

# JLX12864G-139-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
6	时序特性	6~9
7	指令功能及硬件接口与编程案例	10~21

## 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-139-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-139-PN 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-139-PN 图像型点阵液晶模块的特性

2.1 结构牢。

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<200mW) ;

2.4 显示内容:

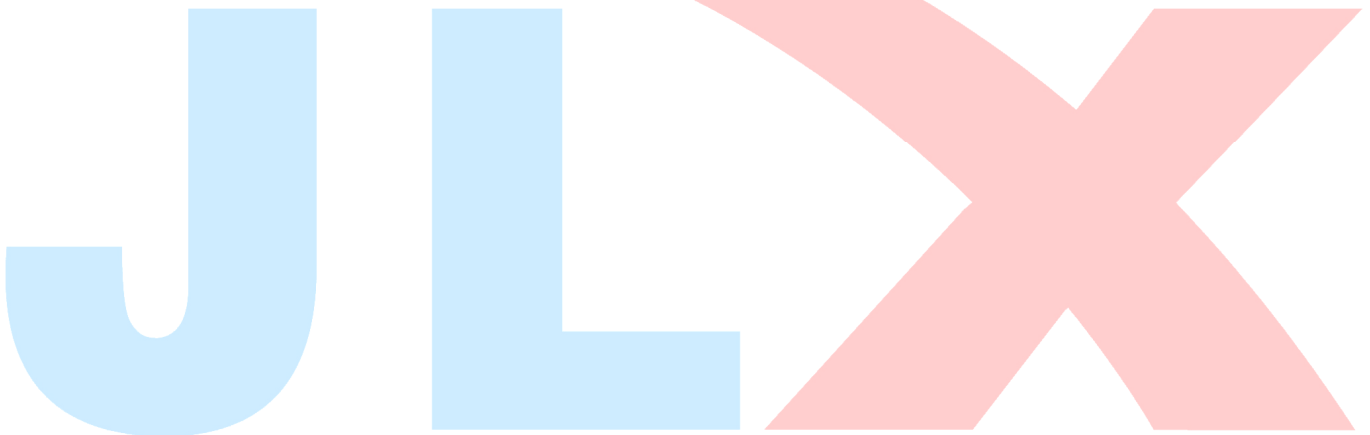
- 128\*64 点阵单色图片;

