

# JLX19264G-329-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	6
6	时序特性	7~10
7	指令功能及硬件接口与编程案例	11~页末

## 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 JLX19264G-329 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。

JLX19264G-329 可以显示 192 列\*64 行点阵单色图片, 或显示 12 个/行\*4 行 16\*16 点阵的汉字, 或显示 24 个/行\*8 行 8\*8 点阵的英文、数字、符号。

## 2. JLX19264G-329 图像型点阵液晶模块的特性

2.1 结构牢。

2.2 IC 采用 IST3020, 功能强大, 稳定性好

2.3 功耗低: 2-200mW (不带背光<2mW, 带背光<200mW) ;

2.4 显示内容:

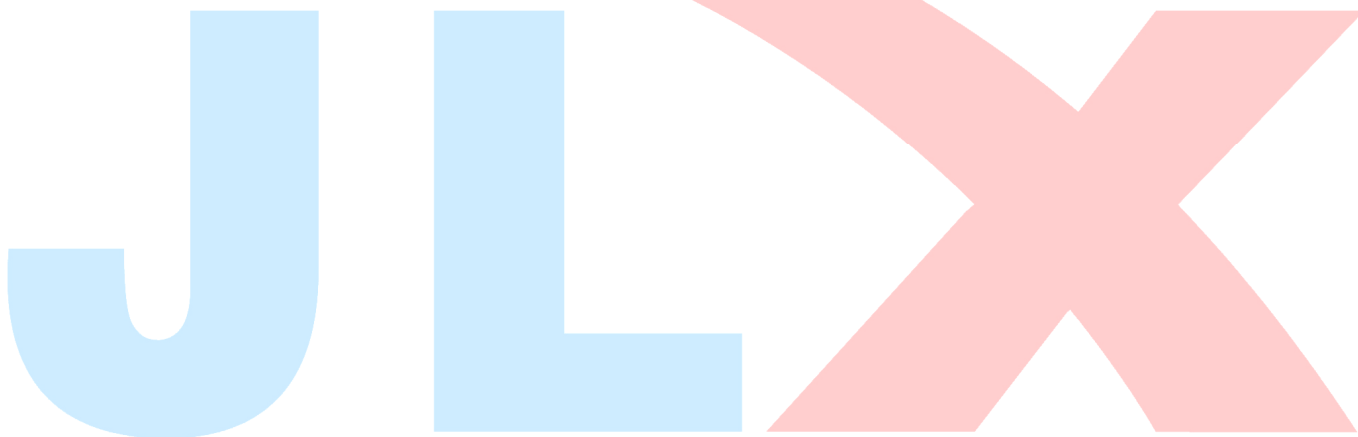
- 192\*64 点阵单色图片;

- 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 12 字/行\*4 行。按照 12\*12 点阵汉字来计算可显示 16 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: 可采用 4 线 SPI 串行接口, 或选择并行接口。

2.7 工作温度宽: -20℃ - 70℃;



3. 外形尺寸及接口引脚功能

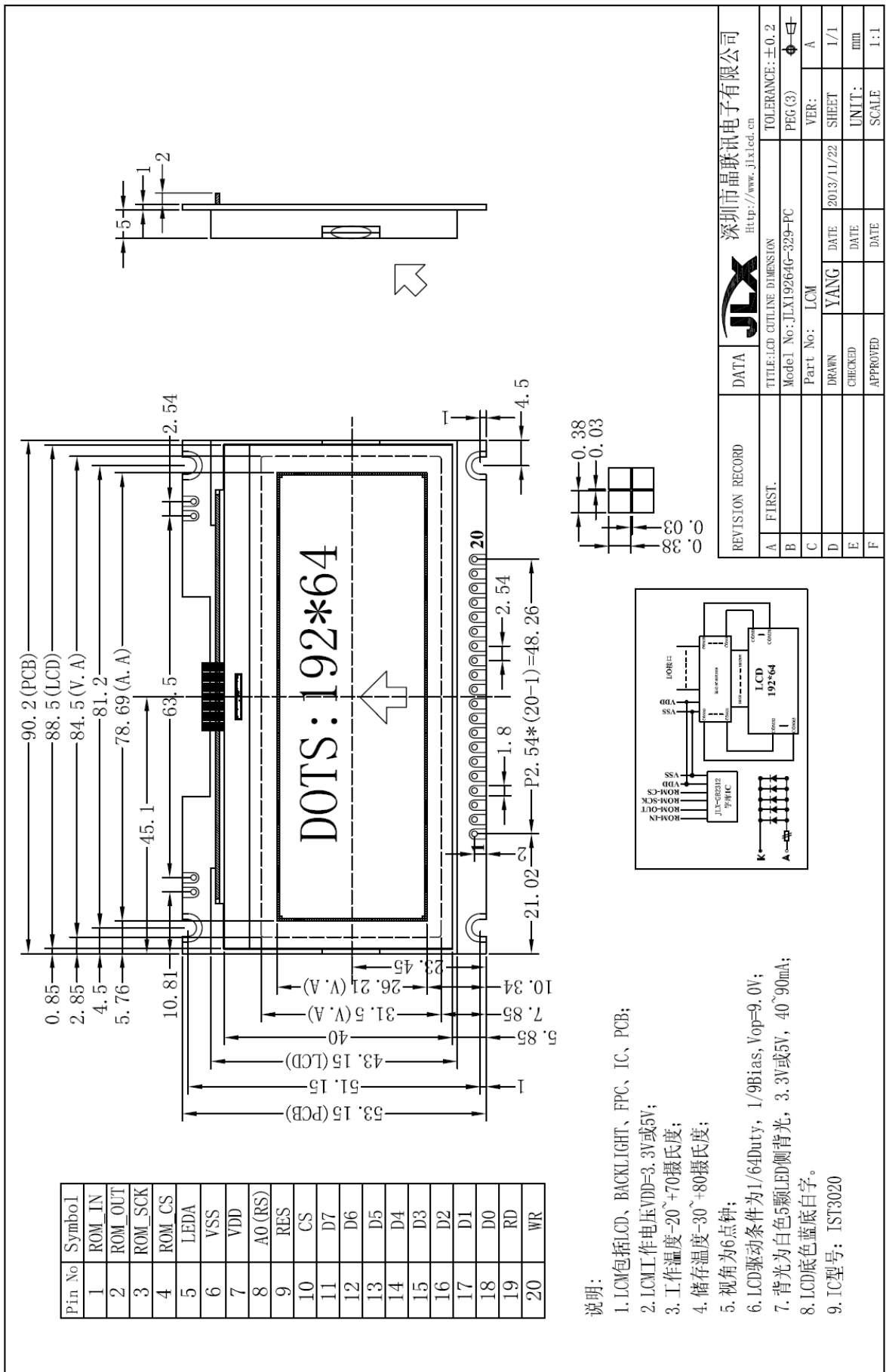


图 1. 液晶模块外形尺寸

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 192×64 点阵, 192 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 工作电图:

图 2 是 JLX19264G-329 图像点阵型模块的电路框图, 它由驱动 IC IST3020 及几个电阻电容组成。

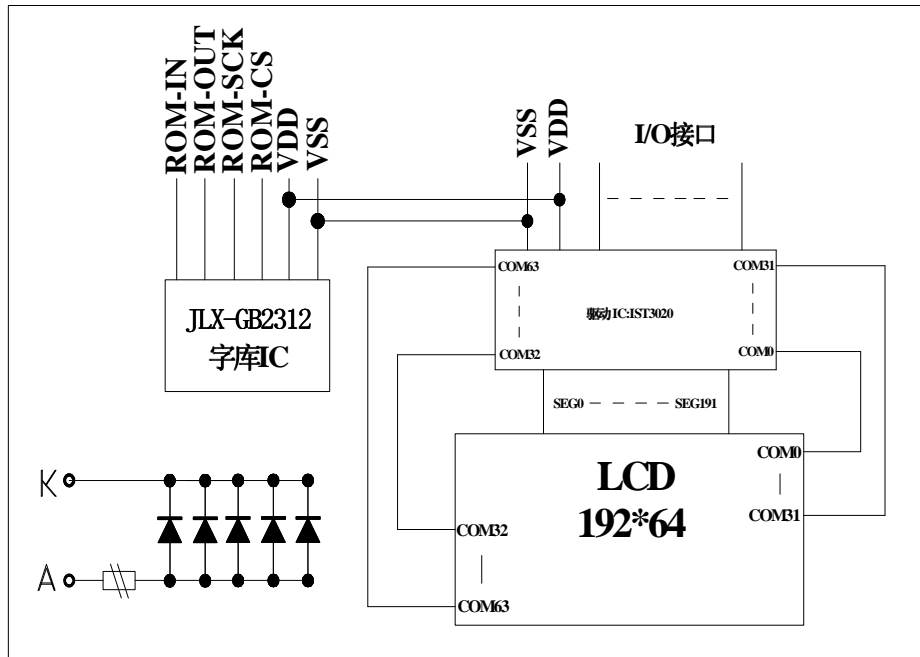


图 2: JLX19264G-329 图像点阵型液晶模块的电路框图

### 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:  $(8 \sim 20) * 5 = 40 \sim 100\text{mA}$  (LED 灯数共 5 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		7.0	V
LCD 驱动电压	VDD - V0	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	3.6	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	VIH	-	2.2		VDD	V
输入低电平	VIO	-	-0.3		0.6	V
输出高电平	VOH	IOH = 0.2mA	2.4		-	V
输出低电平	VOO	IOO = 1.2mA	-		0.4	V
模块工作电流	IDD	VDD = 3.3V	-		0.3	mA
背光工作电流	ILED	VLED=3.0V	40	60	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 IST3020 (Writing Data from CPU to IST3020)

The 4-line SPI Interface

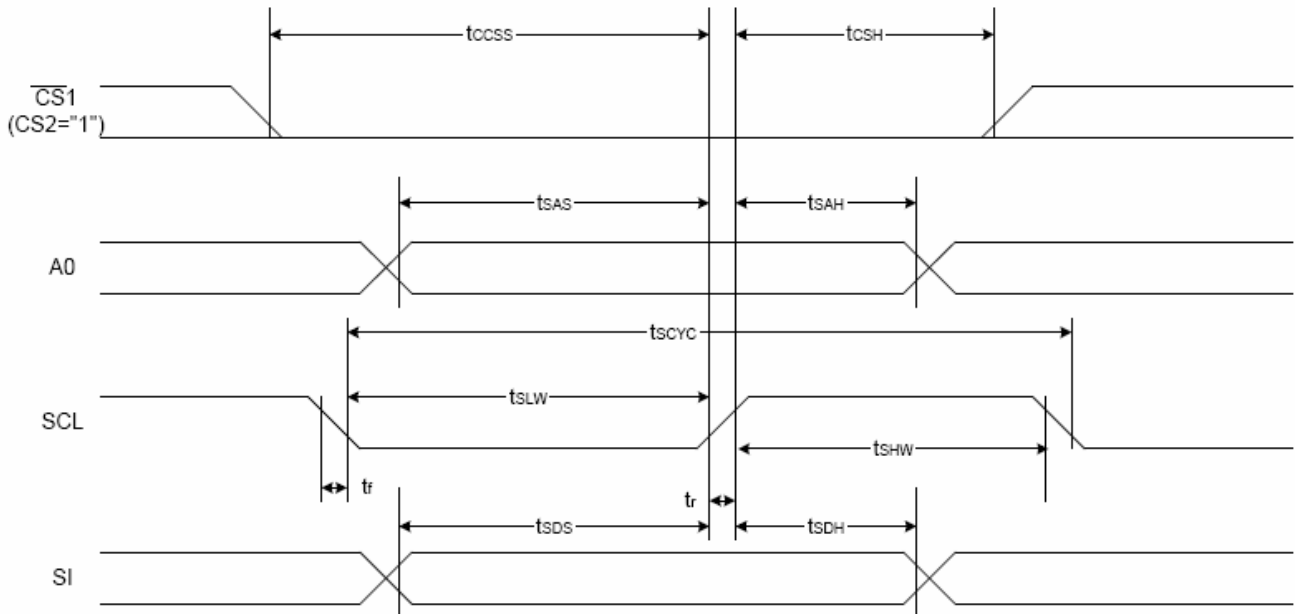


图 4. 从 CPU 写到 IST3020 (Writing Data from CPU to IST3020)

