

JLX12864G-10902-PN 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
5	技术参数	5
6	时序特性	5~9
7	指令功能及硬件接口与编程案例	9~末页

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-10902-PN 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-10902-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. JLX12864G-10902-PN 图像型点阵液晶模块的特性

2.1 结构牢：模块背光带有挡墙及带铁框，焊接式 FPC。

2.2 IC 采用矽创公司 ST7565R，功能强大，稳定性好

2.3 功耗低：2-200mW（不带背光<2mW，带背光<300mW）；

2.4 显示内容：

- 128*64 点阵单色图片；

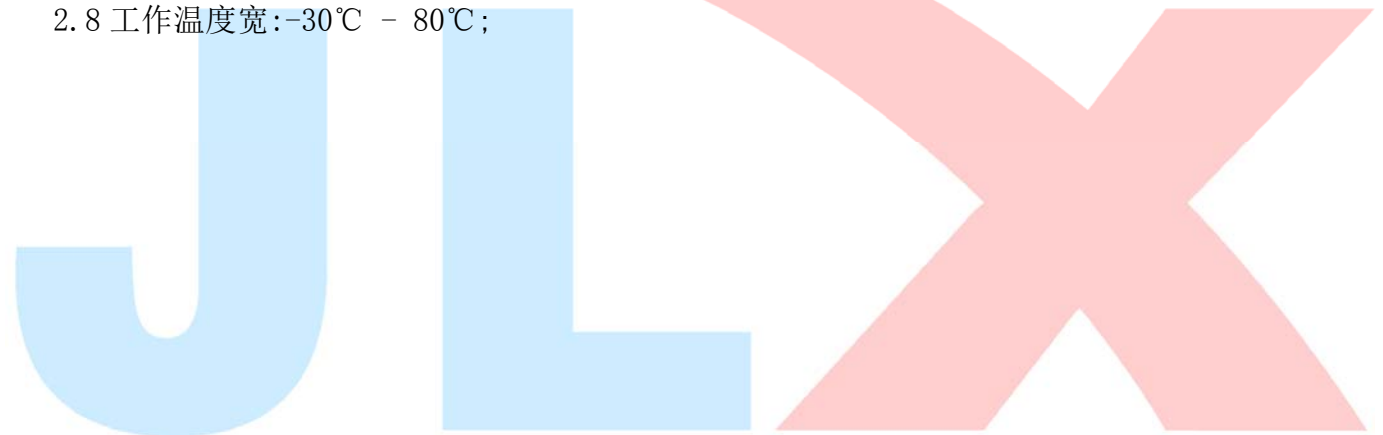
- 可選用 16*16 点阵或其他点阵的图片来自编汉字，按照 16*16 点阵汉字来计算可显示 8 字/行*4 行。按照 12*12 点阵汉字来计算可显示 10 字/行*4 行。