- 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。

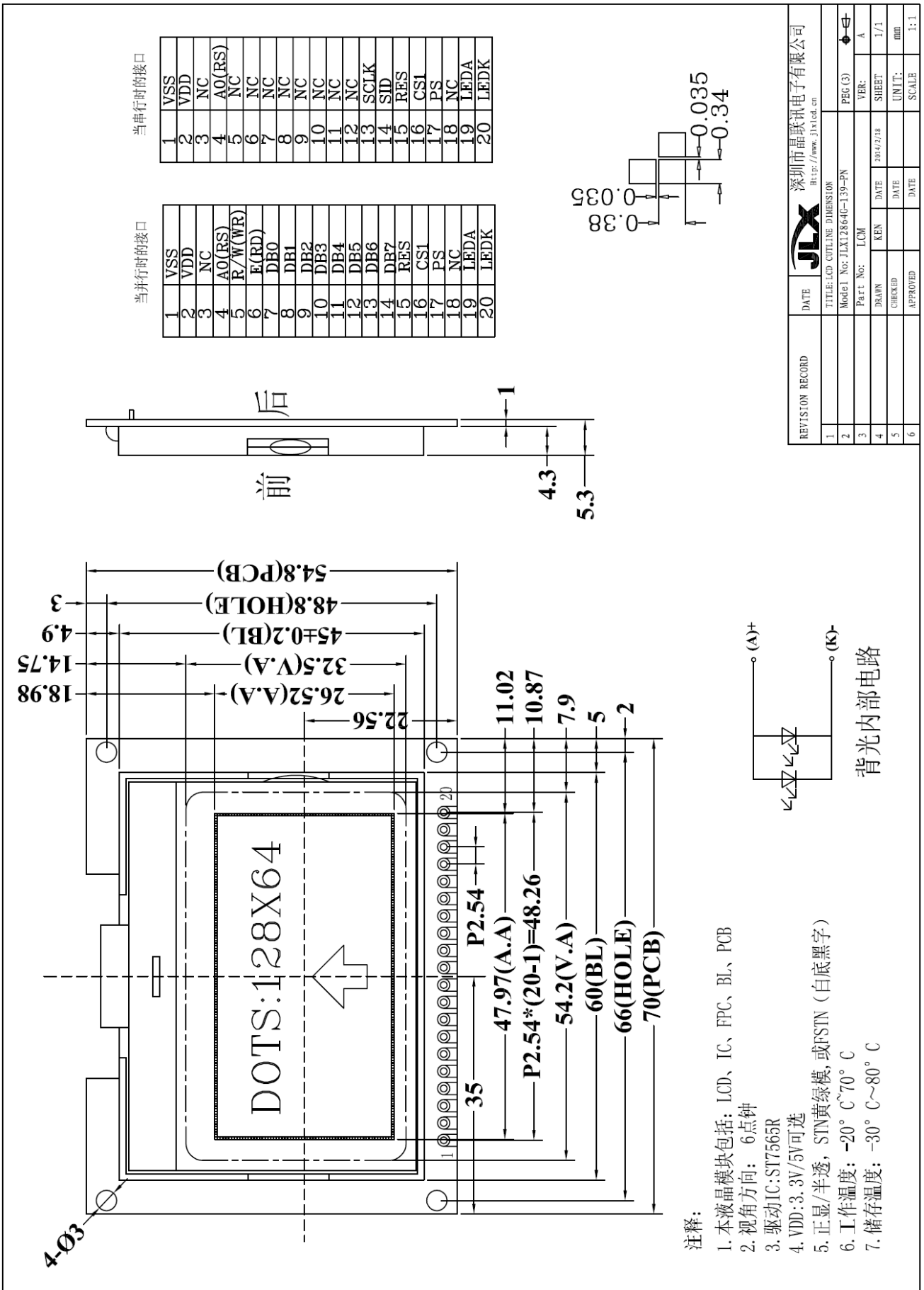
2.5 指令功能强:可组合成各种输入、显示、移位方式以满足不同的要求;

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



REVISION RECORD		DATE	DATE	DATE	DATE	DATE
1						
2	TITLE:LCD OUTLINE DIMENSION					
3	Model No: JLX12864G-139-PN					
4	Part No: LCM					
5	DRAWN	KEN	DATE	2014/2/18	SHEET	1/1
6	CHECKED		DATE		UNIT	mm
	APPROVED		DATE		SCALE	1:1

- 注释:
1. 本液晶模块包括: LCD、IC、FPC、BL、PCB
  2. 视角方向: 6点钟
  3. 驱动IC: ST7565R
  4. VDD: 3.3V/5V可选
  5. 正显/半透, STN黄绿模, 或FSTN (白底黑字)
  6. 工作温度: -20°C~70°C
  7. 储存温度: -30°C~80°C

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	VSS	接地	0V
2	VDD	电路电源	5V 或 3.3V
3	NC		空脚
4	A0 (RS)	寄存器选择信号	H:数据寄存器 0:指令寄存器
5	R/(WWR)	读/书	并行时: H:读数据 0:写数据 串行时: 空
6	(E)RD	使能信号	并行时: 使能信号 串行时: 空
7-12	D0-D5	I/O	并行时: 数据总线 DB0~DB5 串行时: 空
13	D6 (SCLK)	I/O	并行时: 数据总线 DB6 串行时: 串行时钟
14	D7 (SID)	I/O	并行时: 数据总线 DB7 串行时: 串行数据
15	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
16	CS	片选	低电平片选
17	PS	选串并控制接口	H:并行接口, L:串行接口 (一般不接, 因为 PCB 上已跳线接好)
18	NC		空脚
19	LEDA	背光电源	背光电源正极, 同 VDD 电压 (5V 或 3.3V)
20	LEDK	背光电源	背光电源负极

