

# JLX12864G-100 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	4~6
5	技术参数	7
6	时序特性	7~10
7	指令功能及硬件接口与编程案例	10~37

## 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-100 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-100 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-100 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，省去焊接 FPC 带来的麻烦

2.2 IC 采用矽创公司 ST7565R, 功能强大，稳定性好

2.3 功耗低:10 - 100mW (不带背光 10mW, 带背光不大于 100mW) ;

2.4 显示内容:

- 128\*64 点阵单色图片;

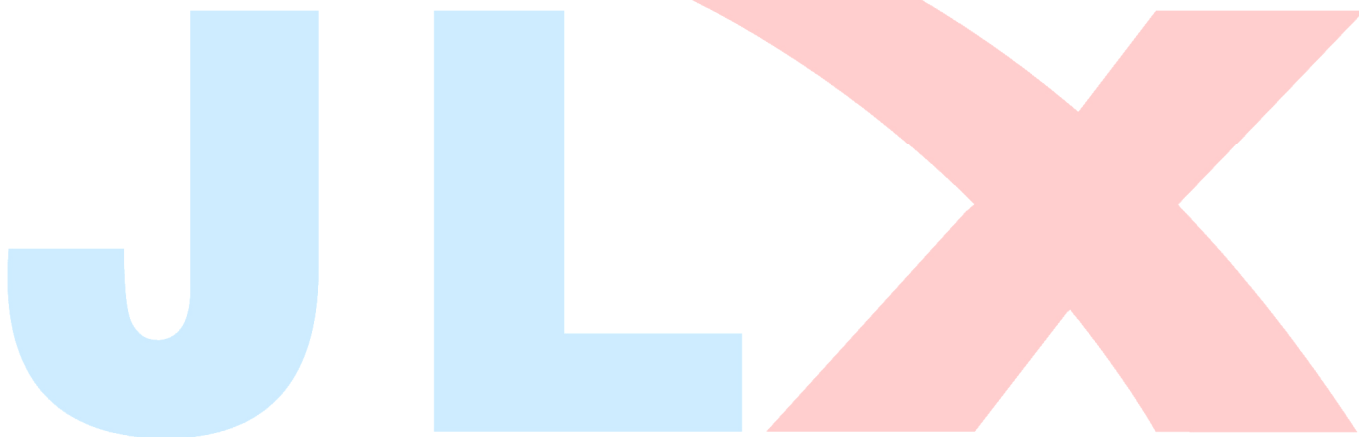
- 可選用 16\*16 点阵或其他点阵的图片来自编汉字，按照 16\*16 点阵汉字来计算可显示 8 字/行\*4 行。按照 12\*12 点阵汉字来计算可显示 10 字/行\*4 行。

2.5 指令功能强;