### 6.2 串行接口: 时序要求 (AC 参数):

写数据到 IST3020 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	250	--	50	ns
保持SCK高电平脉宽 (SCK "H" pulse width)	$T_{shw}$	引脚: SCK	100			ns
保持SCK低电平脉宽 (SCK "L" pulse width)	$T_{slw}$	引脚: SCK	100			ns
地址建立时间 (Address setup time)	$T_{sas}$	引脚: RS	150	--	--	ns
地址保持时间 (Address hold time)	$T_{sah}$	引脚: RS	150	--	--	ns
数据建立时间 (Data setup time)	$T_{sds}$	引脚: SID	100	--	--	ns
数据保持时间 (Data hold time)	$T_{sdh}$	引脚: SID	100	--	--	ns
片选信号建立时间 (CS-SCL time)	$T_{css}$	引脚: CS	150			ns

片选信号保持时间 (CS-SCL time)	$T_{csh}$	引脚: CS	150			ns
---------------------------	-----------	--------	-----	--	--	----

$VDD = 3.0V \pm 5\%$ ,  $T_a = 25^\circ C$

### 6.3 并行接口:

#### 从 CPU 写到 IST3020 (Writing Data from CPU to IST3020)

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

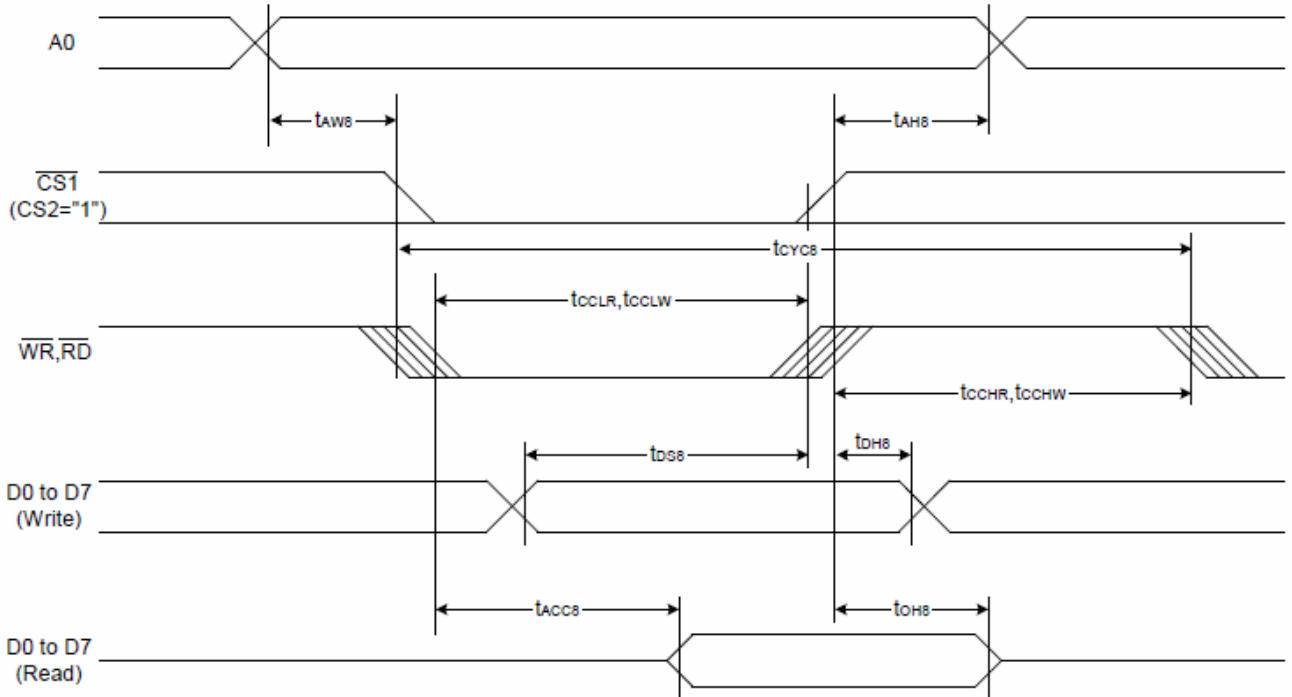
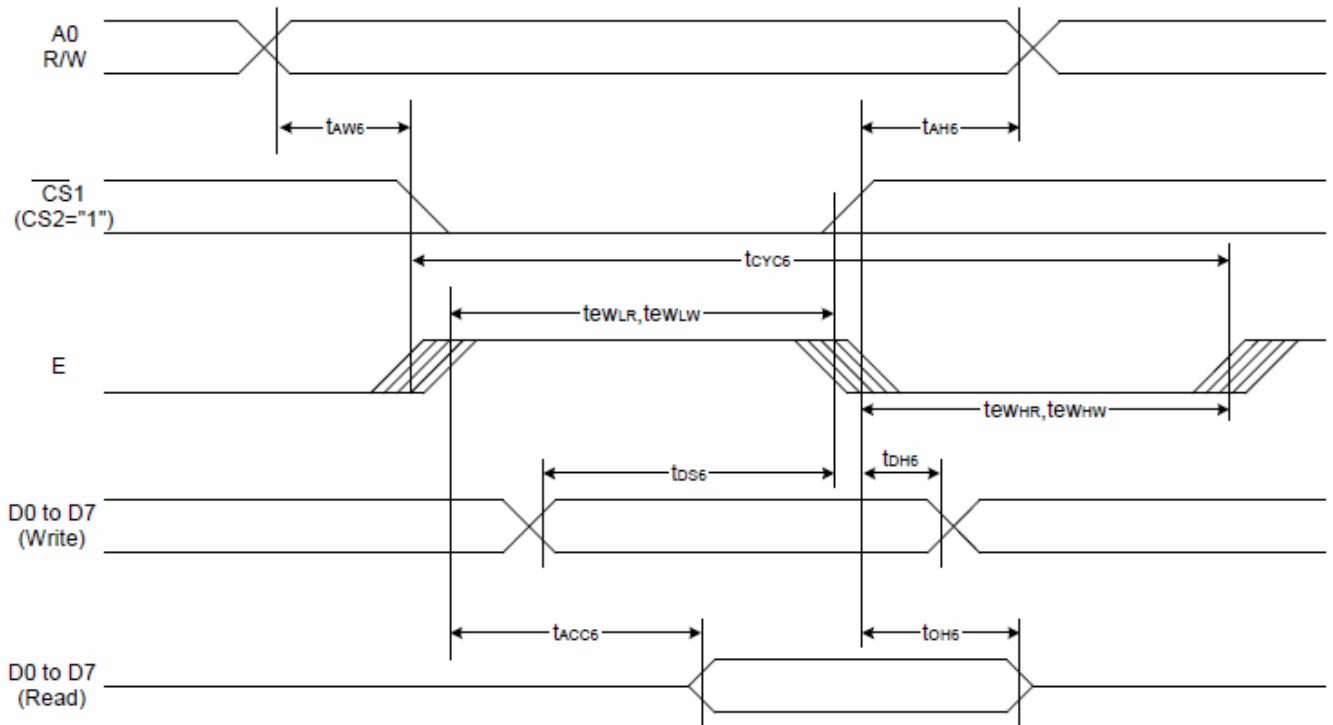


