

# JLX12864G-66011-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	6~10
7	指令功能及硬件接口与编程案例	11~尾页

## 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-66011-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-66011-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-66011-PN 图像型点阵液晶模块的特性

2.1 结构牢：背光带有挡墙。

2.2 IC 采用矽创公司 ST7567, 功能强大，稳定性好

2.3 功耗低：不带背光 1mW (3.3V\*0.3mA), 带背光不大于 150mW (3.3V\*45mA)

2.4 显示内容：

1) 128\*64 点阵单色图片，或其它小于 128\*64 点阵的单色图片；

(2) 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\*4 行；

(3) 按照 12\*12 点阵汉字来计算可显示 10 字\*4 行；

(4) 按照 8\*16 点阵汉字来计算可显示 16 字\*4 行；

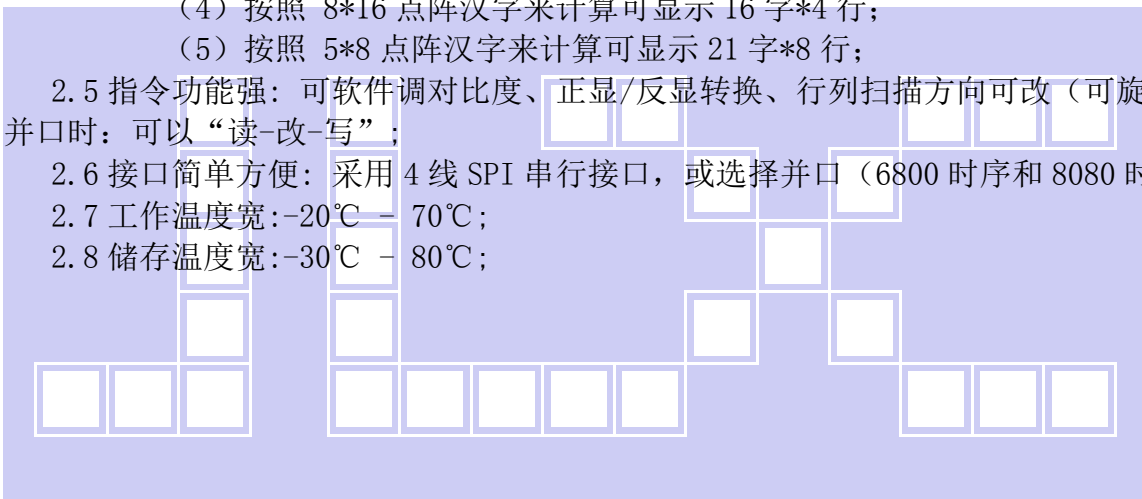
(5) 按照 5\*8 点阵汉字来计算可显示 21 字\*8 行；

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：采用 4 线 SPI 串行接口，或选择并口（6800 时序和 8080 时序可选）。

2.7 工作温度宽：-20℃ - 70℃；

2.8 储存温度宽：-30℃ - 80℃；





## 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	E (RD)	使能信号	并行时：使能信号 串行时：空
20	WR (R/W)	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  $128 \times 64$  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-66011-PN 图像点阵型模块的电路框图。