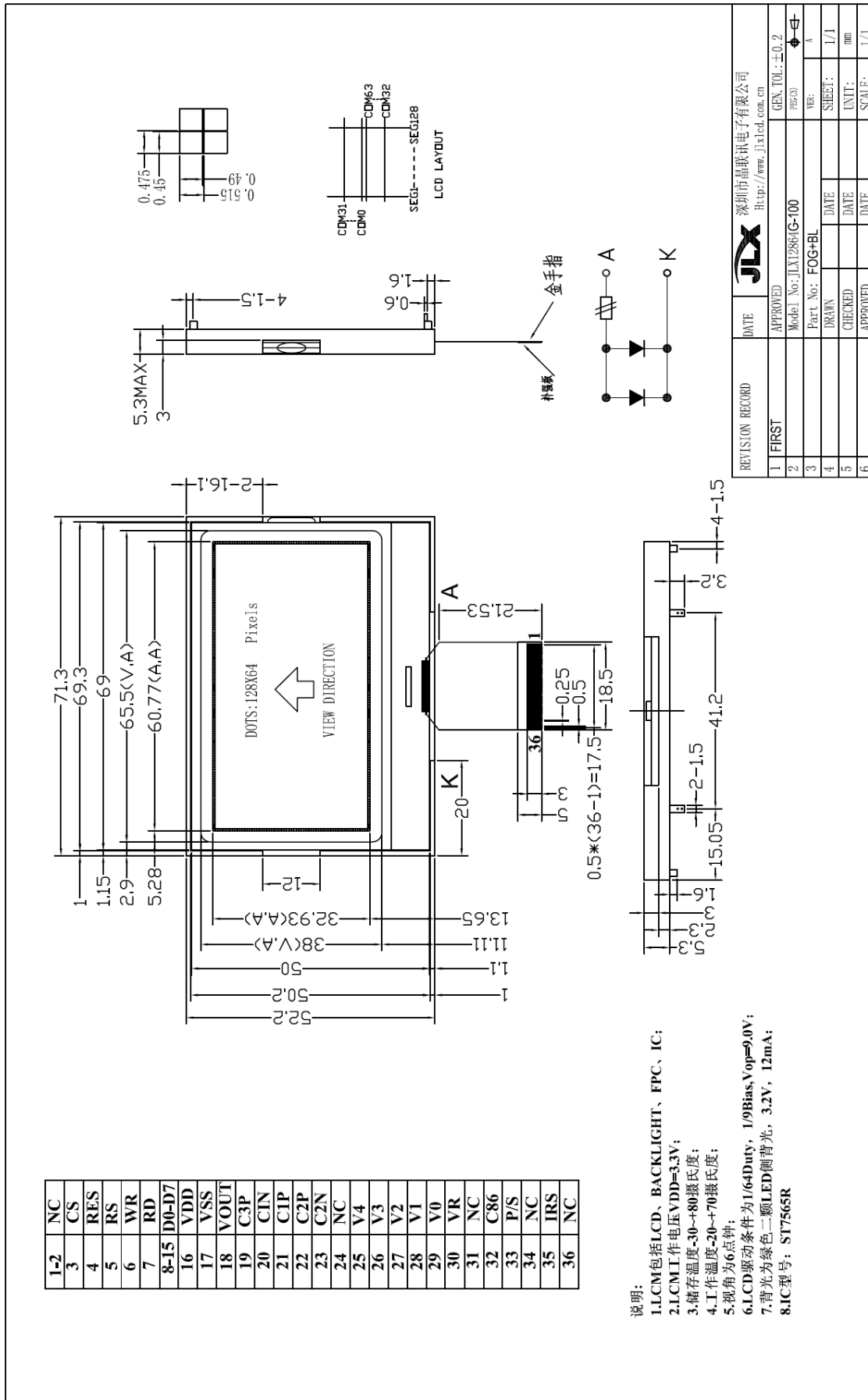
2.6 接口简单方便:可采用 4 线 SPI 串行接口，或选择并行接口。

2.7 工作温度宽:-20℃ - 70℃;

2.8 可靠性高:寿命为 50,000 小时(25℃)。



3. 外形尺寸及接口引脚功能



REVISION RECORD		DATE		APPROVED	
1	FIRST				
2					
3					
4					
5					
6					

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	NC	空脚	空脚
2	NC	空脚	空脚
3	CS	片选	低电平片选
4	RES(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
5	RS	寄存器选择信号	H:数据寄存器 0:指令寄存器
6	R/W(/WR)	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
7	E(/RD)	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
8-13	D0-D5	I/O	数据总线 DB0~DB5 串行接口时: 空脚
14	D6(SCLK)	I/O	并行接口时: 数据总线 DB6 串行接口时: 串行时钟(SCLK)
15	D7(SDA)	I/O	并行接口时: 数据总线 DB7 串行接口时: 串行数据(SDA)
16	VDD	供电电源正极	供电电源正极
17	VSS	接地	0V
18	VOUT	LCD 倍压输出	外接升压电容: 方法见“7. 指令功能及硬件接口与编程案例”
19	C3P	倍压电路	
20	C1N	倍压电路	
21	C1P	倍压电路	
22	C2P	倍压电路	
23	C2N	倍压电路	
24	NC	空脚	空脚
25	V4	偏置电压	LCD 驱动偏置电压。各与 VSS 之间接电容。方法见“7. 指令功能及硬件接口与编程案例”。 电压关系: VOUT>V0>V1>V2>V3>V4>VSS。
26	V3	偏置电压	
27	V2	偏置电压	
28	V1	偏置电压	
29	V0	偏置电压	
30	VR	产品电压调整	电压调整, 通过 V0 与 VSS 之间设置电阻来调整电压。方法见“7. 指令功能及硬件接口与编程案例”。
31	NC	空脚	空脚
32	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
33	P/S	选串并控制接口	H: 并行接口, L: 串行接口
34	NC		空脚
35	IRS	内/外电阻	L: 使用外部电阻, 此时“VR”引脚起作用。 H: 使用 IC 内部电阻进行调对比度。此时“VR”引脚不起作用
36	NC		空脚

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 邦定在 LCD 玻璃上 (这种加工工艺叫 COG)。

##### 4.2 工作电图:

图 2 是 JLX12864G-100 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。

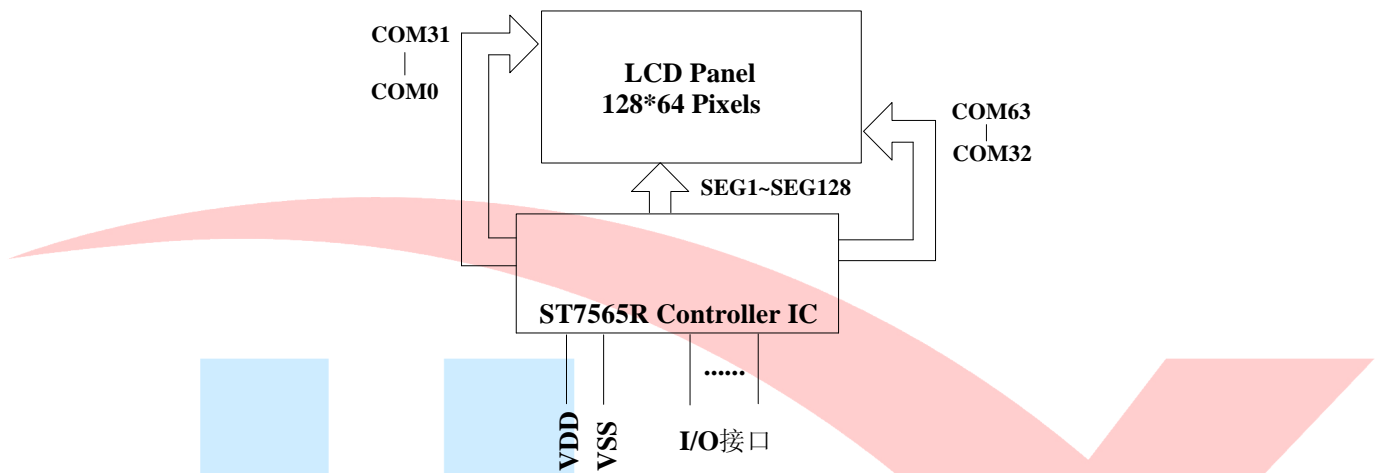


图 2: JLX12864G-100 图像点阵型液晶模块的电路框图

##### 4.2 升压电路图:

详见“7. 指令功能及硬件接口与编程案例”