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）。并口时：可以“读-改-写”；

2.6 接口简单方便：可采用 4 线 SPI 串行接口，或选择并行接口（6800 时序和 8080 时序可选）。

2.7 工作温度宽：-20℃ - 70℃；

2.8 工作温度宽：-30℃ - 80℃；



3. 外形尺寸及接口功能

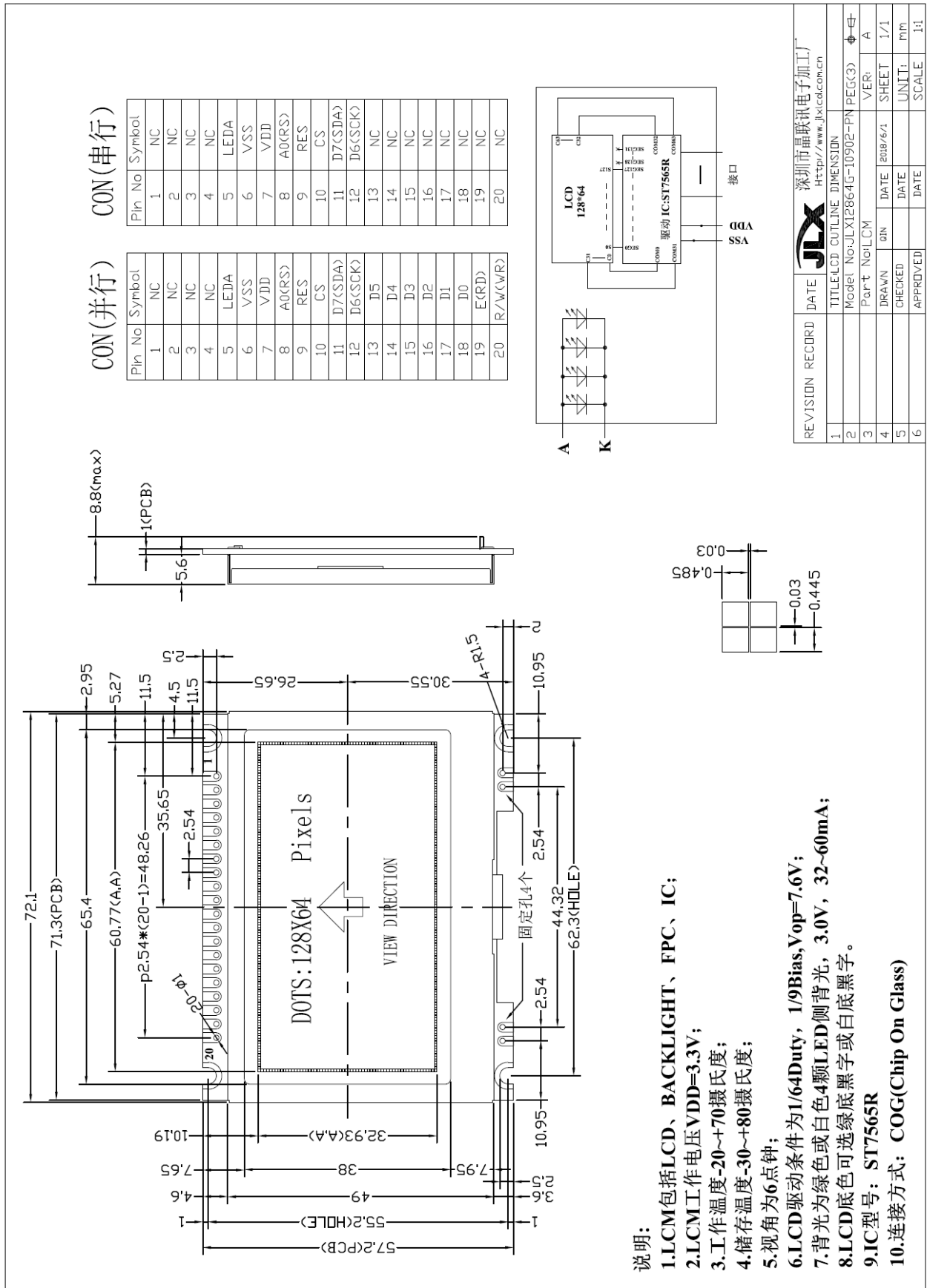


图 1. 液晶模块外形尺寸

模块的接口引脚功能

引线号	符号	名称	功能
1	NC		空脚
2	NC		空脚
3	NC		空脚
4	NC		空脚
5	LEDA	背光电源	背光电源正极，同 VDD 电压（5V 或 3.3V）
6	VSS	接地	0V
7	VDD	电路电源	5V 或 3.3V
8	A0 (RS)	寄存器选择信号	H:数据寄存器 0:指令寄存器（IC 资料上所写为” A0”）
9	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
10	CS	片选	低电平片选
11	D7	I/O	并行时：数据总线 DB7 串行时：串行数据 (SDA)
12	D6	I/O	并行时：数据总线 DB6 串行时：串行时钟 (SCLK)
13-18	D5-D0	I/O	并行时：数据总线 DB0~DB5 串行时：空
19	RD (E)	使能信号	并行时：使能信号 串行时：空
20	WR	读/写	并行时：H:读数据 0:写数据 串行时：空

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 绑定在 LCD 玻璃上（这种加工工艺叫 COG）。

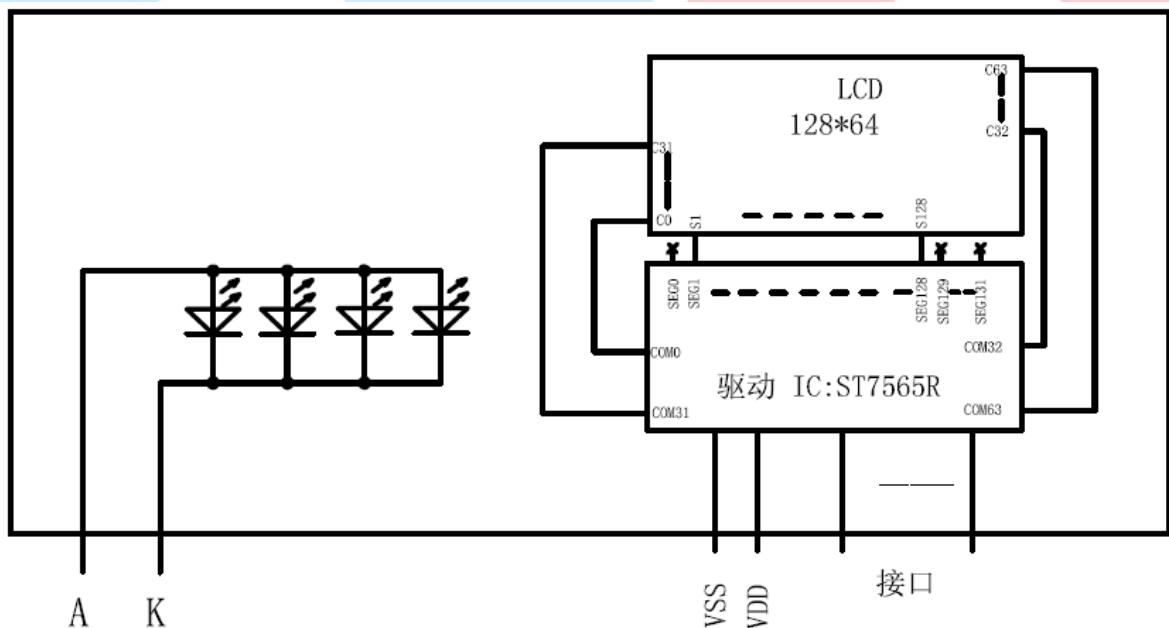


图 2: JLX12864G-10902-PN 图像点阵型液晶模块的电路框图

4.3 背光参数

该型号液晶模块带 LED 背光源。它的性能参数如下:

工作温度: $-20 \sim +70^{\circ}\text{C}$;

存储温度: $-30 \sim +80^{\circ}\text{C}$;

背光板可选择绿色、白色。

正常工作电流为: $(8 \sim 15) * 4 = 32 \sim 60\text{mA}$ (LED 灯数共 4 颗);

工作电压: 5V 或 3.3V, 由你选择的 VDD 电源电压 (5V 或 3.3V) 决定;

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		3.5	V
LCD 驱动电压	VDD - VO	VDD - 13.5		VDD + 0.3	V
静电电压		-	-	100	V
工作温度		-20		+70	$^{\circ}\text{C}$
储存温度		-30		+80	$^{\circ}\text{C}$

