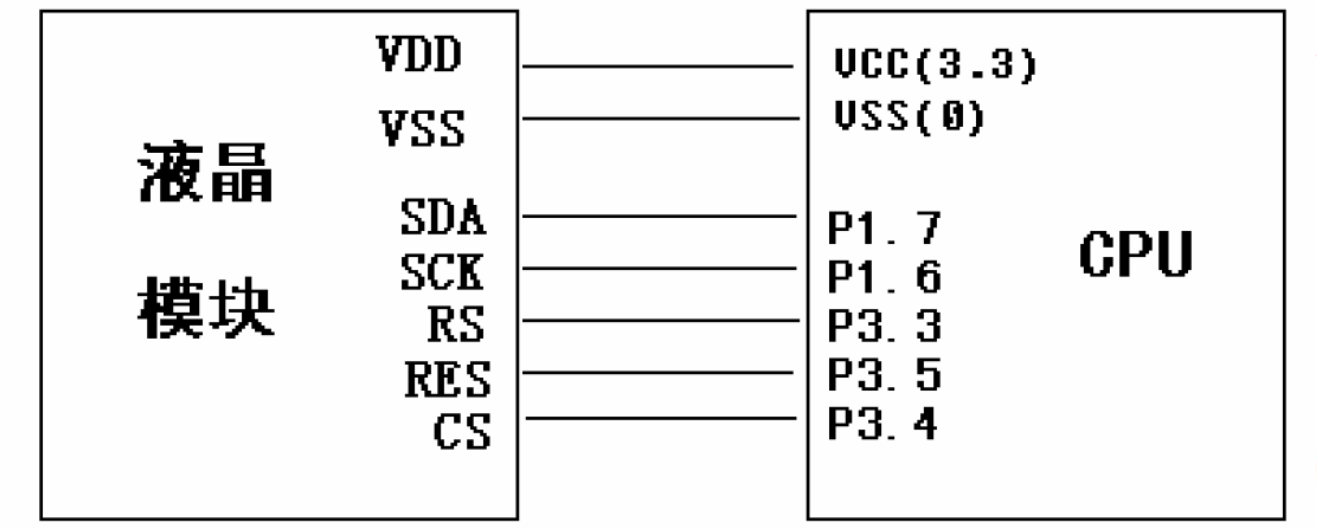
```
/*显示 8x16 点阵图像、ASCII, 或 8x16 点阵的自造字符、其他图标*/
void display_graphic_8x16(uint page,uchar column,uchar *dp)
{
    uint i,j;
    lcd_cs1=0;
    for(j=0;j<2;j++)
    {
        lcd_address(page,column);
        for (i=0;i<8;i++)
        {
            transfer_data_lcd(*dp);                    /*写数据到 LCD,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

/*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;
    lcd_cs1=0;
    lcd_address(page,column);
    for (col_cnt=0;col_cnt<8;col_cnt++)
    {
        transfer_data_lcd(*dp);
        dp++;
    }
    lcd_cs1=1;
}

//==================display a piture of 128*64 dots================
void displaygraphic(char *dp)
{
       int i,j;
        for(i=0;i<4;i++)
       {
       lcd_cs1=0;
        transfer_command_lcd(0xb0+i);   //set page address,
```

```
        transfer_command_lcd(0x10);
        transfer_command_lcd(0x00);
        for(j=0;j<128;j++)
        {
            transfer_data_lcd(*dp);
             dp++;
        }
         }

}


char code graphic1[]={
/*--  调入了一幅图像：D:\Backup\我的文档\图片\G-03703.bmp  --*/
/*--  宽度 x 高度=128x32  --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7E,0x2A,0x2A,0x2A,
0x2A,0x2A,0x2A,0x7E,0x00,0x00,0x00,0x00,0x02,0xFE,0x92,0x92,0x92,0xFE,0x12,0x11,
0x12,0x1C,0xF0,0x18,0x17,0x12,0x10,0x00,0x20,0x21,0x2E,0xE4,0x00,0x42,0x42,0xFE,
0x42,0x42,0x42,0x02,0xFE,0x00,0x00,0x00,0x00,0x00,0xF8,0x48,0x48,0x48,0x48,0xFF,
0x48,0x48,0x48,0x48,0xF8,0x00,0x00,0x00,0x00,0x00,0x02,0x02,0x02,0x02,0x02,0xE2,
0x12,0x0A,0x06,0x02,0x00,0x80,0x00,0x00,0x00,0x00,0x80,0x40,0x30,0x0E,0x84,0x00,
0x00,0x0E,0x10,0x60,0xC0,0x80,0x80,0x00,0x00,0x10,0x92,0x92,0x92,0x92,0x92,0x92,
0x92,0x92,0x12,0x02,0x02,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7F,0x25,0x25,0x25,0x25,0x7F,0x00,
0x00,0x7F,0x25,0x25,0x25,0x25,0x7F,0x00,0x08,0x1F,0x08,0x08,0x04,0xFF,0x05,0x81,
0x41,0x31,0x0F,0x11,0x21,0xC1,0x41,0x00,0x00,0x00,0x00,0x7F,0x20,0x10,0x00,0x7F,
0x00,0x00,0x00,0x00,0x3F,0x40,0x38,0x00,0x00,0x00,0x0F,0x04,0x04,0x04,0x04,0x3F,
0x44,0x44,0x44,0x44,0x4F,0x40,0x70,0x00,0x01,0x01,0x01,0x01,0x41,0x81,0x7F,
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x00,0x00,0x01,0x20,0x70,0x28,0x24,0x23,0x31,
0x10,0x10,0x14,0x78,0x30,0x01,0x00,0x00,0x00,0x00,0x1F,0x04,0x04,0x04,0x04,0x04,
0x04,0x0F,0x00,0x20,0x40,0x3F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x08,0x08,0xF8,0x08,0x08,0x00,
0x08,0xF8,0x08,0x00,0x00,0x00,0x00,0x00,0x08,0x18,0x68,0x80,0x80,0x68,0x18,0x08,
0x00,0x10,0x10,0xF8,0x00,0x00,0x00,0x00,0x00,0x70,0x08,0x08,0x08,0x88,0x70,0x00,
0x00,0x70,0x88,0x08,0x08,0x88,0x70,0x00,0x00,0x30,0x08,0x88,0x88,0x48,0x30,0x00,
0x00,0x70,0x08,0x08,0x08,0x88,0x70,0x00,0xC0,0x30,0x08,0x08,0x08,0x38,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xE0,0x10,0x08,0x08,0x10,0xE0,0x00,
0x00,0x30,0x08,0x88,0x88,0x48,0x30,0x00,0x00,0x38,0x08,0x08,0xC8,0x38,0x08,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x80,0x80,0x80,0x7F,0x00,0x00,0x00,
0x20,0x3F,0x20,0x20,0x20,0x20,0x30,0x00,0x20,0x30,0x2C,0x03,0x03,0x2C,0x30,0x20,
0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00,0x00,0x30,0x28,0x24,0x22,0x21,0x30,0x00,
0x00,0x1C,0x22,0x21,0x21,0x22,0x1C,0x00,0x00,0x18,0x20,0x20,0x20,0x11,0x0E,0x00,
0x00,0x30,0x28,0x24,0x22,0x21,0x30,0x00,0x07,0x18,0x20,0x20,0x22,0x1E,0x02,0x00,
0x00,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x00,0x0F,0x10,0x20,0x20,0x10,0x0F,0x00,
0x00,0x18,0x20,0x20,0x20,0x11,0x0E,0x00,0x00,0x00,0x00,0x3F,0x00,0x00,0x00,0x00,
```