##### 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~+70° C;

存储温度: -30~+80° C;

背光板可选择绿色、白色。

正常工作电流为: 16~40mA (LED 灯数共 2 颗);

工作电压: 3.0V;

正常工作条件下, LED 可连续点亮 5 万小时;

## 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	°C
储存温度		-30		+80	°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		2.4	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	-	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	I00 = 1.2mA	-		0.4	V
模块工作电流	IDD	VDD = 3.0V	-		1.0	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

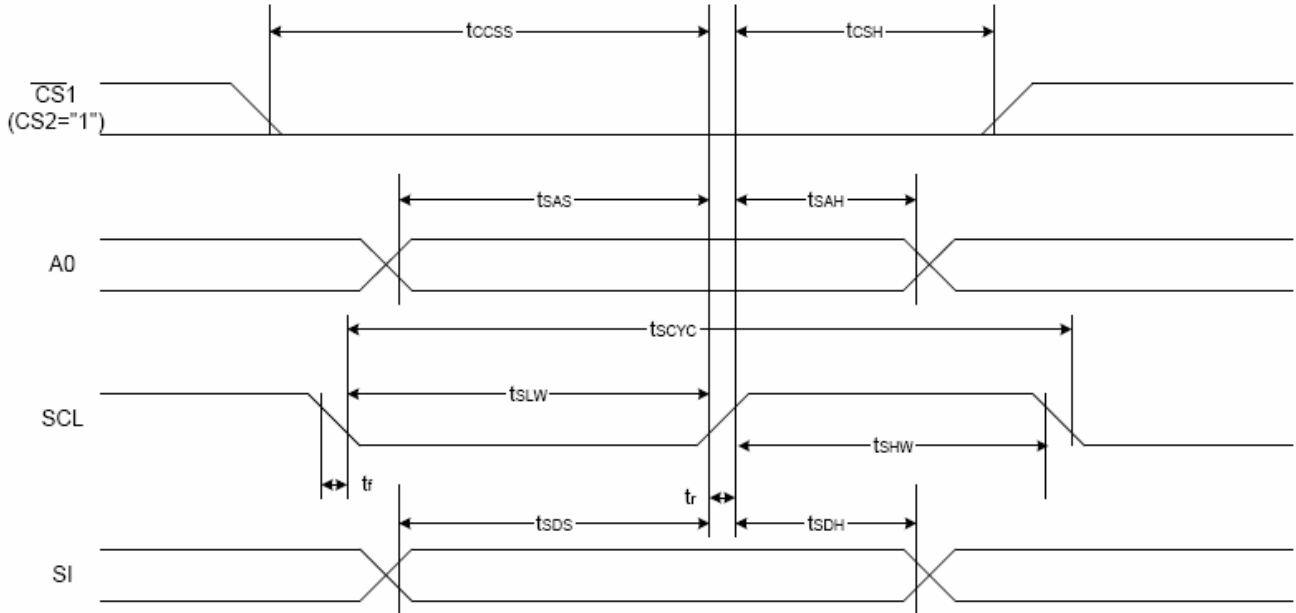


图 4. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

### 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	50	--	25	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)

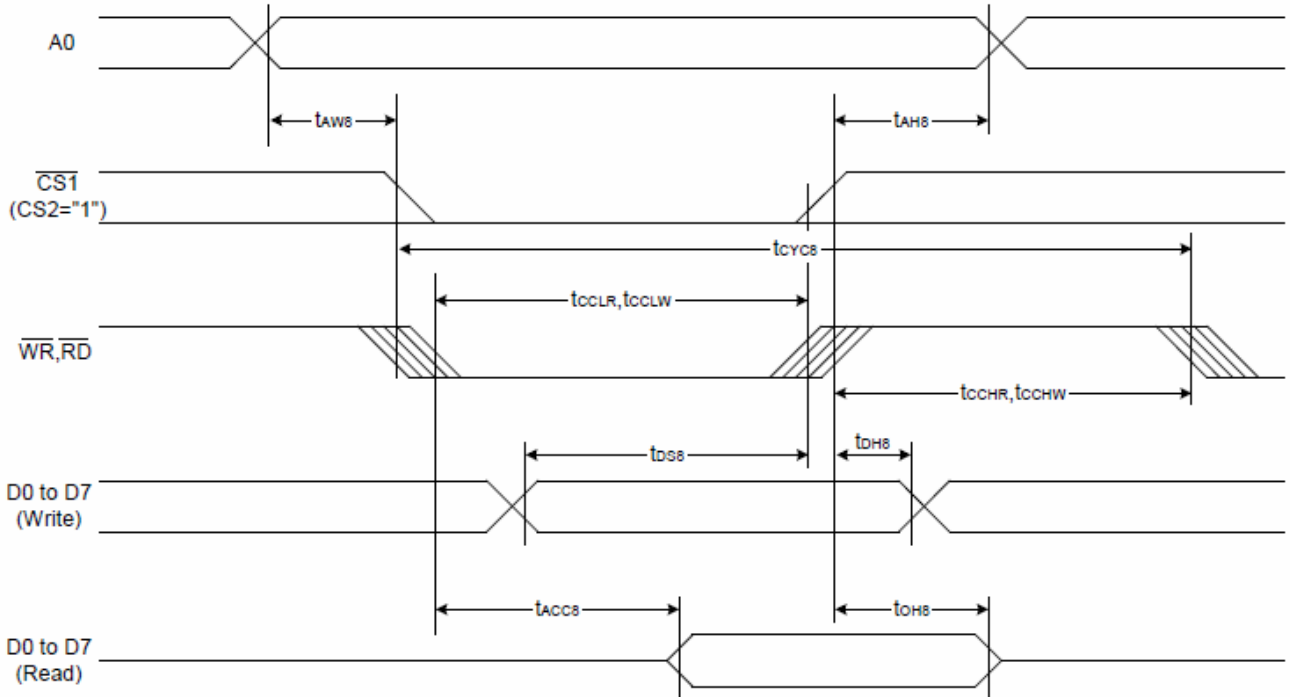


图 5. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)