## 电路框图

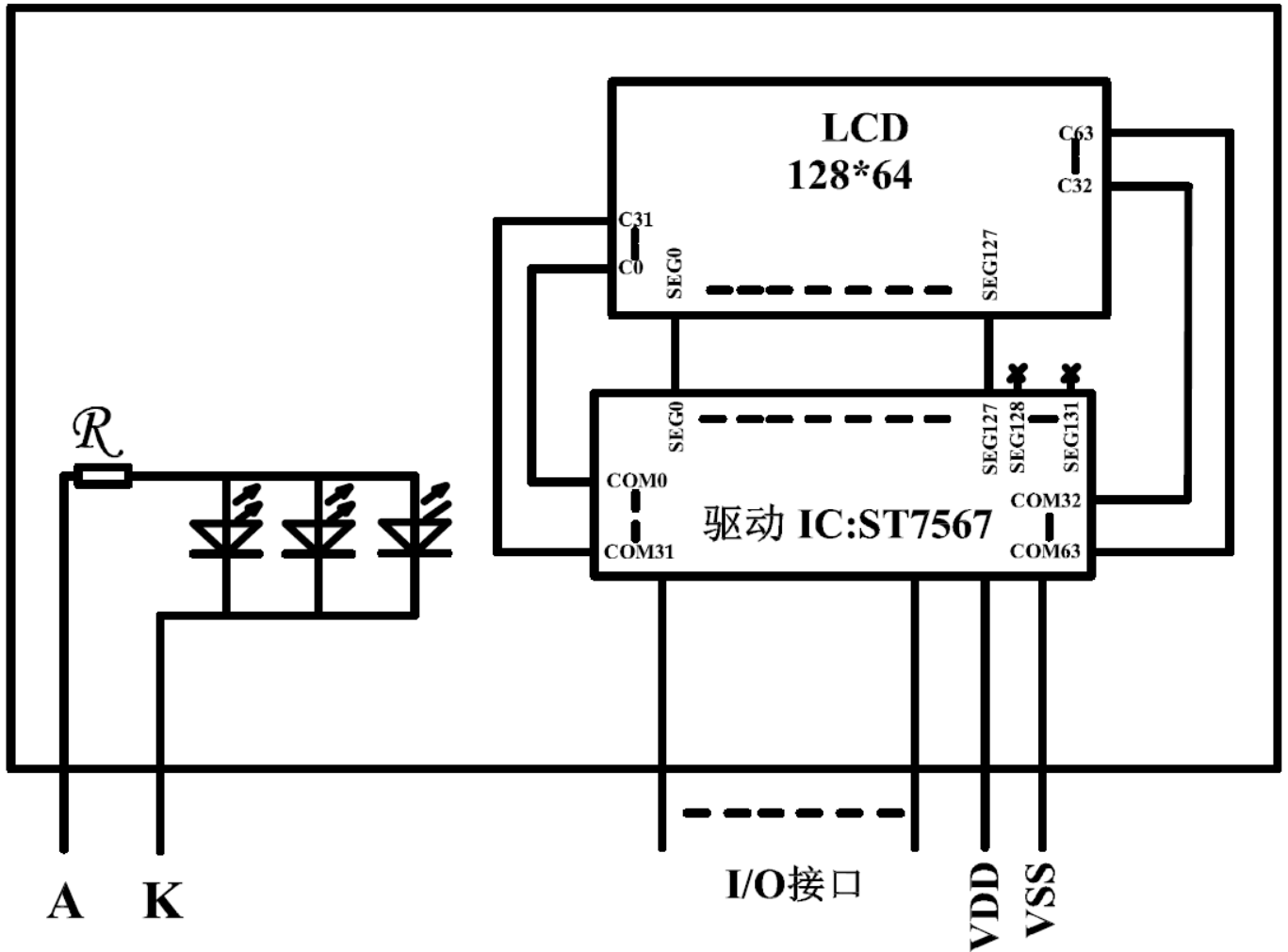


图 2: JLX12864G-66011-PN 图像点阵型液晶模块的电路框图

## 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:

工作温度:  $-20^{\circ}\text{C} \sim +70^{\circ}\text{C}$ ;

存储温度:  $-30^{\circ}\text{C} \sim +80^{\circ}\text{C}$ ;

背光板可选择绿色、白色。

正常工作电流为:  $24 \sim 60\text{mA}$  (LED 灯数共 3 颗);

工作电压:  $3.0\text{V}$ ; (PCB 已加限流电阻, 供电同 VDD 电压即可)

## 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	
电源	VDD - VSS	-0.3	—	3.6	V
LCD 驱动电压	V0 - XV0	-0.3		13.5	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	VDD	—	2.6	3.3	3.6	V
背光工作电压	VLED	—	2.9	3.0	3.1	V
输入高电平	VIH	—	0.8VDD	—	VDD	V
输入低电平	VIO	—	0	—	0.2VDD	V
输出高电平	VOH	IOH = 0.2mA	0.8VDD	—	VDD	V
输出低电平	VOO	I00 = 1.2mA	0	—	0.2VDD	V
模块工作电流	IDD	VDD = 3.0V	—	0.3	1.0	mA
背光工作电流	ILED	VLED=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)