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)。

### 4.2 工作电图:

图 2 是 JLX12864G-139-PN 图像点阵型模块的电路框图, 它由驱动 IC ST7565R 及几个电阻电容组成。

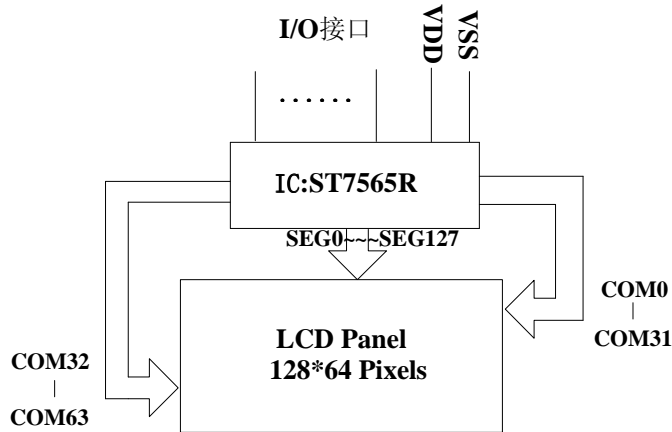


图 2: JLX12864G-139-PN 图像点阵型液晶模块的电路框图

### 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:  $-30 \sim +80^{\circ}\text{C}$ ;

正常工作电流为:  $(8 \sim 20) * 2 = 16 \sim 40\text{mA}$  (LED 灯数共 2 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		-10		+60	$^{\circ}\text{C}$
储存温度		-20		+70	$^{\circ}\text{C}$

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	VO0	I00 = 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	16	30	40	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