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

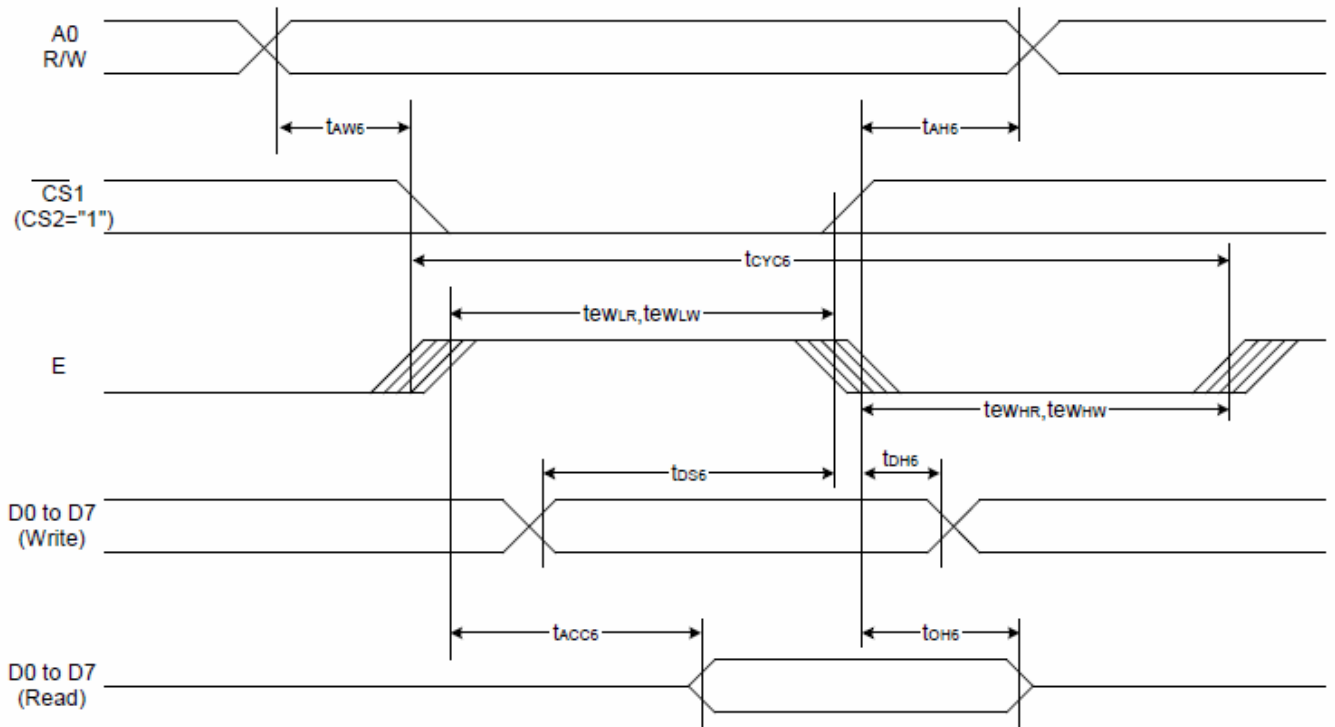


图 6. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

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）:**

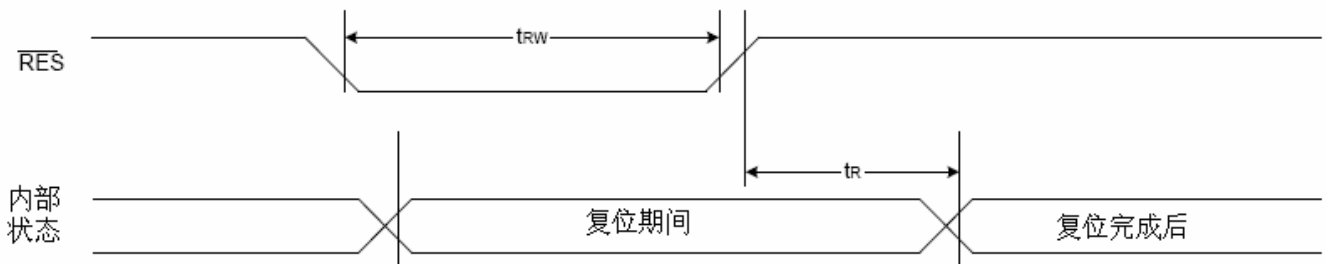


图 7：电源启动后复位的时序

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	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: 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 00000001, 所以此指令表达为: 0x10, 0x01	
											列地址低4位 设置
(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。	