System Bus Timing for 4-Line Serial Interface

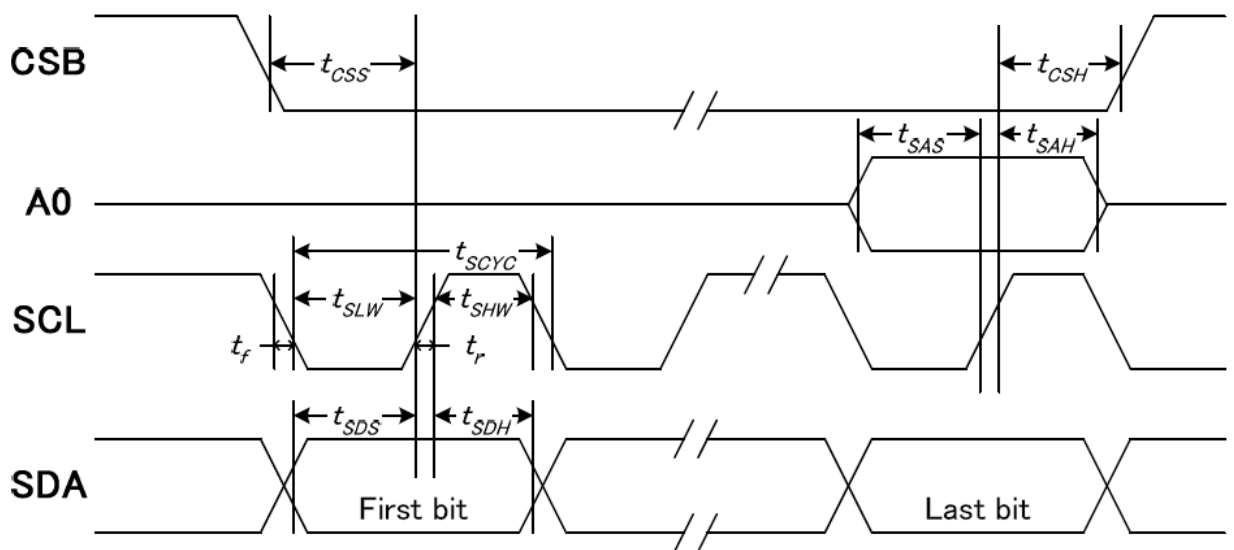


图 3. 从 CPU 写到 ST7567 (Writing Data from CPU to ST7567)

## 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)	T <sub>scyc</sub>	引脚: SCK	50	—	—	ns
保持SCK高电平脉宽 (SCK "H" pulse width)	T <sub>shw</sub>	引脚: SCK	25	—	—	ns
保持SCK低电平脉宽 (SCK "L" pulse width)	T <sub>slw</sub>	引脚: SCK	25	—	—	ns
地址建立时间 (Address setup time)	T <sub>sas</sub>	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

$VDD = 3.3V, T_a = 25^\circ C$

### 6.3 并行接口:

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

#### System Bus Timing for 6800 Series MPU

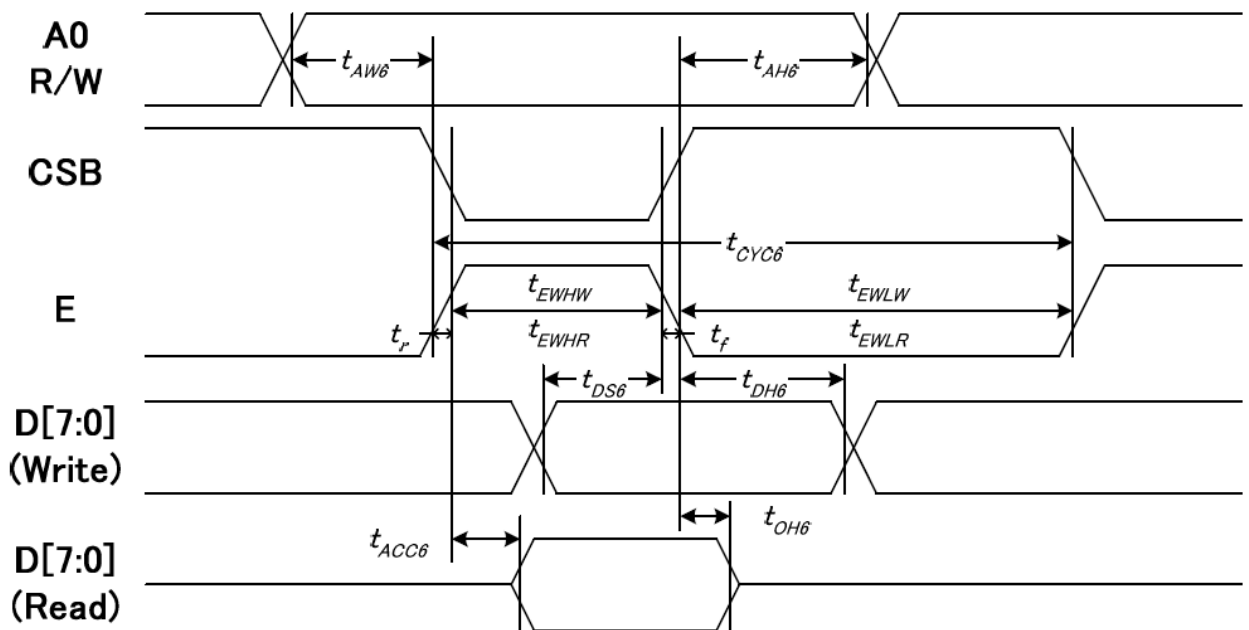


图 4. 从 CPU 写到 ST7567 (Writing Data from CPU to ST7567)