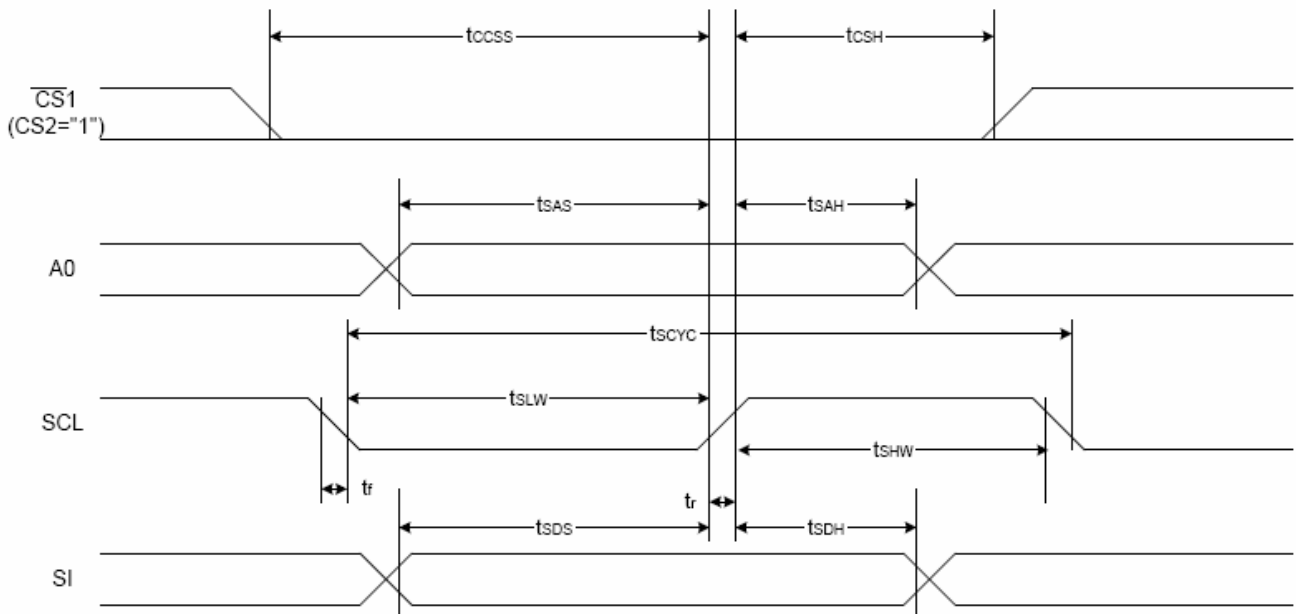


图 4. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

### 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

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

### 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)

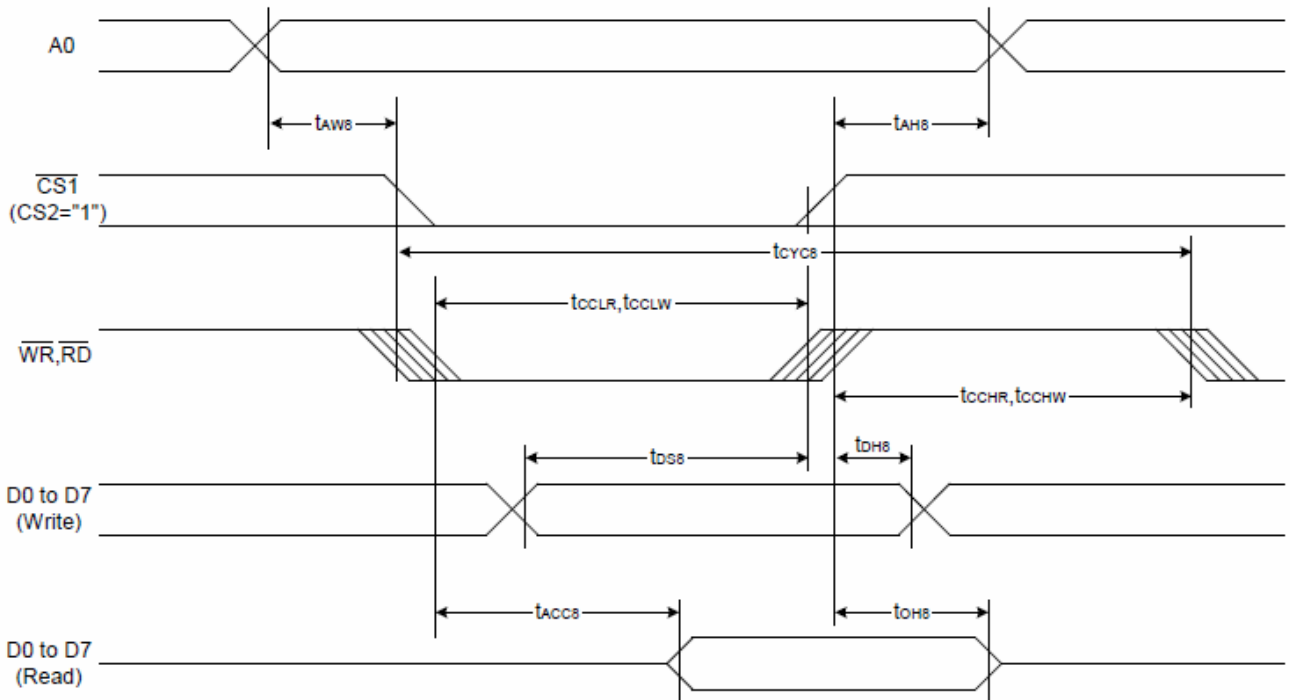


图 5. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

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

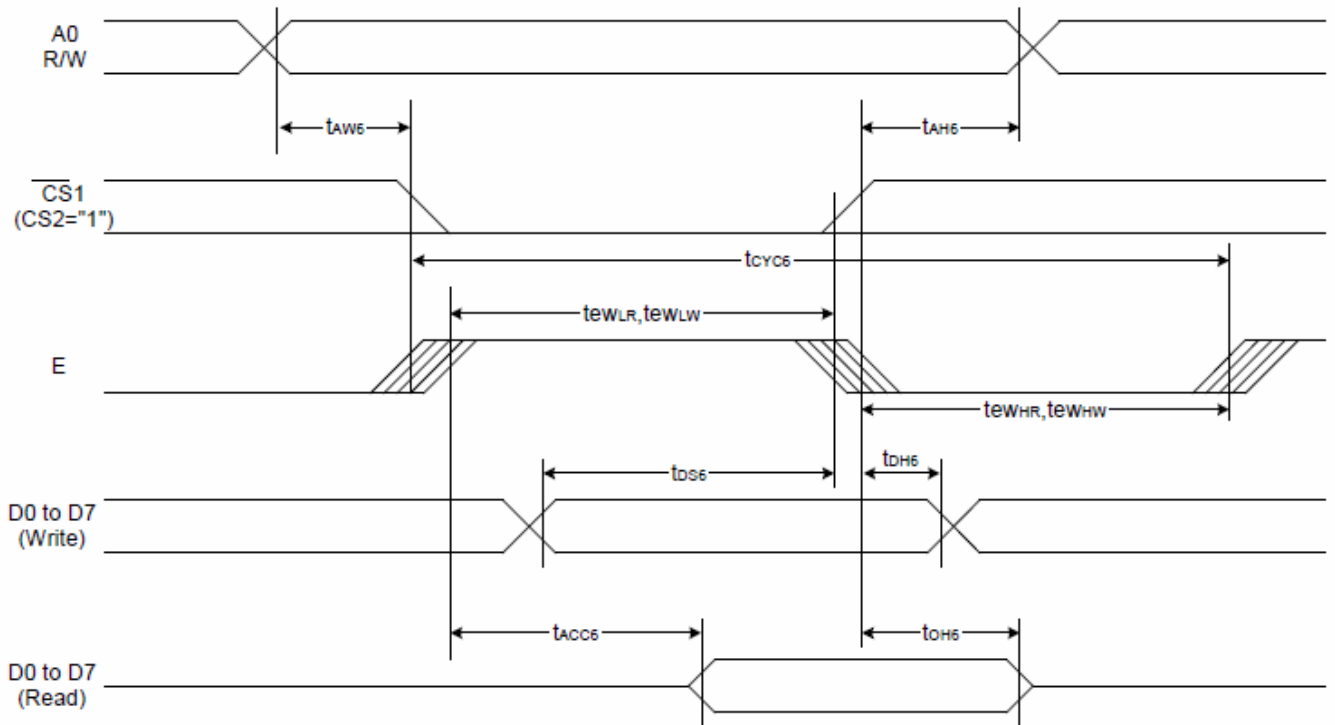


图 6. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

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	--	--	ns
系统循环时间		tCYC8	240	--	--	ns
使能“低”脉冲(写)	WR	tCCLW	80	--	--	ns
使能“高”脉冲(写)		tCCHW	80	--	--	ns
使能“低”脉冲(读)	RD	tCCLR	140	--	--	ns
使能“高”脉冲(读)		tCCHR	80	--	--	ns