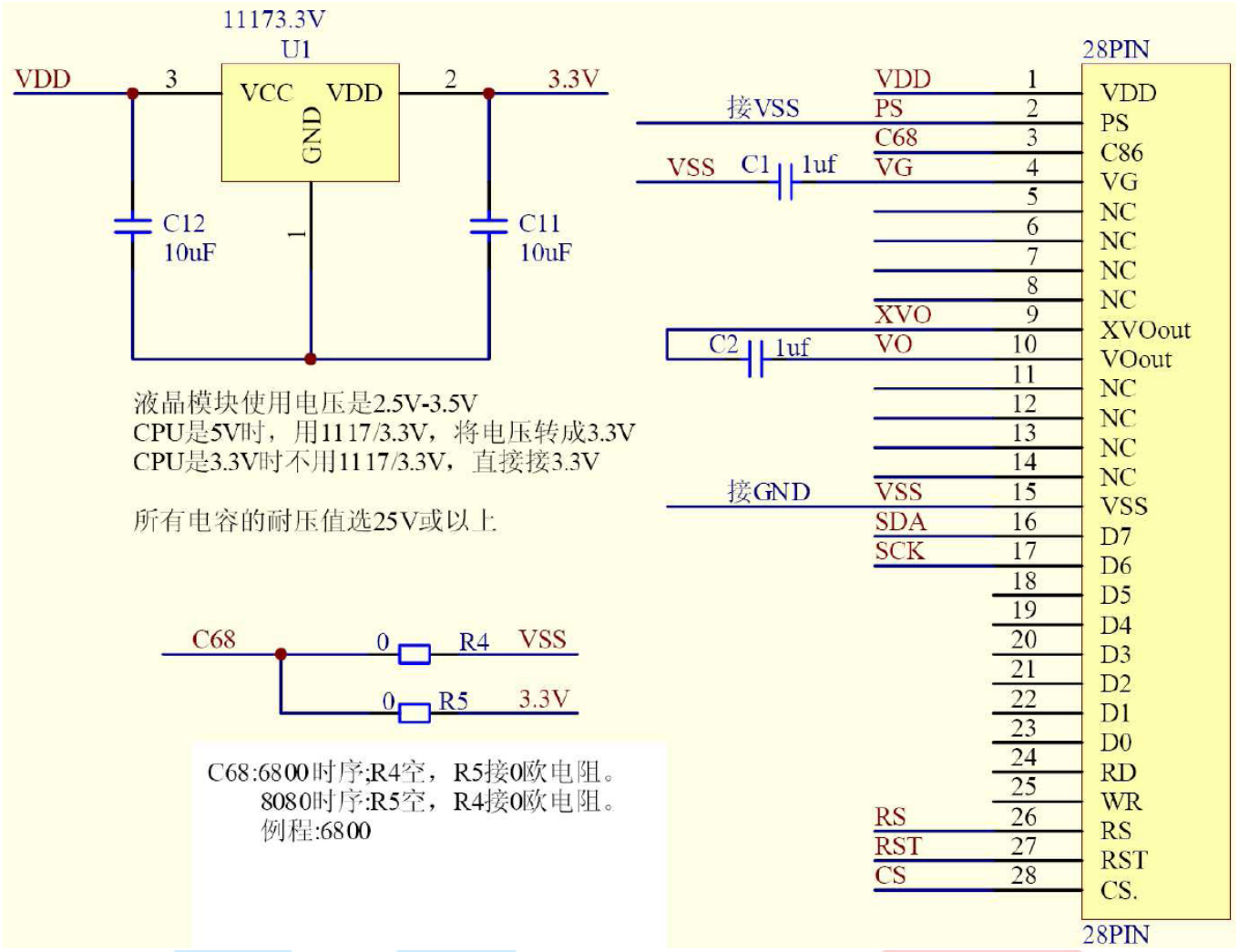
```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

};

//============initial
void initial_lcd()
{
    lcd_cs1=0;
    lcd_reset=0;                  //Reset the chip when reset=0
    delay(20);
    lcd_reset=1;
     transfer_command_lcd(0xe2);      /*软复位*/
     transfer_command_lcd(0x2c);      /*升压步聚 1*/
     delay(5);
     transfer_command_lcd(0x2e);      /*升压步聚 2*/
     delay(5);
     transfer_command_lcd(0x2f);      /*升压步聚 3*/
     delay(5);
     transfer_command_lcd(0x22);      /*粗调对比度，可设置范围 20～27*/
     transfer_command_lcd(0x81);      /*微调对比度*/
     transfer_command_lcd(0x1b);      /*微调对比度的值，可设置范围 0～63*/
     transfer_command_lcd(0xa2);      /*1/9 偏压比（bias）*/
     transfer_command_lcd(0xc8);      /*行扫描顺序：从上到下*/
     transfer_command_lcd(0xa0);      /*列扫描顺序：从左到右*/
     transfer_command_lcd(0x40);      /*起始行：从第一行开始*/
     transfer_command_lcd(0xaf);      /*开显示*/
    lcd_cs1=1;
}

//===============clear all dot martrics=============
void clear_screen()
{
unsigned char i,j;

     for(i=0;i<4;i++)
        {
        lcd_cs1=0;
          transfer_command_lcd(0xb0+i);
          transfer_command_lcd(0x10);
          transfer_command_lcd(0x00);
          for(j=0;j<132;j++)
          {
               transfer_data_lcd(0x00);
          }
          }
}
```

```
 //===============main program===================
void main(void)
 {
//   int i,j,k;
     lcd_cs1=0;
     initial_lcd();
        while(1)
        {

          clear_screen();    //clear all dots
             displaygraphic(graphic1);  //display a picture of 128*64 dots
          waitkey();

        }
 }
```