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

## System Bus Timing for 8080 Series MPU

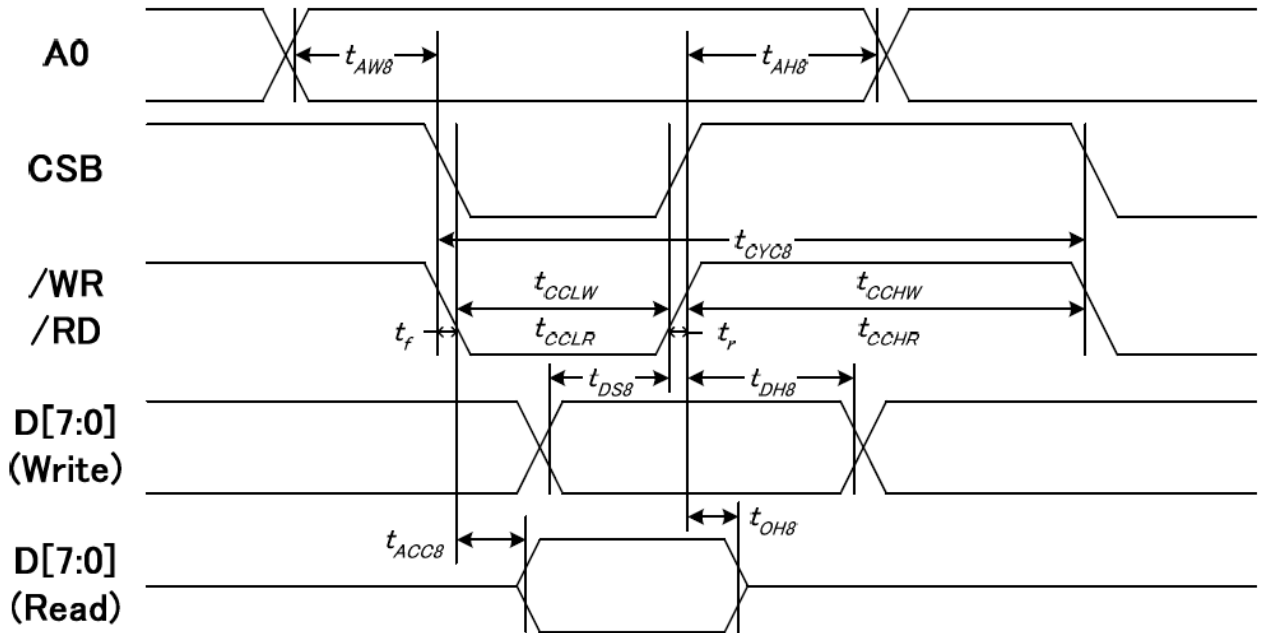


图 5. 从 CPU 写到 ST7567 (Writing Data from CPU to ST7567)

## 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)

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	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	tEWL R	80	—	—	
使能“高”脉冲(读)		tEWH R	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	
读输出允许时间		tOH6	5	—	50	