写数据建立时间	D0-D7	tDS8	40	--	--	ns
写数据保持时间		tDH8	0	--	--	ns
读时间		tACC8	--	--	70	ns
读输出允许时间		tOH8	5	--	50	ns



**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	--	--	ns
地址建立时间		tAW6	0		--	ns
系统循环时间		tCYC6	240		--	ns
使能“低”脉冲（写）	WR	tEWLW	80	--	--	ns
使能“高”脉冲（写）		tEWHW	80	--	--	ns
使能“低”脉冲（读）	RD	tEWLR	80	--	--	ns
使能“高”脉冲（读）		tEWHR	140	--		ns
写数据建立时间	D0-D7	tDS6	40		--	ns
写数据保持时间		tDH6	0		--	
读时间		tACC6	--		70	
读输出允许时间		tOH6	5		50	ns

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

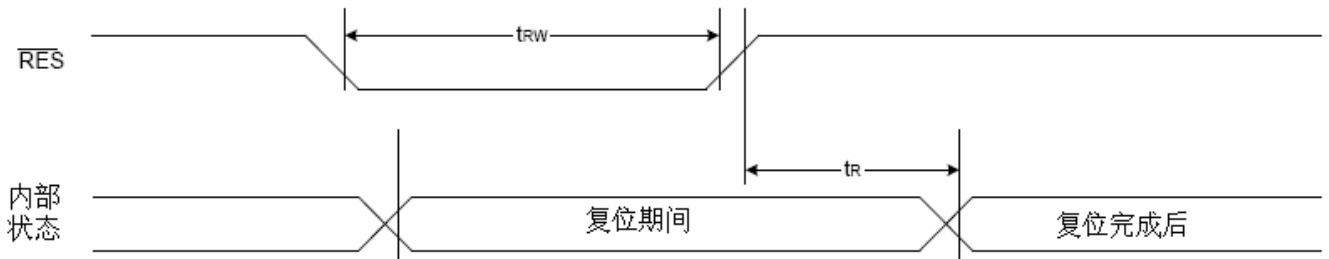


图 7：电源启动后复位的时序

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	$t_r$		--	--	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
----	-----	-----	-----	-----	-----	-----	-----	-----