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 (当 3.3V 供电时)	VDD		2.4	3.3	3.5	V
工作电压 (当 5.0V 供电时)			4.8	5.0	5.2	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} = 0.2mA	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
背光工作电流	I _{LED}	VLED=3.0V	32	60	80	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 ST7565R (Writing Data from CPU to ST7565R)

The 4-line SPI Interface

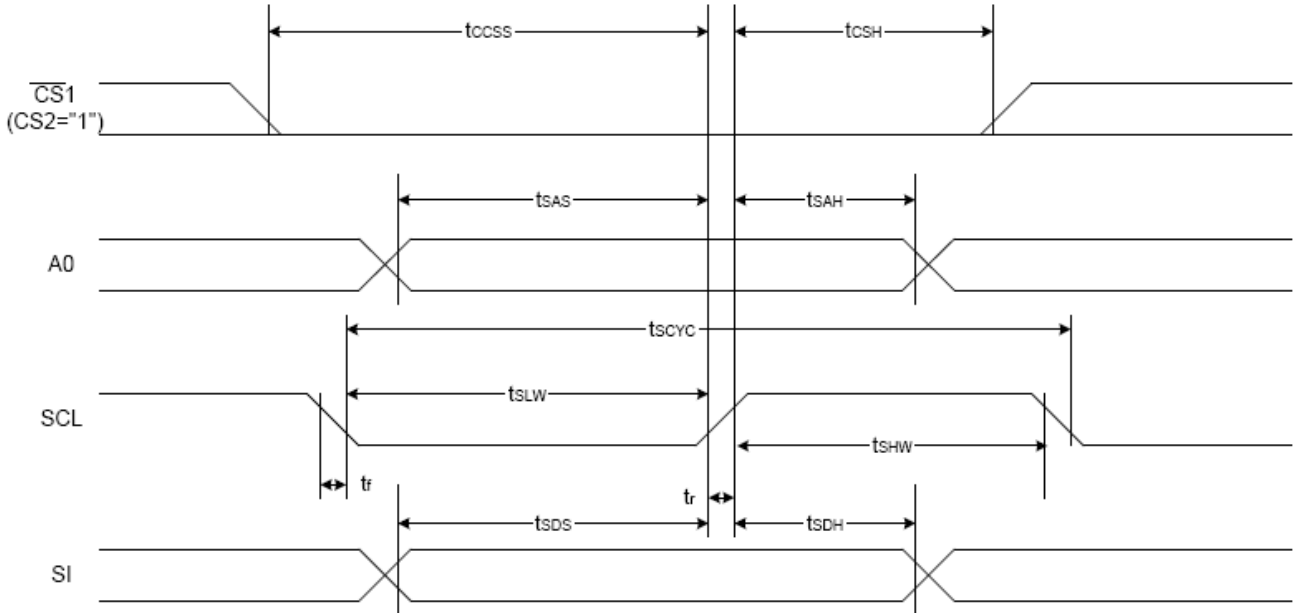


图 4. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

6.2 串行接口: 时序要求 (AC 参数):

写数据到 ST7565R 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	25	--	50	ns
保持SCK高电平脉宽 (SCK "H" pulse width)	T_{shw}	引脚: SCK	25			ns
保持SCK低电平脉宽 (SCK "L" pulse width)	T_{slw}	引脚: SCK	25			ns
地址建立时间 (Address setup time)	T_{sas}	引脚: RS	20	--	--	ns
地址保持时间 (Address hold time)	T_{sah}	引脚: RS	10	--	--	ns
数据建立时间 (Data setup time)	T_{sds}	引脚: SI	20	--	--	ns
数据保持时间 (Data hold time)	T_{sdh}	引脚: SI	10	--	--	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	40			ns

VDD = 3.0V ± 5%, Ta = 25°C

6.3 并行接口:

从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

System Bus Read/Write Characteristics 1 (For the 8080 Series MPU)