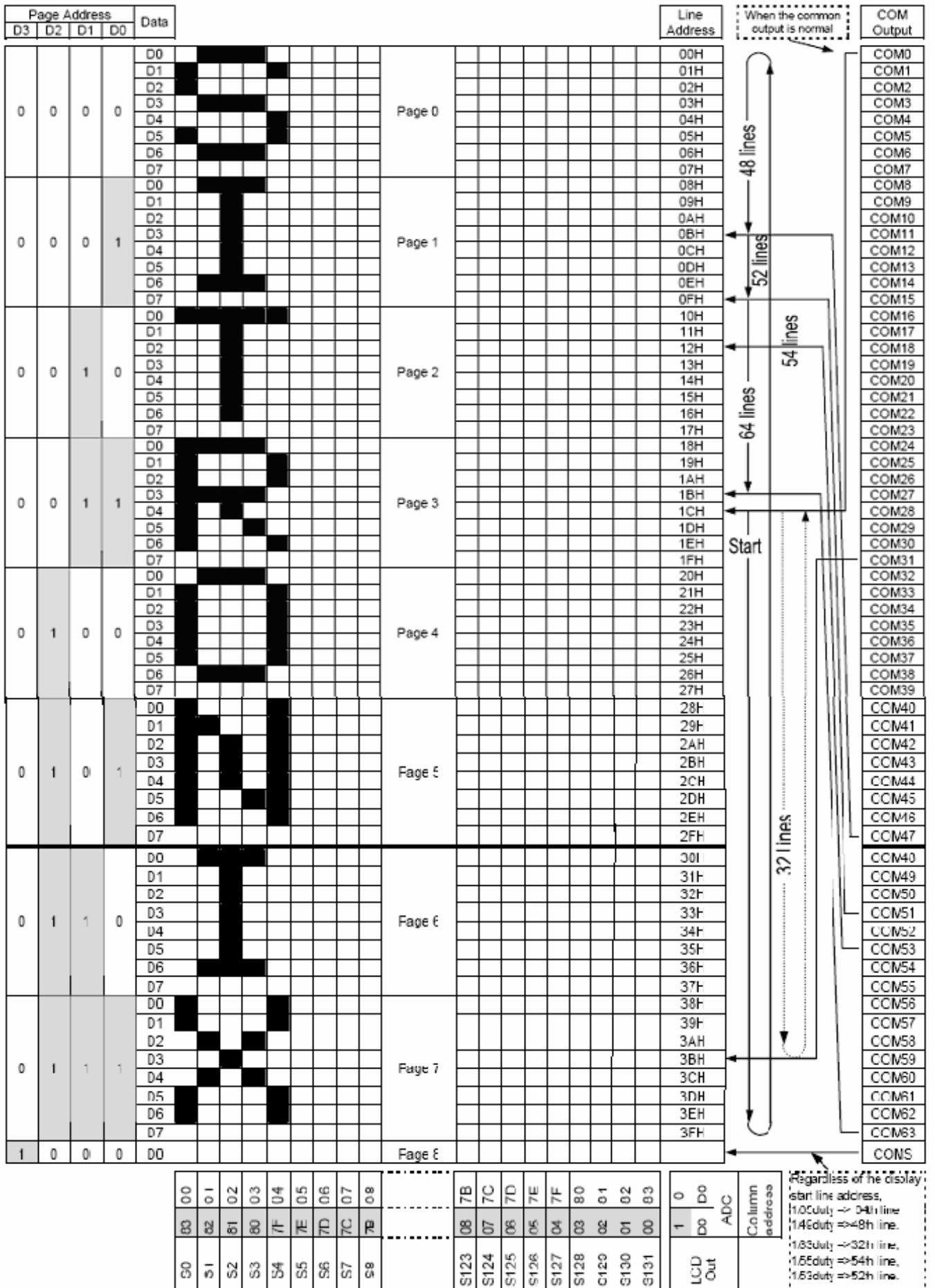
(15) 行扫描顺序选择(Common output mode select)		1	1	0	0	0	0	0	0	行扫描顺序选择: 0: 普通顺序 1: 反向扫描
(16) 电源控制 (Power control set)		0	0	1	0	1	<b>电压操作模式选择, 共 3 位</b>			选择内部电压供应操作模式
(17) 选择内部电阻比例	0	0	0	1	0	0	<b>内部电压值电阻设置</b>			选择内部电阻比例 (Rb/Ra), 本液晶模块通过外置电阻设置, 此指令失效
(18) 内部设置液晶电压模式	0	1	0	0	0	0	0	0	1	设置内部电阻微调, 以设置液晶电压, 此两个指令需紧接着使用
	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	<b>01</b>	0: 关, 1: 开。本液晶屏无此图标。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	<b>2 位数设置升压倍数</b>		
(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!

请详细参考 IC 资料“ST7564R\_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:

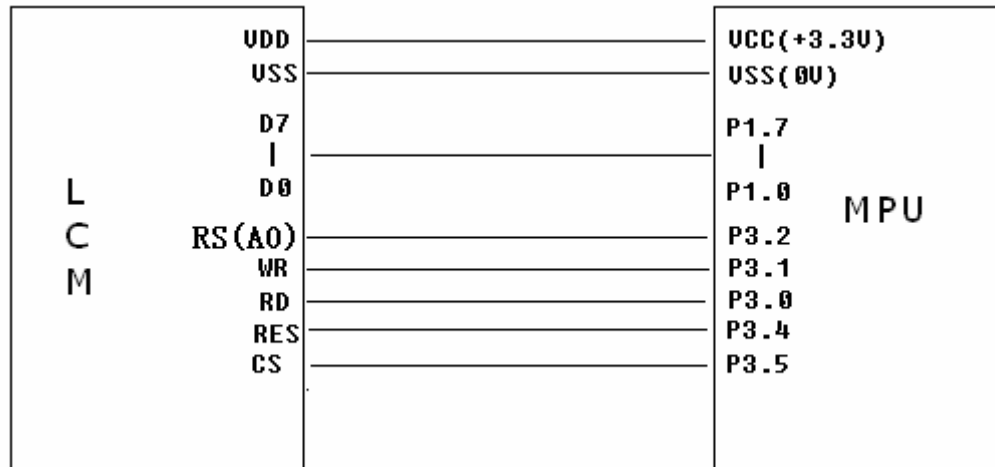
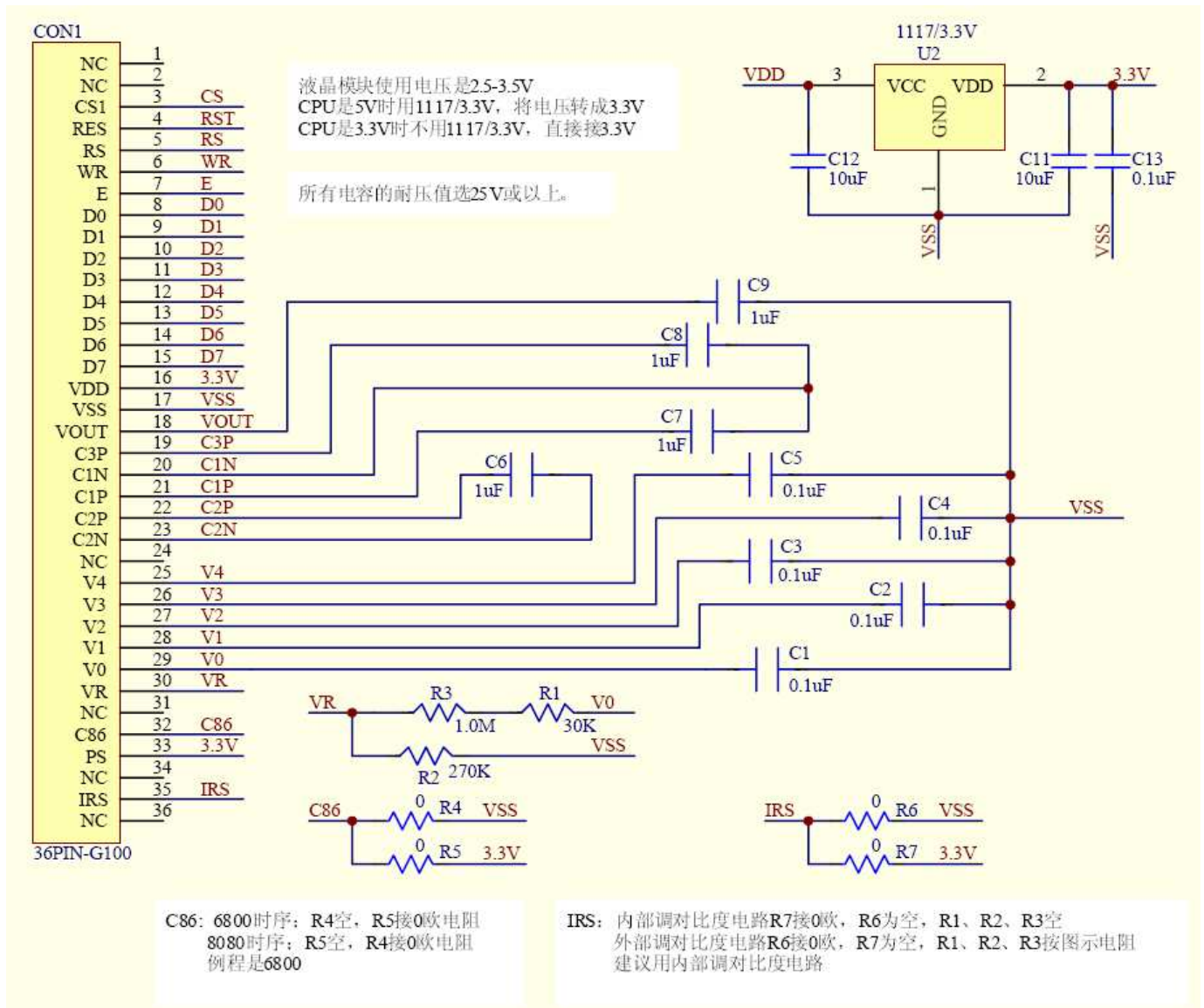


图 8. 并行接口

并行调对比度电路图 (建议用内调对比度电路 IRS 接 3.3V)



C86: 6800时序: R4空, R5接0欧电阻  
8080时序: R5空, R4接0欧电阻  
例程是6800

IRS: 内部调对比度电路R7接0欧, R6为空, R1、R2、R3空  
外部调对比度电路R6接0欧, R7为空, R1、R2、R3按图示电阻  
建议用内部调对比度电路

### 7.51、以下为并行方式的例程序

```
/* Test program for JLX12864G-100, 并行接口
   驱动 IC 是:ST7565R(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=P3^3; /*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*/
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 ascii_table_8x16[95][16];
uchar code ascii_table_5x8[95][5];
uchar code cheng1[];
uchar code gong1[];
uchar code zhuang1[];
uchar code tai1[];
uchar code shi1[];
uchar code yong1[];
uchar code bmp_12864_1[];
uchar code bmp_12864_2[];
uchar code bmp_12864_3[];
uchar code bmp_12864_4[];
uchar code bmp_12864_5[];

//=====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;
}
```

```
/*延时*/
```