VDD=3.3V, Ta=25°C



**写数据到 ST7567 的时序要求: (6800 系列 MPU)**
**表 6.**

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	
读输出允许时间		tOH8	5	—	50	

VDD=3.3V, Ta=25°C

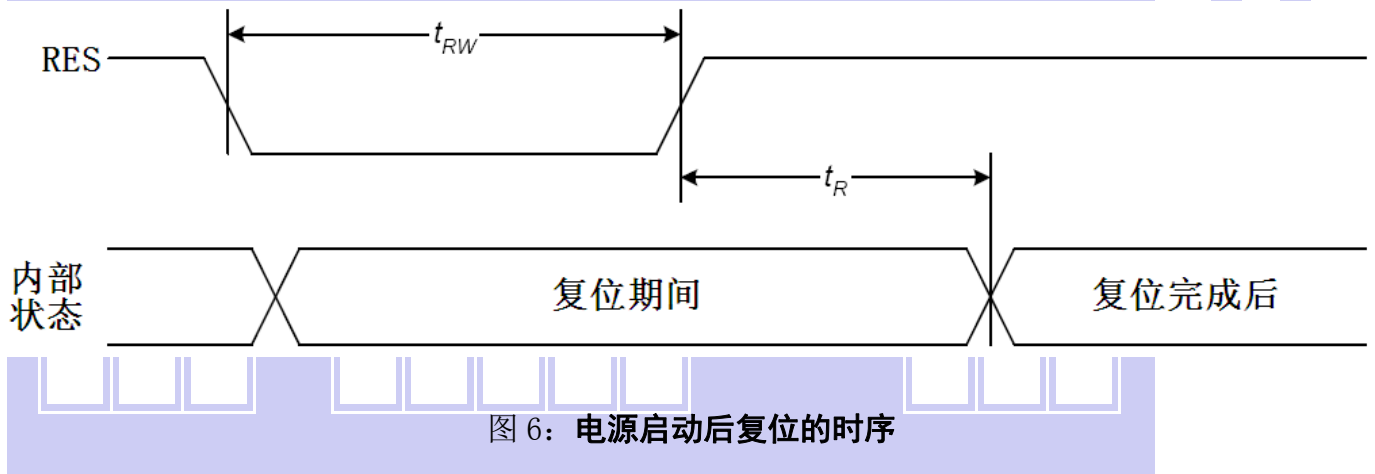
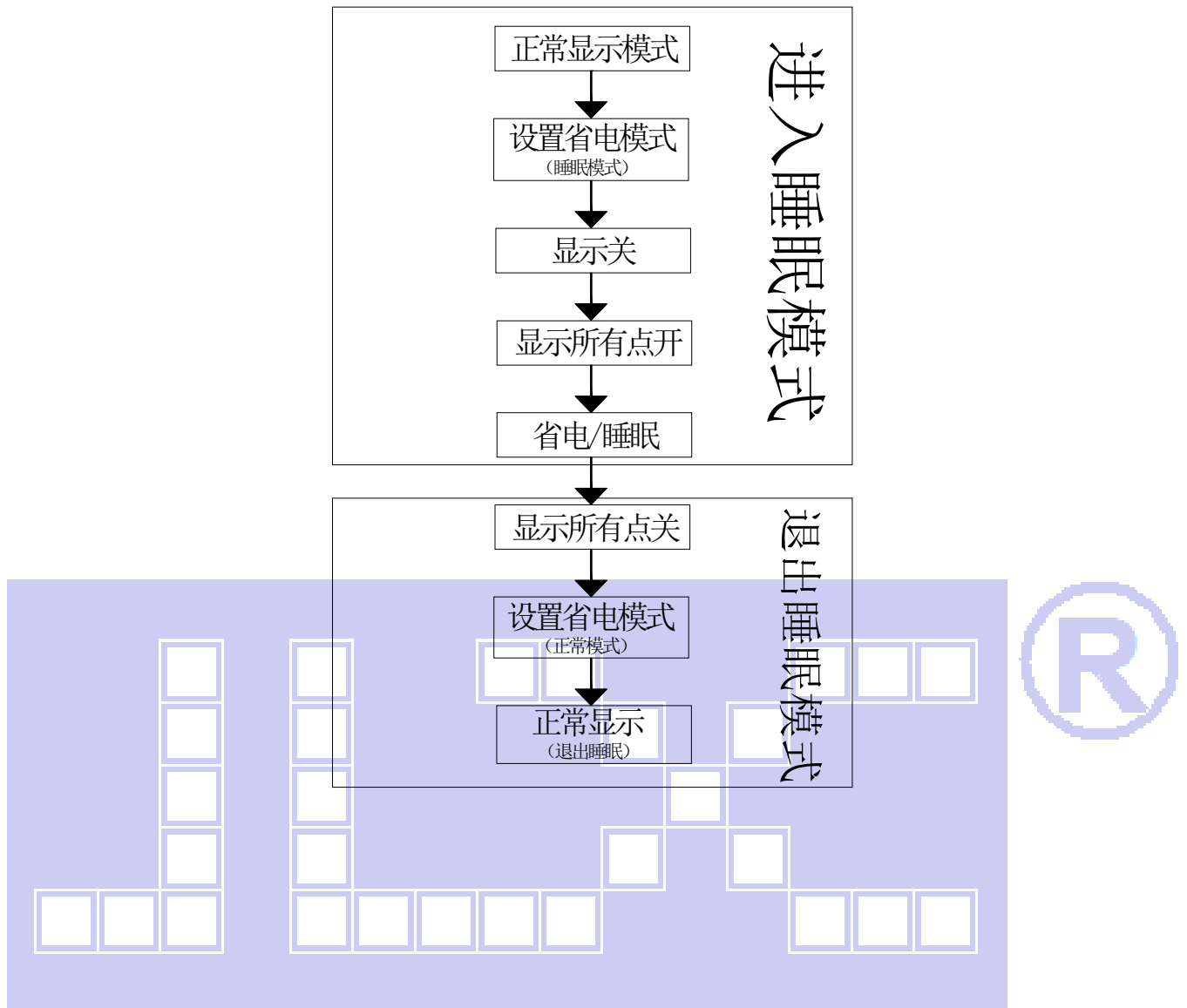
**6.5 电源启动后复位的时序要求 (RESET CONDITION AFTER POWER UP):**


图 6: 电源启动后复位的时序

**表 7: 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 <sub>R</sub>		—	—	1.0	us
复位保持低电平的时间	t <sub>RW</sub>	引脚: RES	1.0	—	—	us