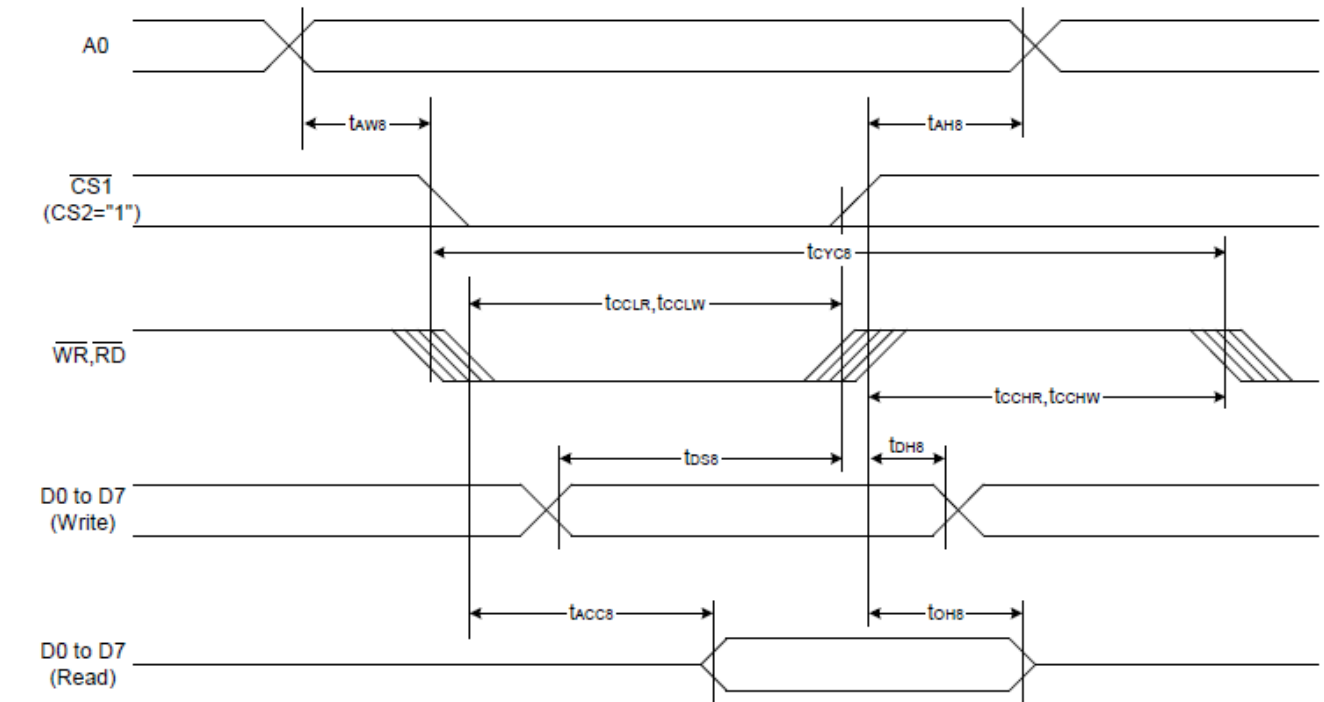


图 5. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

System Bus Read/Write Characteristics 2 (For the 6800 Series MPU)

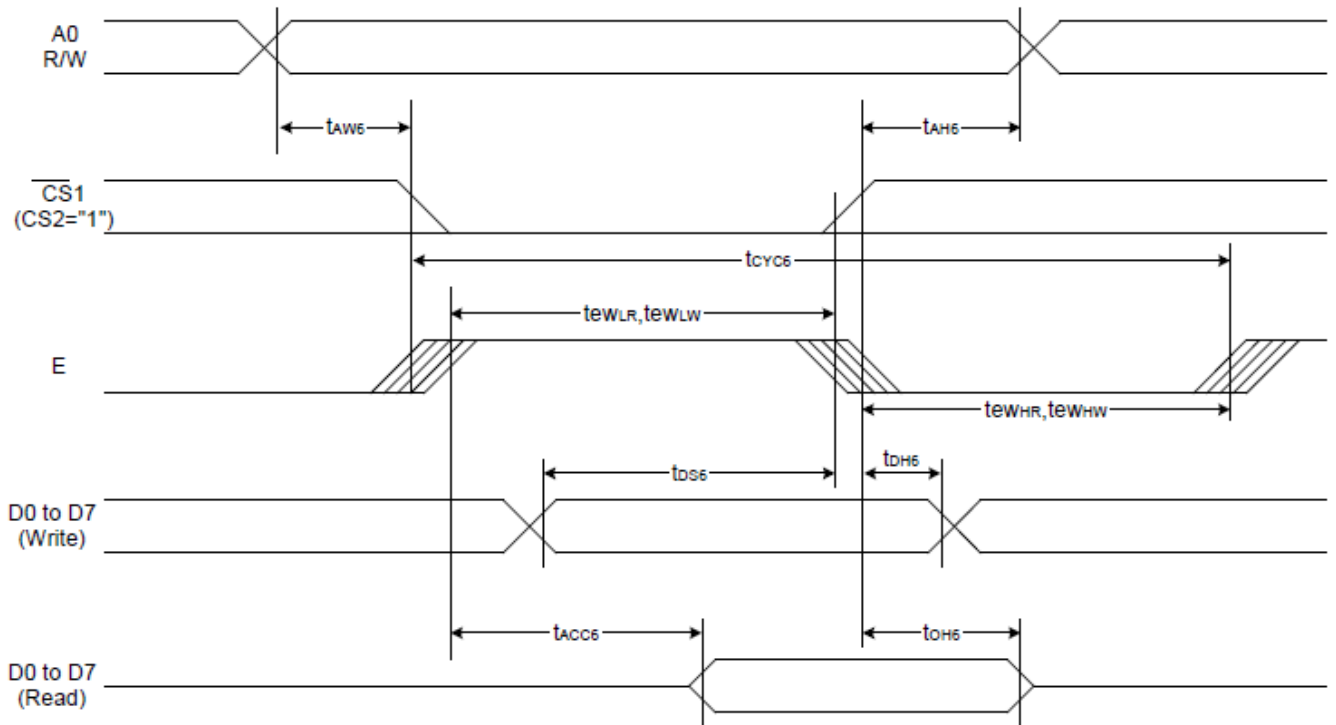


图 6. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

6.4 并行接口: 时序要求 (AC 参数):

写数据到 ST7565R 的时序要求: (8080 系列 MPU)

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	0	—	—	
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	D0-D7	tDS8	40	—	—	
写数据保持时间		tDH8	0	—	—	
读时间		tACC8	—	—	70	
读输出允许时间		tOH8	5	—	50	

写数据到 ST7565R 的时序要求: (6800 系列 MPU)

项目	符号	测试条件	极限值			单位
			MIN	TYPE	MAX	
地址保持时间	A0	tAH6	0	—	—	
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	—	—	ns
使能“低”脉冲（读）	RD	tEWLR	80	—	—	
使能“高”脉冲（读）		tEWHR	140	—	—	
写数据建立时间	D0-D7	tDS6	40	—	—	
写数据保持时间		tDH6	0	—	—	
读时间		tACC6	—	—	70	
读输出允许时间		tOH6	5	—	50	