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

```
{
    int j,k;
    for(j=0;j<i;j++)
    for(k=0;k<110;k++);
}
```

```
void waitkey()
```

```
{
repeat:  if(key==1)
            goto repeat;
            else
            delay(600);
}
```

```
/*LCD 模块初始化*/
```

```
void initial_lcd()
```

```
{
    cs1=0;
    reset=0;      /*低电平复位*/
    delay(20);
    reset=1;      /*复位完毕*/
    delay(20);
    transfer_command(0xe2); /*软复位*/
    delay(50);
    transfer_command(0x2c); /*升压步聚 1*/
    delay(50);
    transfer_command(0x2e); /*升压步聚 2*/
    delay(50);
    transfer_command(0x2f); /*升压步聚 3*/
    delay(50);
}
```

```

transfer_command(0x24); /*粗调对比度，可设置范围 0x20~0x27 25*/
transfer_command(0x81); /*微调对比度*/
transfer_command(0x28); /*微调对比度的值，可设置范围 0x00~0x3f 18*/
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*/
cs1=1;
}

void lcd_address(uchar page,uchar column)
{
    cs1=0;
    column=column; //
    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<132;j++)
        {
            transfer_data(0x00);
        }
    }
    cs1=1;
}

//===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, 0);
    }
}

```

```

        for(i=0;i<128;i++)
        {
            transfer_data(data1);
            transfer_data(data2);
        }
    }
}

```

/\*显示 128x64 点阵图像\*/

```

void display_graphic_128x64(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<128;i++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}

```

/\*显示 32x32 点阵图像、汉字、生僻字或 32x32 点阵的其他图标\*/

```

void display_graphic_32x32(uchar page,uchar column,uchar *dp)
{
    uchar i,j;
    cs1=0;
    for(j=0;j<4;j++)
    {
        lcd_address(page+j,column);
        for (i=0;i<31;i++)
        {
            transfer_data(*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++;
        }
    }
}

```

/\*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_2(uchar reverse,uchar page,uchar column,uchar *dp)
{
    uchar 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);          /*写数据到 LCD, 每写完一个 8 位的数据后列地址自动加 1*/
        else
            transfer_data(~*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++;
    }
}
cs1=1;
}
/*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(uchar page, uchar column, uchar *dp)
{
    uchar i, j;
    cs1=0;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j, column);
        for (i=0;i<16;i++)
        {
            transfer_data(*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++;
        }
    }
    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(uchar page, uchar column, uchar *dp)
{
    uchar i, j;
    cs1=0;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j, column);
        for (i=0;i<8;i++)
        {
            transfer_data(*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++;
        }
    }
    cs1=1;
}
}
```

```
void display_string_8x16(uint page,uint column,uchar *text)
{
    uint i=0,j,k,n;
    csl=0;
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<=0x7e))
        {
            j=text[i]-0x20;
            for(n=0;n<2;n++)
            {
                lcd_address(page+n, column);
                for(k=0;k<8;k++)
                {
                    transfer_data(ascii_table_8x16[j][k+8*n]);/*显示 5x7 的 ASCII 字到 LCD 上, y 为页地址, x 为列地址, 最后为数据*/
                }
            }
            i++;
            column+=8;
        }
        else
            i++;
    }
}
```

```
void display_string_5x8(uint page,uint column,uchar *text)
{
    uint i=0,j,k;
    csl=0;
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<0x7e))
        {
            j=text[i]-0x20;
            lcd_address(page, column);
            for(k=0;k<5;k++)
            {
                transfer_data(ascii_table_5x8[j][k]);/*显示 5x7 的 ASCII 字到 LCD 上, y 为页地址, x 为列地址, 最后为数据*/
            }
            i++;
            column+=6;
        }
        else
            i++;
    }
}
```

```
}
}

void main(void)
{
    while(1)
    {
        initial_lcd();
        clear_screen(); //clear all dots
        display_graphic_128x64(1, 1, bmp_12864_1); //显示 12864 点阵图片
        waitkey();
        clear_screen(); //clear all dots
        display_graphic_128x64(1, 1, bmp_12864_2); //显示 12864 点阵图片
        waitkey();
        clear_screen(); //clear all dots
        display_graphic_128x64(1, 1, bmp_12864_3); //显示 12864 点阵图片
        waitkey();
        clear_screen(); //clear all dots
        display_graphic_128x64(1, 1, bmp_12864_4); //显示 12864 点阵图片
        waitkey();
        clear_screen(); //clear all dots
        display_graphic_128x64(1, 1, bmp_12864_5); //显示 12864 点阵图片
        waitkey();
        clear_screen(); //clear all dots
        display_graphic_32x32(1, (1+32*1), cheng1); /*在第 1 页, 第 49 列显示单个汉字"成"*/
        display_graphic_32x32(1, (1+32*2), gong1); /*在第 1 页, 第 49 列显示单个汉字"功"*/
        display_graphic_16x16_2(0, 6, 1, zhuang1); /*在第 5 页, 第 1 列显示单个汉字"状"*/
        display_graphic_16x16_2(0, 6, (1+16), tail); /*在第 5 页, 第 17 列显示单个汉字"态"*/
        display_string_8x16(6, (1+16+16), ":"); /*在第 5 页, 第 25 列显示单个字符":"*/
        display_graphic_16x16(6, (1+16*2+8), shi1); /*在第 5 页, 第 41 列显示单个汉字"使"*/
        display_graphic_16x16(6, (1+16*3+8), yong1); /*在第 5 页, 第 49 列显示单个汉字"用"*/
        display_string_8x16(6, 89, "12:45"); /*在第 5 页, 第 89 列显示单个数字"0"*/
        waitkey();
        clear_screen(); //clear all dots
        display_string_8x16(1, 1, "0123456789abcdef");/*在第 1 页, 第 1 列显示字符串*/
        display_string_8x16(3, 1, "~`!@#%&*( )_-=");/*在第*页, 第*列显示字符串*/
        display_string_5x8(5, 1, " ! #$$%&'()*+,-./01234");
        display_string_5x8(6, 1, "56789:;<=>?@ABCDEFGHI");
        display_string_5x8(7, 1, "JKLMNOPQRSTUVWXYZ[\\]^");
        display_string_5x8(8, 1, "_`abcdefghijklmnopqrs");
        waitkey();