图 5. 从 CPU 写到 IST3020 (Writing Data from CPU to IST3020)

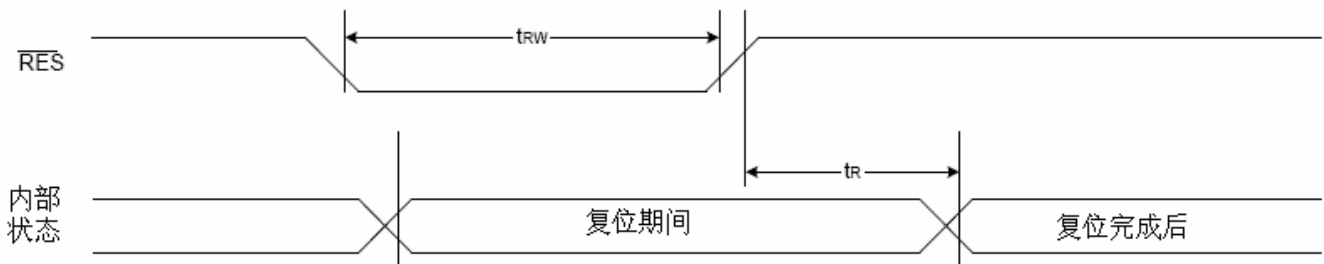


**System Bus Read/Write Characteristics 2 (For the 6800 Series MPU)**

**图 6. 从 CPU 写到 IST3020 (Writing Data from CPU to IST3020)**
**6.4 并行接口：时序要求 (AC 参数):**
**写数据到 IST3020 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	--	--	ns
系统循环时间		tCYC8	300	--	--	ns
使能“低”脉冲(写)	WR	tCCLW	60	--	--	ns
使能“高”脉冲(写)		tCCHW	60	--	--	ns
使能“低”脉冲(读)	RD	tCCLR	60	--	--	ns
使能“高”脉冲(读)		tCCHR	60	--	--	ns
写数据建立时间	D0-D7	tDS8	40		--	ns
写数据保持时间		tDH8	15		--	
读时间		tACC8	--		140	
读输出允许时间		tOH8	10		100	ns

**写数据到 IST3020 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	--	--	ns
地址建立时间		tAW6	0		--	ns
系统循环时间		tCYC6	300		--	ns
使能“低”脉冲(写)	WR	tEWLW	60	--	--	ns
使能“高”脉冲(写)		tEWHW	60	--	--	ns
使能“低”脉冲(读)	RD	tEWLR	60	--	--	ns
使能“高”脉冲(读)		tEWHR	60	--	--	ns
写数据建立时间	D0-D7	tDS6	40		--	ns
写数据保持时间		tDH6	15		--	
读时间		tACC6	--			
读输出允许时间		tOH6	60			ns

**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

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: 0:关, 1: 开
(2)显示初始行设置 (Display start line set)	0	0	1	显示初始行地址, 共 5 位						设置显示存储器的显示初始行
(3)页地址设置 (Page address set)	0	1	0	1	1	显示页地址, 共 4 位				设置显示页地址(注: 每 8 行为一个页, 64 行分为 8 个页, 例 0000 为第一页, 0001 为第二页)
(4) 列地址高4位 设置 列地址低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, 分别指定 128 列中任对应列。本液晶模块的第一列的地址为 00000000, 所以此指令表达为: 0x10, 0x00
		0	0	0	0	列地址的低 4 位				
(5) 读状态 (Status read)	0	状态				0	0	0	0	在本型号液晶模块不用此指令
(6)写数据( Display data write)	1	8 位显示数据								从 CPU 写数据到液晶模块
(7)读数据( Display data read)	1	8 位显示数据								在本型号液晶模块不用此指令
(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: 0: 常规: 从左到右, 1: 反转: 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: 0:常规: 正显 1: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: 0:常规 1: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: 0: 1/9 BIAS 1: 1/7BIAS
(12) Read-modify-write	0	1	1	1	0	0	0	0	0	Column address increment At write: +1

										At read: 0
13) 退出上述指令 (End)	0	1	1	1	0	1	1	1	0	退出上述“read/modify/write”指令
(14) 软件复位 (Reset)	0	1	1	1	0	0	0	1	0	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: 0: 普通顺序 1: 反向扫描
(16) 电源控制 (Power control set)		0	0	1	0	1	<b>电压操作模式选择, 共 3 位</b>			选择内部电压供应操作模式
(17) 选择内部电阻比例	0	0	0	1	0	0	<b>内部电压值电阻设置</b>			选择内部电阻比例 (Rb/Ra), 本液晶模块通过外置电阻设置, 此指令失效
(18) 内部设置液晶电压模式设置的电压值	0	1	0	0	0	0	0	0	1	设置内部电阻微调, 以设置液晶电压, 此两个指令需紧接着使用
		0	0	<b>6 位电压值数据, 0~63 共 64 级</b>						
(19) 静态图标显示: 开/关	0	1	0	1	0	1	1	0	0	0: 关, 1: 开。本液晶屏无此图标, 所以此指令无效
(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”。详细请看 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“IST3020.PDF”的第 32~46 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 192\*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。下图摘自 IST3020 IC 资料, 可通过“IST3020.PDF”之第 18 页获取最佳效果。



液晶模块与 MPU(以 8051 系列单片机为例)接口图如下:

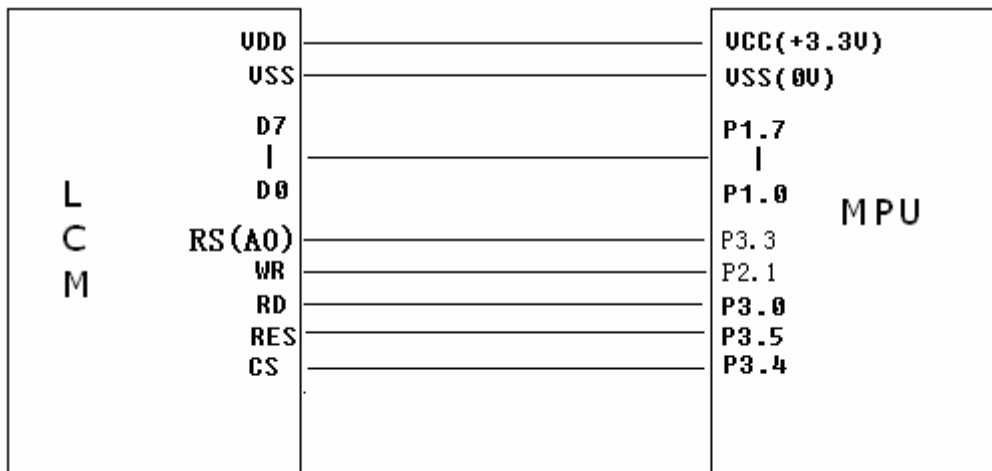


图 8. 并行接口

### 并程序序:

```

/* Test program for JLX19264G-329, 并行接口
   驱动 IC 是:IST3020(or compatible)
   晶联讯电子: 网址 http://www.jlxlcd.cn;
*/
#include <reg51.h>
#include <intrins.h>
#include <ctype.h>

sbit cs1=P3^4; /*接口定义*/
sbit reset=P3^5; /*接口定义*/
sbit rs=P3^3; /*接口定义*/
sbit rd=P3^0; /*接口定义*/
sbit wr=P2^1; /*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*/

#define uchar unsigned char
#define uint unsigned int
#define ulong unsigned long
uchar code bmp_19264[];

uchar code zhuang1[];
uchar code tai1[];
uchar code shi1[];
uchar code yong1[];
    
```

```

void delay(int i);
void delay_us(int i) ;

//=====transfer command to LCM=====
void transfer_command(int data1)
{
    cs1=0;
    rs=0;
    wr=0;
    rd=0;
    delay_us(2);
    P1=data1;
    rd=1;
    delay_us(2);
    cs1=1;
    rd=0;
}

//-----transfer data to LCM-----
void transfer_data(int data1)
{
    cs1=0;
    rs=1;
    wr=0;
    rd=0;
    P1=data1;
    rd=1;
    cs1=1;
    rd=0;
}

/*延时*/
void delay(int i)
{
    int j,k;
    for(j=0;j<i;j++)
    {
        for(k=0;k<109;k++); //1ms
    }
}

/*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(1000);
}

/*LCD 模块初始化*/
void initial_lcd()
{
    cs1=0;
    rd=0;
    reset=0;          /*低电平复位*/
    delay(20);
    reset=1;          /*复位完毕*/
    delay(10);
    transfer_command(0xAB); //开启内部晶振电路
    delay(10);
    transfer_command(0x2C); //升压步骤 1
    delay(20);
    transfer_command(0x2E); //升压步骤 2
    delay(20);
    transfer_command(0x2F); //升压步骤 3
    delay(20);
    transfer_command(0xA2); //BIAS 设置
    transfer_command(0x24); //粗调对比度
    transfer_command(0x81); //微调对比度
    transfer_command(0x14); //微调对比度的值: 从 0x00 到 0x3f
    transfer_command(0x40);
    transfer_command(0xA1); //列扫描顺序: 从左到右
    transfer_command(0xC0); //行扫描顺序: 从左到右
    transfer_command(0xAF);
}

void lcd_address(uchar page,uchar column)
{
    cs1=0;
    column=column+31;          //

```



```

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;
    cs1=0;
    for(i=0; i<9; i++)
    {
        lcd_address(1+i, 1);
        for(j=0; j<224; 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++)
    {
        cs1=0;
        lcd_address(j+1, 1);
        for(i=0; i<192; i++)
        {
            transfer_data(data1);
            transfer_data(data2);
        }
    }
}
    
```

```
void display_graphic_16x192(uchar page, uchar column, uchar data1)
{
    int i, j;
    for(j=0; j<2; j++)
    {
        cs1=0;
        lcd_address(page+j, column);
    }
}
    
```

```

        for(i=0;i<192;i++)
        {
            transfer_data(data1);
        }
    }
}

void display_graphic_16x16(uchar reverse,uchar page,uchar column,uchar *dp)
{
    int i,j;
    cs1=0;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j,column);
        for(i=0;i<16;i++)
        {
            if(reverse==1)
                transfer_data(*dp);
            else
                transfer_data(~*dp);
            dp++;
        }
    }
    cs1=1;
}

void display_graphic_19264(uchar page,uchar column,uchar *dp)
{
    int i,j;
    for(j=0;j<8;j++)
    {
        cs1=0;
        lcd_address(page+j,column);
        for(i=0;i<192;i++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}

void main(void)
{
    while(1)

```

```

{
    cs1=0;
    initial_lcd();
    clear_screen(); //clear all dots
    display_graphic_19264(1, 1, bmp_19264);
    waitkey();
    clear_screen();
    display_graphic_16x192(1, 1, 0xff);
    display_graphic_16x16(0, 1, (16*3), zhuang1);
    display_graphic_16x16(0, 1, (16*4), tail);
    waitkey();
    test_display(0xff, 0xff); //全显示
    waitkey();
    test_display(0xff, 0x00); //横线 1
    waitkey();
    test_display(0x00, 0xff); //横线 2
    waitkey();
    test_display(0xaa, 0xaa); //竖线 1
    waitkey();
    test_display(0x55, 0x55); //竖线 2
    waitkey();
}
}

uchar code zhuang1[]={
/*-- 文字: 状 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x08, 0x30, 0x00, 0xFF, 0x20, 0x20, 0x20, 0x20, 0xFF, 0x20, 0xE1, 0x26, 0x2C, 0x20, 0x20, 0x00,
0x04, 0x02, 0x01, 0xFF, 0x40, 0x20, 0x18, 0x07, 0x00, 0x00, 0x03, 0x0C, 0x30, 0x60, 0x20, 0x00};

uchar code tail[]={
/*-- 文字: 态 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x00, 0x04, 0x04, 0x04, 0x84, 0x44, 0x34, 0x4F, 0x94, 0x24, 0x44, 0x84, 0x84, 0x04, 0x00, 0x00,
0x00, 0x60, 0x39, 0x01, 0x00, 0x3C, 0x40, 0x42, 0x4C, 0x40, 0x40, 0x70, 0x04, 0x09, 0x31, 0x00};

uchar code bmp_19264[]={
/*-- 调入了一幅图像: E:\work\图片收藏夹\JLX19264G-803.bmp --*/
/*-- 宽度 x 高度=192x64 --*/
0xFF, 0x81, 0x11, 0x21, 0x01, 0x41, 0x31, 0x91, 0x51, 0x11, 0x11, 0x51, 0x91, 0x51, 0x31, 0x01,
0x01, 0x81, 0x81, 0xF9, 0x81, 0x81, 0x01, 0xF9, 0x01, 0x01, 0xF1, 0x01, 0x01, 0xF9, 0x01, 0x01,
0x01, 0x21, 0x21, 0x21, 0x21, 0x21, 0x29, 0xF1, 0x21, 0x21, 0x21, 0x21, 0x21, 0x31, 0x21, 0x01,
0x01, 0x01, 0x01, 0x01, 0x01, 0xF1, 0x51, 0x51, 0x51, 0x51, 0x51, 0xF9, 0x11, 0x01, 0x01, 0x01,
0x01, 0x11, 0xF1, 0x91, 0x91, 0xF1, 0x11, 0x91, 0x89, 0xB1, 0x81, 0xA1, 0x99, 0x81, 0x81, 0x01,