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

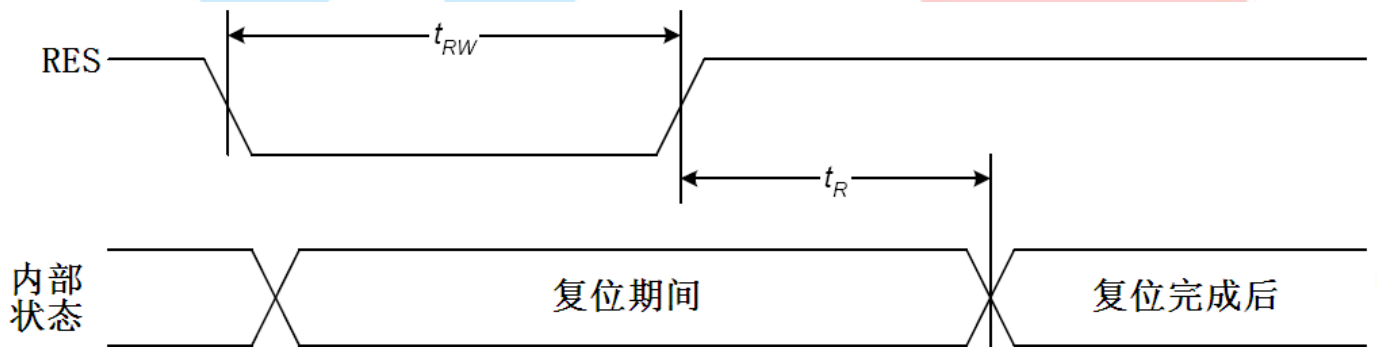


图 7：电源启动后复位的时序

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		—	—	1.0	us
复位保持低电平的时间	t_{RW}	引脚：RES	1.0	—	—	us

7. 指令功能:

7.1 指令表

格式:

RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
----	-----	-----	-----	-----	-----	-----	-----	-----

共11种指令:1. 清除, 2. 返回, 3. 输入方式设置, 4. 显示开关, 5. 控制, 移位, 6. 功能设置, 7. CGRAM 地址设置, 8. DDRAM 地址设置, 9. 读忙标志, 10. 写数据到 CG/DDRAM, 11. 读数据由 CG/DDRAM.

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
	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									串口时: 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	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	
(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	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	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!

温馨提示: 请详细参考 IC 资料”ST7565R_V1.9.PDF”的第 28~36 页。

7.2 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：

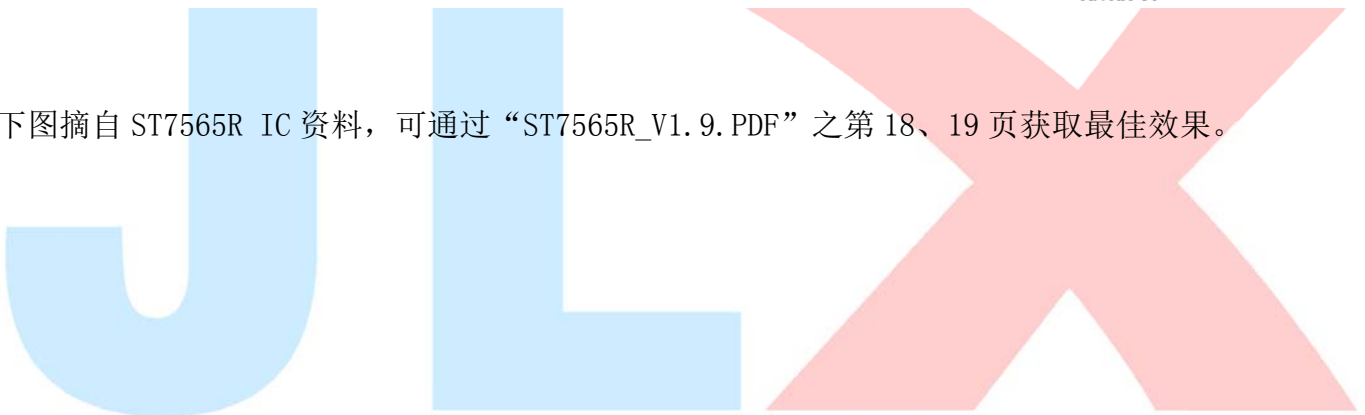
D0	0	1	1	1		0
D1	1	0	0	0		0
D2	0	0	0	0		0
D3	0	1	1	1		0
D4	1	0	0	0		0
-						