### 6.6 省电模式设置



## 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: <b>0XAE</b> : 关, <b>0XAF</b> : 开	
(2) 显示初始行设置 (Display start line set)	0	0	1	<b>显示初始行地址, 共 6 位</b>							设置显示存储器的显示初始行, 可设置值为 <b>0X40~0X7F</b> , 分别代表第 <b>0~63</b> 行, 针对该液晶屏一般设置为 <b>0x40</b>
(3) 页地址设置 (Page address set)	0	1	0	1	1	<b>显示页地址, 共 4 位</b>				设置页地址。每 8 行为一个页, 64 行分为 8 个页, 可设置值为: <b>0XB0~0XB8</b> 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 <b>0XB0~0XB7</b> 分别对应第一页~第八页。	
(4) 列地址高4位设置 列地址低4位设置	0	0	0	0	1	<b>列地址的高 4 位</b>				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 <b>0x64</b> , 那么此指令由 2 个字节来表达: <b>0x16, 0x04</b>	
		0	0	0	0	<b>列地址的低 4 位</b>					
(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	<b>8 位显示数据</b>									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	<b>8 位显示数据</b>									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: <b>0xA0</b> : 常规: 列地址从左到右, <b>0xA1</b> : 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: <b>0xA6</b> : 常规: 正显 <b>0xA7</b> : 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: <b>0xA4</b> : 常规 <b>0xA5</b> : 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: <b>0XA2</b> : BIAS=1/9 (常用) <b>0XA3</b> : BIAS=1/7	
(12) 读-改-写 (Read-modify-write)	0	1	1	1	0	0	0	0	0	<b>0XE0</b> : “读-改-写” 开始。 <b>本液晶模块使用串行接口, 不具备此功能。</b> <a href="#">详情请参考 IC 资料</a>	
(13) 退出上述“读-改-写”指令 (End)	0	1	1	1	0	1	1	1	0	<b>0XEE</b> : 上述“读-改-写”指令结束 <b>本液晶模块使用串行接口, 不具备此功能。</b> <a href="#">详情请参考 IC 资料</a>	
(14) 软件复位 (Reset)	0	1	1	1	0	0	0	1	0	<b>0XE2</b> : 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: <b>0XC0</b> :普通扫描顺序: 从上到下 <b>0XC8</b> :反转扫描顺序: 从下到上
(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: 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 <b>0x2C,0x2E,0x2F</b> 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 <b>0x2F</b> , 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	<b>内部电压值电阻设置</b>			选择内部电阻比例 (Rb/Ra): 可以理解为 <b>粗调</b> 对比度值。可设置范围为: <b>0x20~0x27</b> , 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	1	设置内部电阻微调, 可以理解为 <b>微调</b> 对比度值, 此两个指令需紧接着使用。上面一条指令 <b>0x81</b> 是不改的, 下面一条指令可设置范围为: <b>0x00~0x3F</b> , 数值越大对比度越浓, 越小越淡
	设置的电压值	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	静态图标的开关设置: <b>0xAC</b> : 关, <b>0xAD</b> : 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
(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!

**温馨提示: 请详细参考 IC 资料”ST7567\_V1.7.PDF”的第 21~24 页。**

### 7.3 点阵与 DD RAM(显示数据存储)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”。

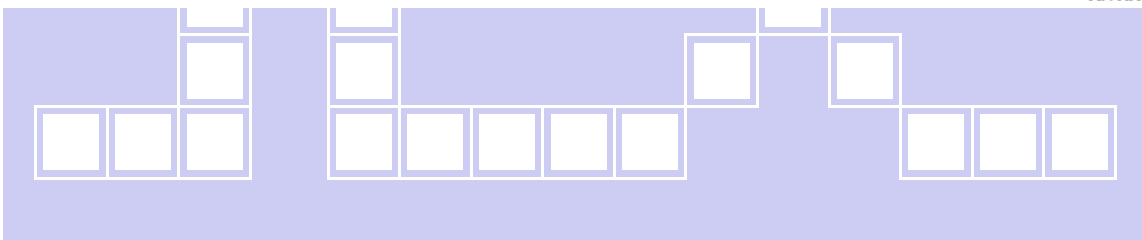
DB7--DB0 的排列方向: 数据是从下向上排列的。最低位 D0 是在最上面, 最高位 D7 是在最下面。每一位 (bit) 数据对应一个点阵, 通常“1”代表点亮该点阵, “0”代表关掉该点阵. 如下图所示:

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
-						

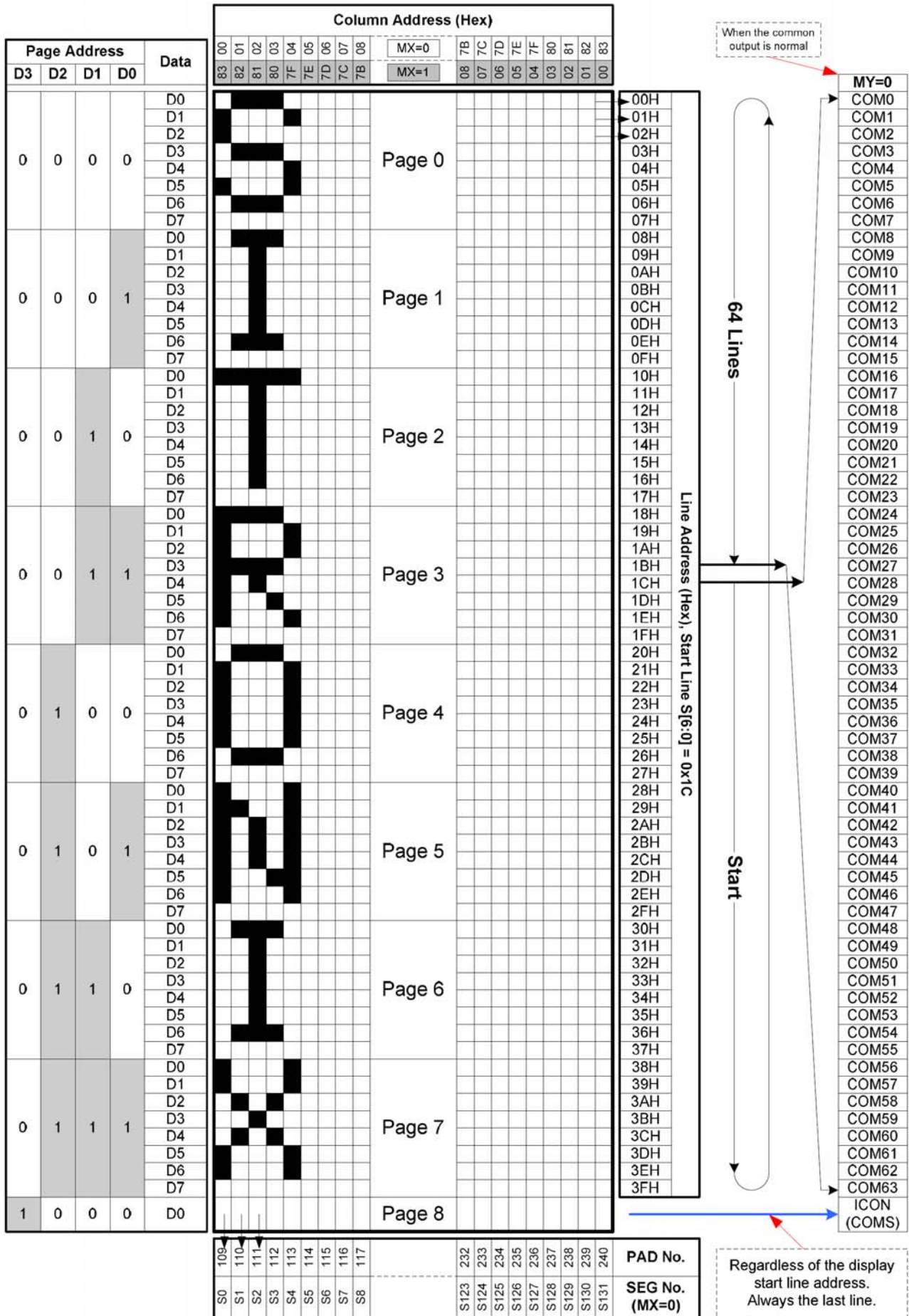
COM0		█				
COM1	█					
COM2						
COM3		█				
COM4	█					
-						

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

### 点亮液晶模块的步骤

**硬件准备:**  
开发板 (或专门设计的主板)、单片机、电源、连接线、仿真器或程序下载器 (又名烧录器)

**正确地接线**  
根据说明书正确地与开发板连接, 连接的线包括: 液晶模块电源线、背光电源线、IO端口 (接口)  
IO端口包括: 并口时: CS、RESET、RW、E、RS、D0—D7, 串口时: CS、SCLK、SDA、RESET、RS