```

0x01, 0x01, 0x09, 0x11, 0x31, 0x01, 0x11, 0x11, 0xD1, 0x11, 0x11, 0x11, 0xF9, 0x11, 0x01, 0x01,  
0x01, 0x01, 0x01, 0xC1, 0x41, 0x41, 0x41, 0xF9, 0x41, 0x41, 0x41, 0x41, 0xE1, 0x41, 0x01, 0x01,  
0x01, 0x01, 0x01, 0x11, 0x11, 0x11, 0x11, 0x11, 0x91, 0x91, 0x51, 0x31, 0x19, 0x11, 0x01, 0x01,  
0x01, 0x21, 0x21, 0x21, 0x21, 0xA1, 0x61, 0x39, 0x21, 0x21, 0x21, 0xA1, 0x21, 0x31, 0x21, 0x01,  
0x01, 0x01, 0xF1, 0x11, 0x91, 0x71, 0x01, 0xF1, 0x51, 0x51, 0x51, 0xF9, 0x11, 0x01, 0x01,  
0x01, 0x01, 0x01, 0x01, 0x81, 0x41, 0x31, 0x01, 0x01, 0x39, 0xC1, 0x01, 0x01, 0x01, 0x01, 0x01,  
0x01, 0x81, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0x91, 0xD1, 0x91, 0x11, 0xF9, 0x11, 0x01, 0xFF,  
0xFF, 0x10, 0xF3, 0x0C, 0x03, 0x84, 0x45, 0x24, 0x14, 0xFF, 0x0C, 0x14, 0x24, 0xC5, 0x44, 0x00,  
0x00, 0x20, 0x60, 0x3F, 0x10, 0x90, 0x40, 0x3F, 0x00, 0x00, 0x7F, 0x00, 0x00, 0xFF, 0x00, 0x00,  
0x00, 0x00, 0x00, 0xFF, 0x01, 0x01, 0x01, 0xFF, 0x01, 0x01, 0x41, 0x81, 0x7F, 0x00, 0x00, 0x00,  
0x00, 0x00, 0xFC, 0xA4, 0xA4, 0xA5, 0xFD, 0x01, 0x01, 0xFD, 0xA5, 0xA5, 0xA4, 0xFC, 0x00, 0x00,  
0x00, 0x40, 0x7F, 0x24, 0x24, 0xFF, 0x10, 0x14, 0x84, 0x64, 0x1F, 0x24, 0x44, 0x84, 0x84, 0x00,  
0x00, 0x01, 0x01, 0x01, 0xFF, 0x40, 0x22, 0x02, 0xFF, 0x02, 0x02, 0x02, 0x02, 0x7F, 0x80, 0xE0, 0x00,  
0x00, 0x00, 0x00, 0x3F, 0x12, 0x12, 0x12, 0xFF, 0x12, 0x12, 0x12, 0x12, 0x1F, 0x00, 0xC0, 0x00,  
0x00, 0x02, 0x02, 0x02, 0x02, 0x02, 0x82, 0x02, 0xFF, 0x02, 0x02, 0x02, 0x02, 0x03, 0x02, 0x00,  
0x00, 0x08, 0x04, 0x02, 0x01, 0xFF, 0x15, 0x15, 0x15, 0x15, 0x15, 0xFF, 0x01, 0x00, 0x00, 0x00,  
0x00, 0x00, 0xFF, 0x10, 0x21, 0x1E, 0x00, 0xFF, 0x82, 0x9E, 0x22, 0x52, 0x8B, 0x84, 0x80, 0x00,  
0x00, 0x04, 0x02, 0x81, 0xC0, 0xA0, 0x98, 0x86, 0x80, 0x90, 0xA0, 0xC1, 0x82, 0x06, 0x02, 0x00,  
0x00, 0x00, 0x00, 0x3E, 0x12, 0x12, 0x12, 0x12, 0x3E, 0x00, 0x80, 0x00, 0xFF, 0x00, 0x00, 0xFF,  
0xFF, 0x80, 0x41, 0x20, 0xF8, 0x54, 0x50, 0x50, 0xF4, 0x59, 0x50, 0x50, 0x50, 0x10, 0x00, 0x00,  
0x00, 0x08, 0x88, 0xF8, 0x89, 0x88, 0x00, 0x08, 0x08, 0xF8, 0x08, 0x08, 0xF8, 0x0D, 0x08, 0x00,  
0x00, 0x00, 0x60, 0x58, 0x40, 0x40, 0xFC, 0x41, 0x40, 0x40, 0x44, 0x58, 0x40, 0x60, 0x40, 0x00,  
0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x01, 0x00, 0x00, 0x01, 0x00, 0x00,  
0x00, 0x00, 0x00, 0xC0, 0x78, 0x41, 0x40, 0x41, 0xFC, 0x40, 0x40, 0x40, 0x60, 0x41, 0x00, 0x00,  
0x00, 0x00, 0x10, 0x10, 0x10, 0x30, 0xD0, 0x14, 0x19, 0x90, 0x70, 0x10, 0x10, 0x10, 0x11, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00,  
0x00, 0x80, 0xE0, 0x9C, 0x90, 0x90, 0x10, 0xC9, 0x50, 0x40, 0x7C, 0x40, 0x50, 0xC8, 0x00, 0x00,  
0x00, 0x80, 0x40, 0x20, 0xFC, 0x51, 0x50, 0x54, 0xF8, 0x50, 0x51, 0x51, 0x58, 0x10, 0x00, 0x00,  
0x00, 0x00, 0x01, 0x08, 0x08, 0x08, 0x08, 0x09, 0xF8, 0x08, 0x08, 0x08, 0x0C, 0x89, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x80, 0x00, 0x00,  
0x00, 0x00, 0x80, 0xE0, 0x1C, 0x20, 0x20, 0x20, 0xA0, 0xFC, 0xA0, 0x21, 0x20, 0x30, 0x20, 0xFF,  
0xFF, 0x90, 0x90, 0x50, 0x57, 0x35, 0x15, 0x15, 0xFF, 0x15, 0x35, 0x55, 0x55, 0x94, 0x90, 0x00,  
0x00, 0x02, 0x7F, 0x10, 0x10, 0x1F, 0x81, 0x41, 0x31, 0x0F, 0x01, 0x01, 0xFF, 0x01, 0x01, 0x00,  
0x00, 0x80, 0x40, 0xA0, 0x98, 0x86, 0x41, 0x47, 0x29, 0x11, 0x29, 0x45, 0x43, 0xC0, 0x40, 0x00,  
0x00, 0x00, 0x08, 0x30, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x82, 0x81, 0x80, 0x84, 0x84, 0x84, 0x84, 0xFF, 0x84, 0x84, 0x84, 0x84, 0xC0, 0x80, 0x00,  
0x00, 0x80, 0x60, 0x1F, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00,  
0x00, 0x00, 0x08, 0x30, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x04, 0x04, 0xFF, 0x44, 0x24, 0x00, 0xFF, 0x0A, 0x0A, 0x0A, 0x4A, 0x8A, 0x7F, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0xFF, 0x95, 0x95, 0x95, 0x97, 0x95, 0x95, 0x95, 0xF5, 0x04, 0x00, 0x00,  
0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x41, 0x81, 0x7F, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00,  
0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x00,  
0x00, 0x01, 0x00, 0xFF, 0x10, 0x08, 0x14, 0x13, 0x10, 0xFF, 0x10, 0x13, 0x14, 0x08, 0x10, 0xFF,  
0xFF, 0x20, 0xC2, 0x0C, 0xC0, 0x08, 0xC8, 0x38, 0x08, 0x8A, 0x7C, 0x48, 0x48, 0xCC, 0x08, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x7C, 0x54, 0x54, 0x54, 0x54, 0x54, 0x7E, 0x04, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0xFC, 0x54, 0x54, 0x54, 0x54, 0x54, 0x54, 0xFE, 0x04, 0x00, 0x00,