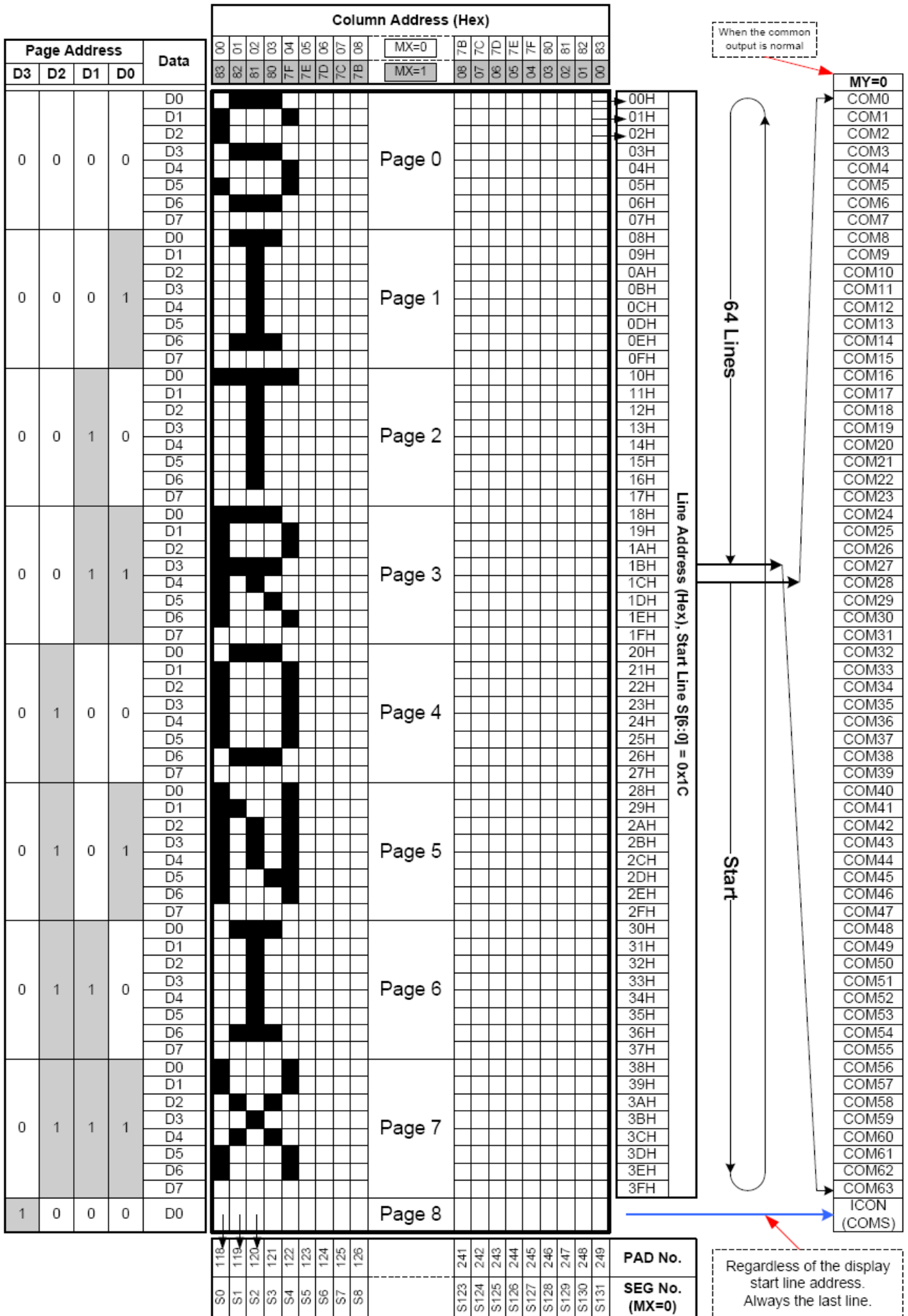
Display data RAM
(显示数据存储器)

COM0						
COM1						
COM2						
COM3						
COM4						
-						

Liquid crystal display
(液晶屏)

下图摘自 ST7565R IC 资料，可通过“ST7565R_V1.9.PDF”之第 18、19 页获取最佳效果。





When the common output is normal

64 Lines

Start

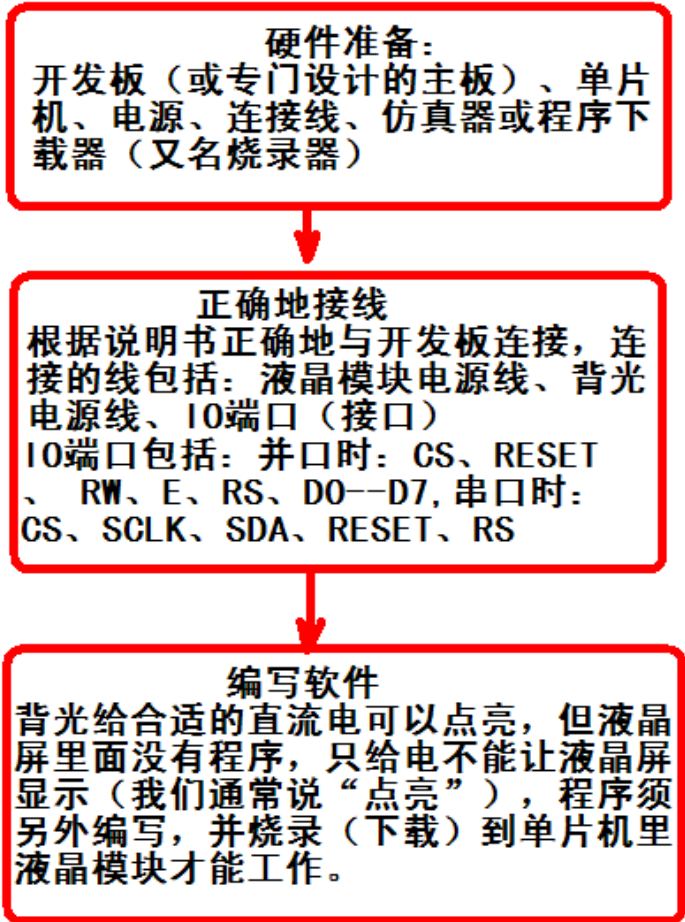
Line Address (Hex), Start Line S[6:0] = 0x1C

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

点亮液晶模块的步骤



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:

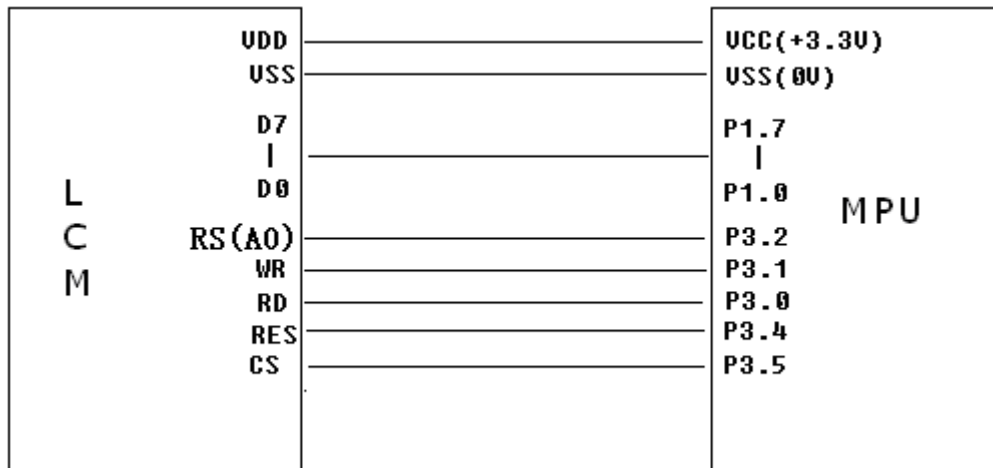
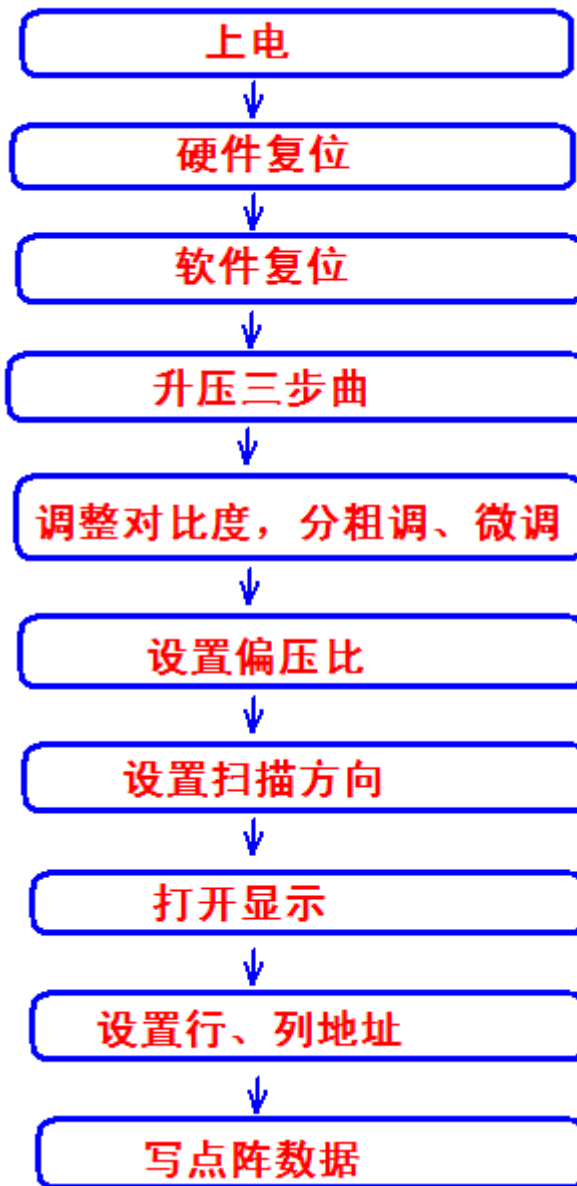


图 8. 并行接口

7.51、程序

点亮液晶模块的编程步骤



并行程序:

并行程序:

```

/* Test program for JLX12864G-10902, 并行接口
   驱动 IC 是:ST7565R(or compatible)
   晶联讯电子: 网址 http://www.jlxlcd.cn; http://www.jlxlcd.com.cn
*/
#include <reg51.H>

sbit rs=P3^2; /*接口定义:lcd_rs 就是 LCD 的 rs*/
sbit rd=P3^0; /*接口定义:lcd_e 就是 LCD 的 rd*/
sbit wr=P3^1; /*接口定义:lcd_rw 就是 LCD 的 wr*/
  
```

```
sbit reset=P3^4; /*接口定义:lcd_reset 就是 LCD 的 reset*/
sbit cs1=P3^5; /*接口定义:lcd_cs1 就是 LCD 的 cs1*/
```

```
void transfer_data(int data1);
void transfer_command(int data1);
char code graphic1[];
char code graphic2[];
char code graphic3[];
char code graphic4[];
char code graphic5[];
char code graphic6[];
void delay(int i);
void Delay1(int i);
void disp_grap(char *dp);
void initial_lcd();
void clear_screen();
void waitkey();
```

```
//=====main program=====
void main(void)
{
    int i,j,k;
    initial_lcd();
    while(1)
    {
        clear_screen(); //clear all dots
        disp_grap(graphic1); //display a picture of 128*64 dots
        waitkey();
        disp_grap(graphic2); //display a picture of 128*64 dots
        waitkey();
        disp_grap(graphic4); //display a picture of 128*64 dots
        waitkey();
        disp_grap(graphic5); //display a picture of 128*64 dots
        waitkey();
        disp_grap(graphic6); //display a picture of 128*64 dots
        waitkey();
    }
}

//=====initial
void initial_lcd()
{
```



```

    reset=0;                /*低电平复位*/
    delay(20);
    reset=1;                /*复位完毕*/
    delay(20);
    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(0x24); /*粗调对比度，可设置范围 0x20~0x27*/
    transfer_command(0x81); /*微调对比度*/
    transfer_command(0x1b); /*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(0x60); /*起始行：第一行开始*/
    transfer_command(0xaf); /*开显示*/
}

```

//=====clear all dot martrices=====

```

void clear_screen()
{
    unsigned char i, j;

    for(i=0; i<9; i++)
    {
        cs1=0;
        transfer_command(0xb0+i);
        transfer_command(0x10);
        transfer_command(0x00);
        for(j=0; j<132; j++)
        {
            transfer_data(0x00);
        }
    }
}