**编写软件**  
背光给合适的直流电可以点亮, 但液晶屏里面没有程序, 只给电不能让液晶屏显示 (我们通常说“点亮”), 程序须另外编写, 并烧录 (下载) 到单片机里液晶模块才能工作。

### 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:

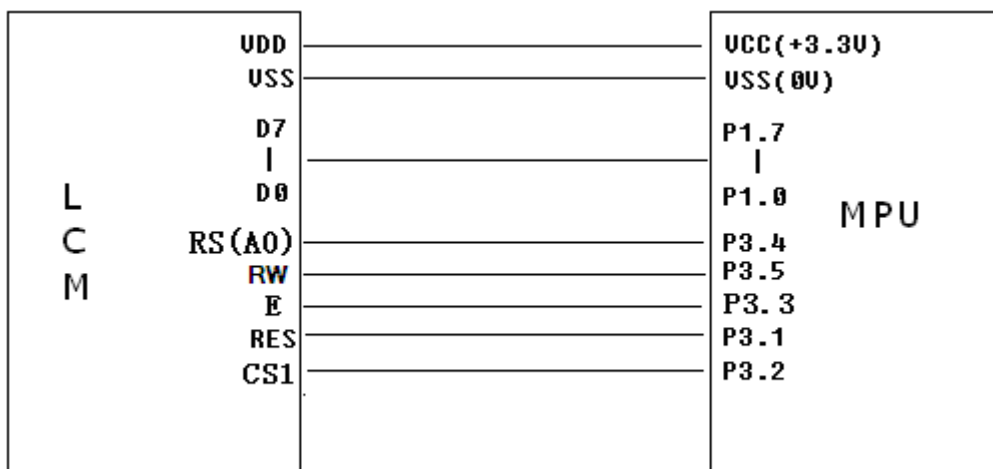


图 7. 并行接口

```

/* Test program for JLX12864G-66011-PN-P, 并行接口
   驱动 IC 是:ST7567(or competible)
   晶联讯电子: 网址 http://www.jlxlcd.cn; http://www.jlxlcd.com.cn
*/
#include <reg51.H>

sbit rs=P3^4;    /*接口定义:lcd_rs 就是 LCD 的 rs*/
sbit rd=P3^3;    /*接口定义:lcd_e 就是 LCD 的 rd*/
sbit wr=P3^5;    /*接口定义:lcd_rw 就是 LCD 的 wr*/
sbit reset=P3^1; /*接口定义:lcd_reset 就是 LCD 的 reset*/
sbit cs1=P3^2;   /*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)
{

    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(0x25); /*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(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*/
}

```

```

//=====clear all dot matrices=====
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<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:\work\图片收藏夹\12864G. bmp --*/
/*-- 宽度 x 高度=128x64 --*/
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xBF, 0xBE, 0x31, 0xFB, 0xFF, 0x7F, 0xBF, 0x41,
0x7D, 0x7D, 0x7D, 0x41, 0x3F, 0xBF, 0xBF, 0xFF, 0xFF, 0xDF, 0xD0, 0x56, 0x56, 0x50, 0x16, 0x46,
0x56, 0x50, 0x56, 0x56, 0xD0, 0xDF, 0xFF, 0xFF, 0xFF, 0xDB, 0xBB, 0x7B, 0x7B, 0x83, 0xFF, 0x6F,
0xEF, 0xEF, 0xEF, 0x00, 0xEF, 0xEF, 0xEF, 0xFF, 0xFF, 0xFF, 0x00, 0xDF, 0xDF, 0xDF, 0xDF, 0xFF,
}

```



液晶模块与 MPU(以 8051 系列单片机为例) 串行接口图如下:

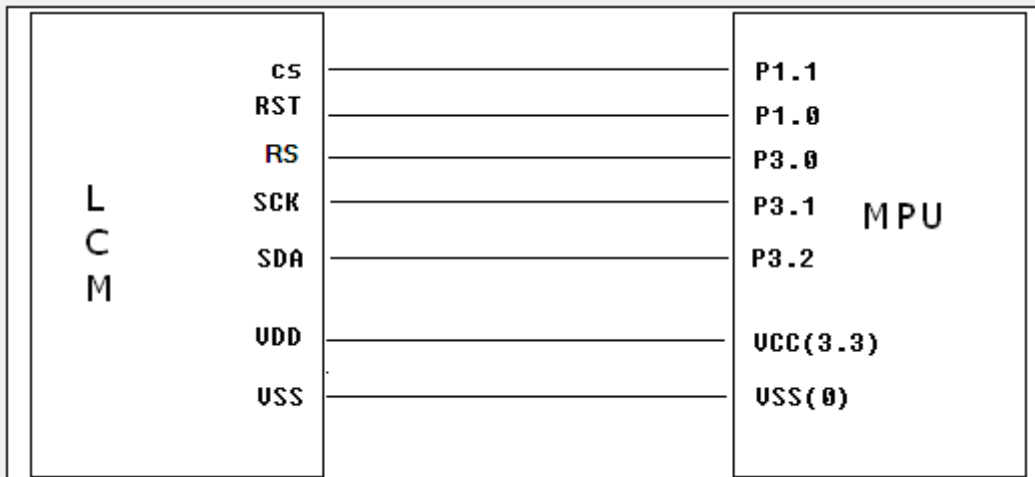


图 8. 串行接口

与并行方式相比较，只需改变传送数据和命令子程序即可

```

//-----
sbit cs1=P1^1; //3.4 接口定义
sbit reset=P1^0; //3.3 接口定义
sbit rs=P3^0; //接口定义
sbit sclk=P3^1; //接口定义
sbit sid=P3^2; //接口定义。
sbit key=P2^0; //按键接口，P2.0 口与 GND 之间接一个按键
//-----

/*-----写指令-----*/
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(2);
        sclk=1;
        Delay1(2);
        data1=data1<<=1;
    }
}

/*-----写数据-----*/
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;
    }
}

```