0x00, 0x00, 0x40, 0x44, 0x44, 0x44, 0x44, 0x44, 0x44, 0x44, 0xC4, 0x44, 0x44, 0x44, 0x46, 0x64, 0x40, 0x00,  
0x00, 0x10, 0x90, 0xFE, 0x50, 0x90, 0x08, 0xE8, 0xBE, 0xA8, 0xA8, 0xBE, 0xA8, 0xEC, 0x08, 0x00,  
0x00, 0x20, 0x20, 0xFE, 0x20, 0x20, 0x20, 0x10, 0x10, 0xFE, 0x10, 0x10, 0xF8, 0x10, 0x00, 0x00,  
0x00, 0x00, 0xF0, 0x18, 0x16, 0x10, 0xF0, 0x40, 0x20, 0x98, 0x16, 0x10, 0x10, 0xF8, 0x10, 0x00,  
0x00, 0x00, 0x04, 0x04, 0x04, 0x74, 0x54, 0x54, 0x56, 0x54, 0x54, 0x74, 0x04, 0x04, 0x04, 0x00,  
0x00, 0x28, 0x28, 0x28, 0xFC, 0x26, 0x24, 0x00, 0x40, 0x88, 0x30, 0x00, 0xFE, 0x00, 0x00, 0x00,  
0x00, 0x10, 0x10, 0x10, 0xFE, 0x90, 0x00, 0x90, 0x90, 0x90, 0xFE, 0x90, 0x90, 0x98, 0x10, 0x00,  
0x00, 0x00, 0x80, 0x40, 0x20, 0x10, 0x0C, 0x80, 0x00, 0x0E, 0x30, 0x40, 0x80, 0x80, 0x80, 0x00,  
0x00, 0x20, 0x24, 0xA4, 0xA4, 0xA4, 0xA4, 0xA4, 0x34, 0x24, 0x04, 0xFE, 0x04, 0x00, 0xFF,  
0xFF, 0x04, 0x7C, 0x03, 0x00, 0x01, 0x7F, 0x42, 0x41, 0x22, 0x14, 0x09, 0x16, 0x21, 0x40, 0x00,  
0x00, 0x00, 0x7F, 0x29, 0x29, 0x29, 0x7F, 0x00, 0x00, 0x7F, 0x29, 0x29, 0x29, 0x7F, 0x00, 0x00,  
0x00, 0x40, 0x42, 0x44, 0x58, 0x40, 0x7F, 0x40, 0x40, 0x7F, 0x50, 0x48, 0x44, 0x62, 0x40, 0x00,  
0x00, 0x00, 0x10, 0x08, 0x04, 0x03, 0x20, 0x40, 0x3F, 0x00, 0x01, 0x02, 0x04, 0x18, 0x00, 0x00,  
0x00, 0x06, 0x01, 0x7F, 0x00, 0x49, 0x48, 0x2B, 0x1A, 0x0E, 0x0A, 0x1A, 0x2A, 0x4B, 0x48, 0x00,  
0x00, 0x08, 0x18, 0x0F, 0x44, 0x44, 0x22, 0x12, 0x0E, 0x03, 0x06, 0x0A, 0x13, 0x62, 0x22, 0x00,  
0x00, 0x00, 0x7F, 0x21, 0x21, 0x21, 0x7F, 0x00, 0x00, 0x00, 0x23, 0x40, 0x20, 0x1F, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x7F, 0x01, 0x01, 0x1D, 0x15, 0x15, 0x15, 0x1D, 0x01, 0x41, 0x7F, 0x00, 0x00,  
0x00, 0x08, 0x06, 0x01, 0x7F, 0x01, 0x02, 0x04, 0x04, 0x05, 0x04, 0x04, 0x7F, 0x02, 0x02, 0x00,  
0x00, 0x02, 0x22, 0x41, 0x3F, 0x00, 0x40, 0x40, 0x23, 0x14, 0x08, 0x14, 0x22, 0x41, 0x40, 0x00,  
0x00, 0x01, 0x00, 0x20, 0x70, 0x28, 0x26, 0x21, 0x20, 0x24, 0x28, 0x30, 0x60, 0x01, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x0F, 0x04, 0x04, 0x04, 0x04, 0x0F, 0x00, 0x20, 0x40, 0x3F, 0x00, 0x00, 0xFF,  
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x04, 0xFC, 0x04, 0x04,  
0x00, 0x04, 0xFC, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x0C, 0x34, 0xC0, 0x34, 0x0C, 0x04,  
0x00, 0x00, 0x08, 0x08, 0xFC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0x84, 0x84, 0x84, 0x48, 0xF0,  
0x00, 0x00, 0x18, 0x04, 0x04, 0x84, 0x44, 0x38, 0x00, 0x00, 0xF0, 0x88, 0x44, 0x44, 0x4C, 0x80,  
0x00, 0x00, 0xC0, 0x20, 0x10, 0xFC, 0x00, 0x00, 0x00, 0xF0, 0x08, 0x04, 0x04, 0x04, 0x1C, 0x00,  
0x00, 0x40, 0x40, 0x40, 0x40, 0x40, 0x40, 0x00, 0x00, 0x38, 0xC4, 0x44, 0x44, 0xC4, 0x38,  
0x00, 0x00, 0xF0, 0x08, 0x04, 0x04, 0x08, 0xF0, 0x00, 0x00, 0x18, 0x04, 0x44, 0x44, 0xA4, 0x18,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x04, 0x04, 0xFC, 0x04, 0x04, 0x00, 0x00, 0x00, 0x38, 0x44, 0x44, 0x84, 0x84, 0x1C,  
0x00, 0x0C, 0x04, 0x04, 0xFC, 0x04, 0x04, 0x0C, 0x00, 0x00, 0x18, 0x04, 0x44, 0x44, 0xA4, 0x18,  
0x00, 0x00, 0xF0, 0x08, 0x04, 0x04, 0x08, 0xF0, 0x00, 0x00, 0x18, 0x04, 0x04, 0x84, 0x44, 0x38,  
0x00, 0x00, 0xF0, 0x08, 0x04, 0x04, 0x08, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,  
0xFF, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0xB0, 0xA0, 0xA0, 0xA0, 0x9F, 0x80, 0x80,  
0x80, 0x88, 0x8F, 0x88, 0x88, 0x88, 0x88, 0x88, 0x8C, 0x80, 0x88, 0x8C, 0x8B, 0x80, 0x8B, 0x8C, 0x88,  
0x80, 0x80, 0x88, 0x88, 0x8F, 0x88, 0x88, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x8C, 0x88, 0x88, 0x84, 0x83,  
0x80, 0x80, 0x8C, 0x8A, 0x89, 0x88, 0x88, 0x8C, 0x80, 0x80, 0x87, 0x88, 0x88, 0x88, 0x88, 0x87,  
0x80, 0x80, 0x81, 0x81, 0x89, 0x8F, 0x89, 0x89, 0x80, 0x83, 0x84, 0x88, 0x88, 0x89, 0x87, 0x81,  
0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x87, 0x88, 0x88, 0x88, 0x87,  
0x80, 0x80, 0x83, 0x84, 0x88, 0x88, 0x84, 0x83, 0x80, 0x80, 0x86, 0x88, 0x88, 0x88, 0x88, 0x87,  
0x80, 0x80, 0x8B, 0x87, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80,  
0x80, 0x80, 0x88, 0x88, 0x8F, 0x88, 0x88, 0x80, 0x80, 0x80, 0x8E, 0x88, 0x88, 0x88, 0x88, 0x87,  
0x80, 0x80, 0x80, 0x88, 0x8F, 0x88, 0x80, 0x80, 0x80, 0x80, 0x86, 0x88, 0x88, 0x88, 0x88, 0x87,  
0x80, 0x80, 0x83, 0x84, 0x88, 0x88, 0x84, 0x83, 0x80, 0x80, 0x8C, 0x8A, 0x89, 0x88, 0x88, 0x8C,  
0x80, 0x80, 0x83, 0x84, 0x88, 0x88, 0x84, 0x83, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0xFF

```
};
```