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	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
	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	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！

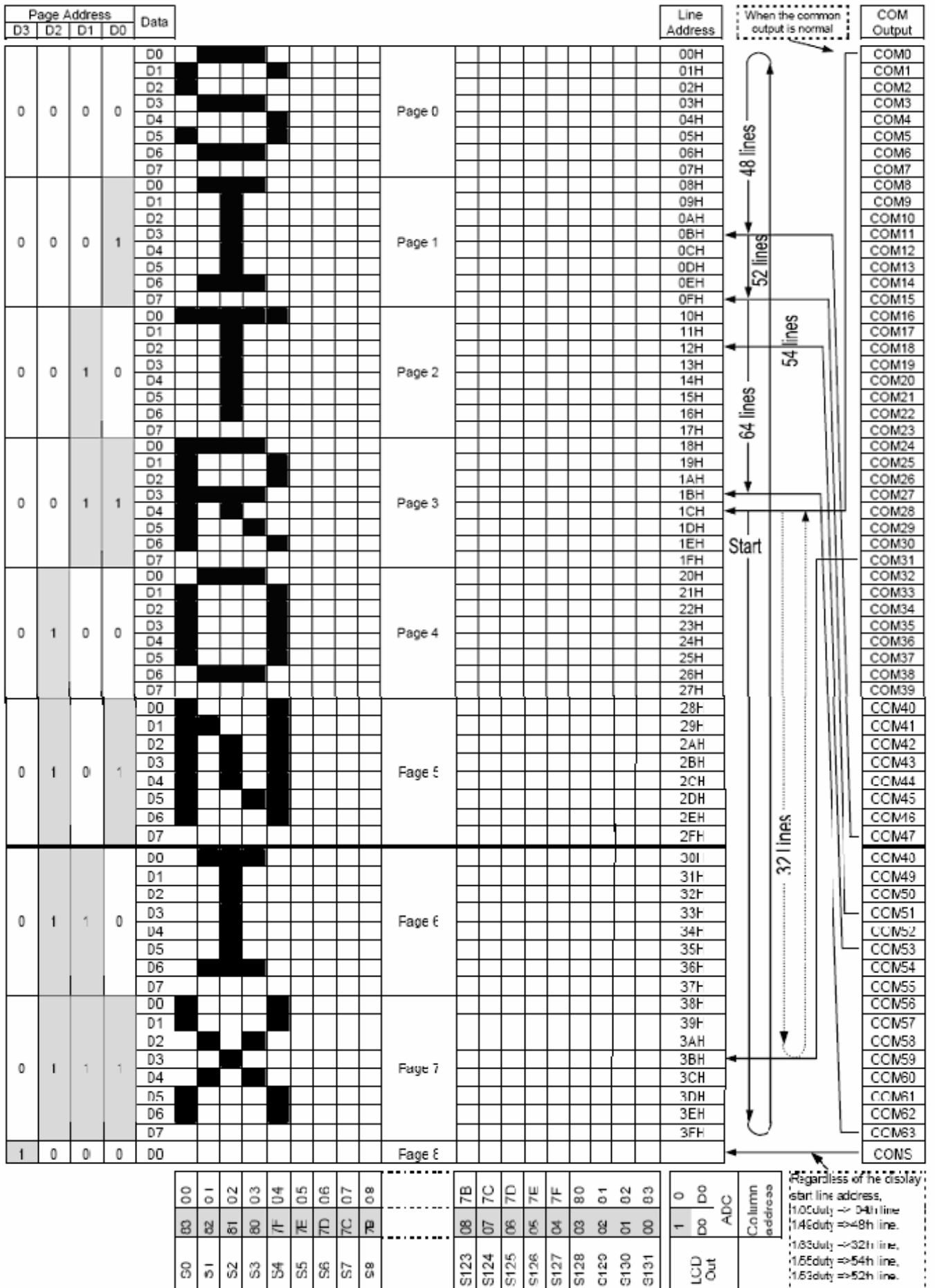
请详细参考 IC 资料“ST7565R\_V15.PDF”的第 42~49 页。