串行接口：



图为：串行接口



7.5.2、以下为串行接口方式范例程序

与并行方式相比较，只需改变接口顺序以及传送数据、传送命令这两个函数即可：

```
//-------------------------------------------------------
sbit lcd_cs1 =P3^4;  //接口定义，CS:片选
sbit lcd_reset =P3^5;  //接口定义，RESET:复位
sbit lcd_rs  =P3^3;  //接口定义，RS:命令/数据寄存器选择。也叫"A0",或“CD”
sbit lcd_sid =P1^7;  //接口定义，SID 即 SDA:串行数据
sbit lcd_sclk =P1^6;  //接口定义，sclk：串行时钟
//-------------------------------------------------------
//写命令到 LCD 模块
void transfer_command(int data1)
{
 char i;
 lcd_cs1=0;    //cs1=0,片选清零才可以传送命令或数据
 lcd_rs=0;    //rs=0：表示以下发送 1 个字节的命令
 for(i=0;i<8;i++)
 {
  lcd_sclk=0;
  if(data1&0x80) lcd_sid=1;
  else lcd_sid=0;
  lcd_sclk=1;
```

```
    data1<<=1;
  }
  lcd_cs1=1;    //cs1=1,当不用传数据给液晶屏时片选尽量置高，以免接收到一些干扰信号
}
//写数据到 LCD 模块
void transfer_data(int data1)
{
 char i;
 lcd_cs1=0;    //cs1=0,片选清零才可以传送命令或数据
 lcd_rs=1;    //rs=0：表示以下发送 1 个字节的数据
 for(i=0;i<8;i++)
 {
  lcd_sclk=0;
  if(data1&0x80) lcd_sid=1;
  else lcd_sid=0;
  lcd_sclk=1;
   data1<<=1;
 }
 lcd_cs1=1;       //cs1=1,当不用传数据给液晶屏时片选尽量置高，以免接收到一些干扰信号
```