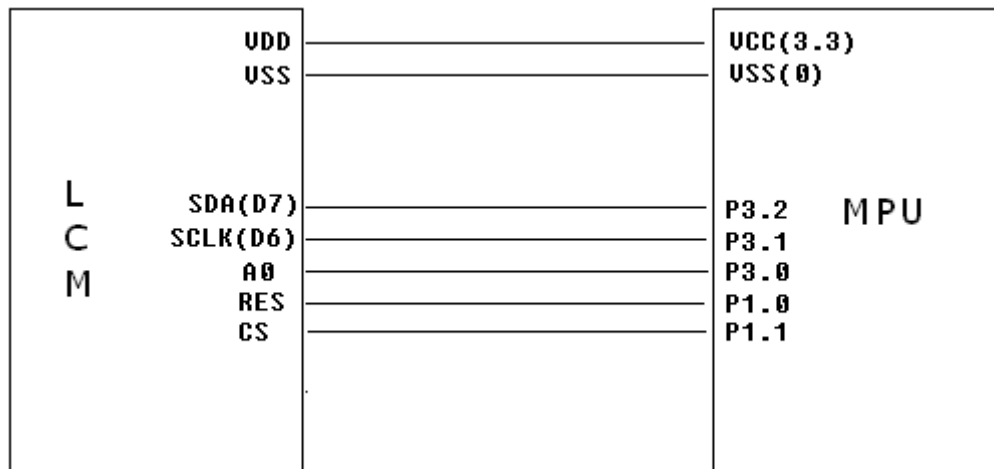


图 9. 串行接口

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=P3^3; /*接口定义:lcd_rs 就是 LCD 的 rs*/
sbit lcd_sclk=P1^6; /*接口定义:lcd_sclk 就是 LCD 的 sclk*/
sbit lcd_sid=P1^7; /*接口定义:lcd_sid 就是 LCD 的 sid*/
sbit reset=P3^5; /*接口定义:lcd_reset 就是 LCD 的 reset*/
sbit cs1=P3^4; /*接口定义:lcd_cs1 就是 LCD 的 cs1*/

void transfer_command(int data1)
{
    char i;
    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;
    }
}

/*写数据到 LCD 模块*/
void transfer_data(int data1)
{
    char i;
    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;  
  }  
}
```