### 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\*32 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 ST7565R IC 资料，可通过“ST7565R\_V15. PDF”之第 27 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:

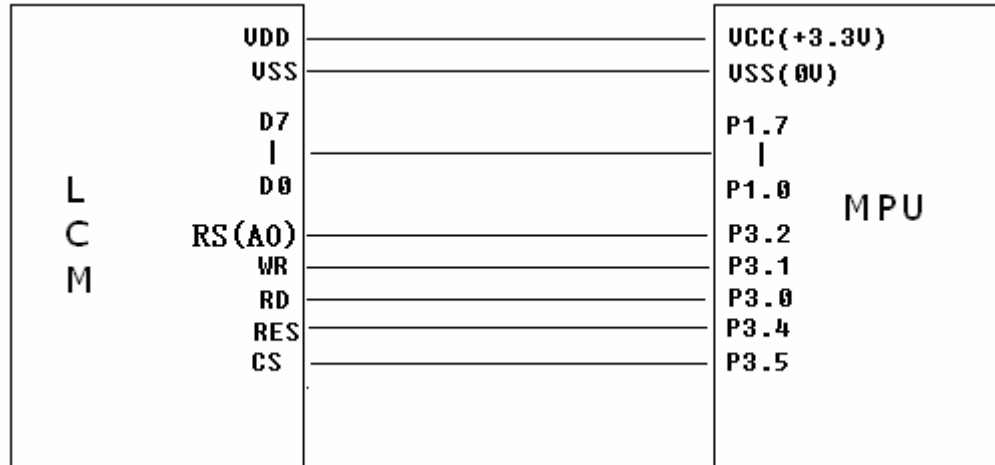


图 8. 并行接口

### 并程序序:

```

/* Test program for JLX12864G-139-PN, 并行接口
   驱动 IC 是:ST7565R(or competible)
   晶联讯电子: 网址 http://www.jlxlcd.cn; http://www.jlxlcd.com.cn
*/
#include <reg51.H>

sbit cs1=P3^5;    /*接口定义*/
sbit reset=P3^4; /*接口定义*/
sbit rs=P3^2;    /*接口定义*/
sbit rd=P3^0;    /*接口定义*/
sbit wr=P3^1;    /*接口定义。另外 P1.0~1.7 对应 DB0~DB7*/

void transfer_data(int data1);
void transfer_command(int data1);
char code graphic1[];
char code graphic2[];
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();
    }
}
/*LCD 初始化*/
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(0x1c); //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(0xc0); //行扫描顺序：从上到下*/
    transfer_command(0xa1); //列扫描顺序：从左到右*/
    transfer_command(0xaf); //开显示*/
}
//=====clear all dot martrics=====
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);
        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<110;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: D:\Backup\我的文档\My Pictures\12864-139 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, 0xF1, 0x01, 0xF1, 0x01, 0x01, 0x01, 0x01, 0x01, 0x11, 0x61,
0x81, 0x61, 0x11, 0x01, 0x01, 0x21, 0xF1, 0x01, 0x01, 0x01, 0x61, 0x11, 0x11, 0x11, 0xE1, 0x01,
0x61, 0x91, 0x91, 0x91, 0x61, 0x01, 0xC1, 0xA1, 0x91, 0x91, 0x21, 0x01, 0x01, 0xC1, 0x21, 0xF1,
```



```

0x01, 0x01, 0xE1, 0x11, 0x11, 0x11, 0x61, 0x01, 0x81, 0x81, 0x81, 0x81, 0x01, 0x01, 0x21,
0xF1, 0x01, 0x01, 0x01, 0x21, 0x11, 0x91, 0x91, 0x61, 0x01, 0x11, 0x11, 0x11, 0x91, 0x71, 0x01,
0x01, 0x21, 0xF1, 0x01, 0x01, 0x01, 0x61, 0x11, 0x11, 0x11, 0xE1, 0x01, 0x01, 0x01, 0x01, 0x01,
0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0xFF,
.
.
.
.
};
    
```