```

//=====display a picture of 128*64 dots=====

```

void disp_grap(char *dp)
{
    int i, j;

```

```
for(i=0;i<8;i++)
{
    cs1=0;
    transfer_command(0xb0+i); //set page address,
    transfer_command(0x10);
    transfer_command(0x00+1);
    for(j=0;j<128;j++)
    {
        transfer_data(*dp);
        dp++;
    }
}
```

```
//=====transfer command to LCM=====
```

```
void transfer_command(int data1)
```

```
{
    cs1=0;
    rs=0;
    rd=0;
    wr=0;
    P1=data1;
    rd=1;
    cs1=1;
    rd=0;
}
```

```
//-----transfer data to LCM-----
```

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

```
{
    cs1=0;
    rs=1;
    rd=0;
    wr=0;
    P1=data1;
    rd=1;
    cs1=1;
    rd=0;
}
```

```
//=====delay time=====
```

```
void delay(int i)
```

```
{
    int j,k;
    for(j=0;j<i;j++)
```

```

for(k=0;k<990;k++);
}

//=====delay time=====
void Delay1(int i)
{
    int j,k;
    for(j=0;j<i;j++)
        for(k=0;k<10;k++);
}

//-----wait a switch, jump out if P2.0 get a signal"0"-----
void waitkey()
{
    repeat:
        if (P2&0x01) goto repeat;
        else delay(6);
        if (P2&0x01) goto repeat;
        else
            delay(40);;
}

char code graphic1[]={
/*-- 调入了一幅图像: E:\新开发部\显示图案收藏\12864G-10902 英文.bmp --*/
/*-- 宽度 x 高度=128x64 --*/
0xFF, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,
0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0xF1, 0x01, 0xF1, 0x01, 0x01, 0x01, 0x01, 0x11,
0x61, 0x81, 0x81, 0x61, 0x11, 0x01, 0x41, 0x21, 0xF1, 0x01, 0x01, 0x01, 0x21, 0x11, 0x11, 0x11,
0xE1, 0x01, 0x61, 0x91, 0x91, 0x91, 0x61, 0x01, 0xE1, 0x91, 0x91, 0x91, 0x21, 0x01, 0x01, 0xC1,
0x21, 0xF1, 0x01, 0x01, 0xC1, 0x21, 0x11, 0x11, 0x11, 0x21, 0x41, 0x01, 0x01, 0x01, 0x01, 0x01,
0x41, 0x21, 0xF1, 0x01, 0x01, 0x01, 0xE1, 0x11, 0x11, 0x11, 0xE1, 0x01, 0x11, 0x11, 0x91, 0x71,
};

```

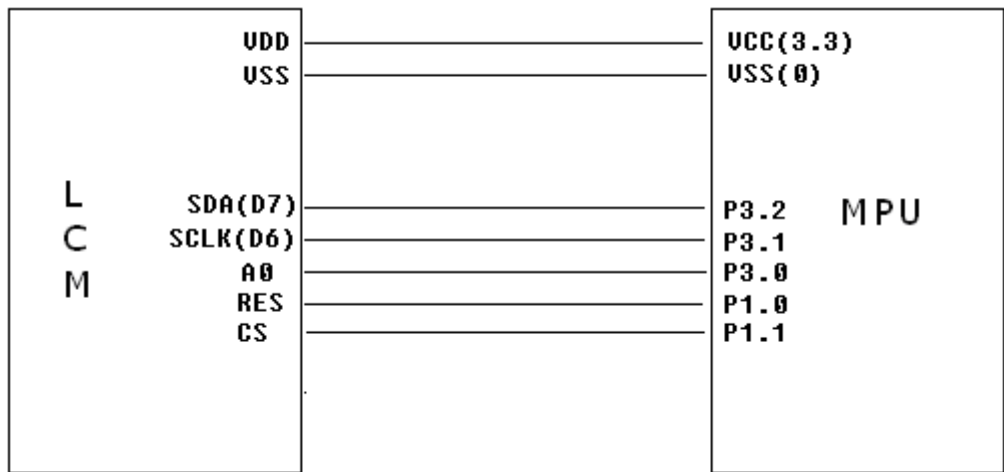


图 9. 串行接口

串行程序与并行只是接口定义和写数据和命令不一样，其它都一样

串行程序:

```
#include <reg51.H>

sbit lcd_rs=P3^0; /*接口定义:lcd_rs 就是 LCD 的 rs*/
sbit lcd_sclk=P3^1; /*接口定义:lcd_sclk 就是 LCD 的 sclk*/
sbit lcd_sid=P3^2; /*接口定义:lcd_sid 就是 LCD 的 sid*/
sbit reset=P1^0; /*接口定义:lcd_reset 就是 LCD 的 reset*/
sbit cs1=P1^1; /*接口定义:lcd_cs1 就是 LCD 的 cs1*/

void transfer_command(int data1)
{
    char i;
    cs1=0;
    lcd_rs=0;
    for (i=0;i<8;i++)
    {
        lcd_sclk=0;
        if(data1&0x80) lcd_sid=1;
        else lcd_sid=0;
        lcd_sclk=1;
        data1=data1<<=1;
    }
    cs1=1;
}

/*写数据到 LCD 模块*/
void transfer_data(int data1)
{
    char i;
    cs1=0;
    lcd_rs=1;
    for (i=0;i<8;i++)
    {
        lcd_sclk=0;
        if(data1&0x80) lcd_sid=1;
        else lcd_sid=0;
        lcd_sclk=1;
        data1=data1<<=1;
    }
    cs1=1;
}
```