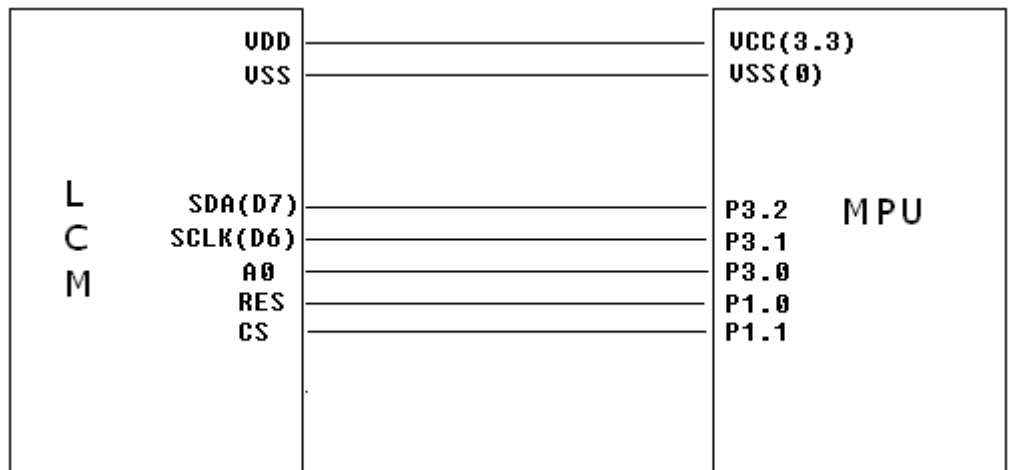


图 9. 串行接口

串程序序:

```

/* Test program for JLX12864G-139, 串行接口
Driver IC is:ST7565R(or competible)
Programmed by Ken, Dec. 24, 2010
JLX electronic Co., ltd, http://www.jlxlcd.cn;http://www.jlxlcd.com.cn
*/
#include <reg51.H>

sbit cs1=P1^1;
sbit reset=P1^0;
sbit rs=P3^0;
sbit sclk=P3^1;
sbit sid=P3^2;

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[];
    
```

```
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(graphic3); //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();
    }
}

/*LCD 初始化*/
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); /*升压步骤 1*/
    Delay(5);
    transfer_command(0x2e); /*升压步骤 2*/
    Delay(5);
    transfer_command(0x2f); /*升压步骤 3*/
    Delay(5);
    transfer_command(0x23); /*粗调对比度，可设置范围 0x20~0x27*/
}
```

```

transfer_command(0x81); /*微调对比度*/
transfer_command(0x1a); /*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(0xc0); /*行扫描顺序：从上到下*/
transfer_command(0xa1); /*列扫描顺序：从左到右*/
transfer_command(0xaf); /*开显示*/
}

//=====clear all dot martrics=====
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);
        for(j=0; j<128; j++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}

//=====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;
        Delay1(5);
        sclk=1;
        Delay1(5);
        data1=data1<<=1;
    }
}

```

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

```

void transfer_data(int data1)
{
    char i;
    cs1=0;
    rs=1;
    for(i=0;i<8;i++)
    {
        sclk=0;
        if(data1&0x80) sid=1;
        else sid=0;
        sclk=1;
        data1=data1<<=1;
    }
}

```

//=====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(1);
    if (P2&0x01) goto repeat;
    else;
}
char code graphic1[]={
/*-- 调入了一幅图像: D:\Backup\我的文档\My Pictures\12864-139 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, 0xF1, 0x01, 0xF1, 0x01, 0x01, 0x01, 0x01, 0x01, 0x11, 0x61,
0x81, 0x61, 0x11, 0x01, 0x01, 0x21, 0xF1, 0x01, 0x01, 0x01, 0x61, 0x11, 0x11, 0x11, 0xE1, 0x01,
0x61, 0x91, 0x91, 0x91, 0x61, 0x01, 0xC1, 0xA1, 0x91, 0x91, 0x21, 0x01, 0x01, 0xC1, 0x21, 0xF1,
0x01, 0x01, 0xE1, 0x11, 0x11, 0x11, 0x61, 0x01, 0x81, 0x81, 0x81, 0x81, 0x81, 0x01, 0x01, 0x21,
};
```