        //delay(2000);
    }
}
```

```
uchar code ascii_table_8x16[95][16]={

/*-- 文字:  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16  --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

/*-- 文字:  !  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16  --*/
0x00, 0x00, 0x00, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x33, 0x30, 0x00, 0x00, 0x00,

/*-- 文字:  "  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16  --*/
0x00, 0x10, 0x0C, 0x06, 0x10, 0x0C, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

/*-- 文字:  #  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16  --*/
0x40, 0xC0, 0x78, 0x40, 0xC0, 0x78, 0x40, 0x00, 0x04, 0x3F, 0x04, 0x04, 0x3F, 0x04, 0x04, 0x00,

/*-- 文字:  $  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16  --*/
0x00, 0x70, 0x88, 0xFC, 0x08, 0x30, 0x00, 0x00, 0x00, 0x18, 0x20, 0xFF, 0x21, 0x1E, 0x00, 0x00,

/*-- 文字:  %  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16  --*/
0xF0, 0x08, 0xF0, 0x00, 0xE0, 0x18, 0x00, 0x00, 0x00, 0x21, 0x1C, 0x03, 0x1E, 0x21, 0x1E, 0x00,

/*-- 文字:  &  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16  --*/
0x00, 0xF0, 0x08, 0x88, 0x70, 0x00, 0x00, 0x00, 0x1E, 0x21, 0x23, 0x24, 0x19, 0x27, 0x21, 0x10,

/*-- 文字:  '  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16  --*/
0x10, 0x16, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

/*-- 文字:  (  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16  --*/
0x00, 0x00, 0x00, 0xE0, 0x18, 0x04, 0x02, 0x00, 0x00, 0x00, 0x00, 0x07, 0x18, 0x20, 0x40, 0x00,

/*-- 文字:  )  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16  --*/
0x00, 0x02, 0x04, 0x18, 0xE0, 0x00, 0x00, 0x00, 0x40, 0x20, 0x18, 0x07, 0x00, 0x00, 0x00,

/*-- 文字:  *  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16  --*/
0x40, 0x40, 0x80, 0xF0, 0x80, 0x40, 0x40, 0x00, 0x02, 0x02, 0x01, 0x0F, 0x01, 0x02, 0x02, 0x00,
```

/\*-- 文字： + --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x01, 0x1F, 0x01, 0x01, 0x01, 0x00,

/\*-- 文字： , --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0xB0, 0x70, 0x00, 0x00, 0x00, 0x00,

/\*-- 文字： - --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01,

/\*-- 文字： . --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00,

/\*-- 文字： / --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0x00, 0x80, 0x60, 0x18, 0x04, 0x00, 0x60, 0x18, 0x06, 0x01, 0x00, 0x00, 0x00,

/\*-- 文字： 0 --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x0F, 0x10, 0x20, 0x20, 0x10, 0x0F, 0x00,

/\*-- 文字： 1 --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x10, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,

/\*-- 文字： 2 --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x70, 0x08, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x30, 0x28, 0x24, 0x22, 0x21, 0x30, 0x00,

/\*-- 文字： 3 --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x30, 0x08, 0x88, 0x88, 0x48, 0x30, 0x00, 0x00, 0x18, 0x20, 0x20, 0x20, 0x11, 0x0E, 0x00,

/\*-- 文字： 4 --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x00, 0xC0, 0x20, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x07, 0x04, 0x24, 0x24, 0x3F, 0x24, 0x00,

/\*-- 文字： 5 --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0xF8, 0x08, 0x88, 0x88, 0x08, 0x08, 0x00, 0x00, 0x19, 0x21, 0x20, 0x20, 0x11, 0x0E, 0x00,

/\*-- 文字： 6 --\*/



/\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0xE0, 0x10, 0x88, 0x88, 0x18, 0x00, 0x00, 0x00, 0x0F, 0x11, 0x20, 0x20, 0x11, 0x0E, 0x00,

/\*-- 文字： 7 --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x38, 0x08, 0x08, 0xC8, 0x38, 0x08, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x00, 0x00, 0x00, 0x00,

/\*-- 文字： 8 --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x70, 0x88, 0x08, 0x08, 0x88, 0x70, 0x00, 0x00, 0x1C, 0x22, 0x21, 0x21, 0x22, 0x1C, 0x00,

/\*-- 文字： 9 --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0xE0, 0x10, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x00, 0x00, 0x31, 0x22, 0x22, 0x11, 0x0F, 0x00,

/\*-- 文字： : --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0xC0, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x30, 0x00, 0x00, 0x00,

/\*-- 文字： ; --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x60, 0x00, 0x00, 0x00, 0x00,

/\*-- 文字： < --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x80, 0x40, 0x20, 0x10, 0x08, 0x00, 0x00, 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x00,

/\*-- 文字： = --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x40, 0x40, 0x40, 0x40, 0x40, 0x40, 0x40, 0x00, 0x04, 0x04, 0x04, 0x04, 0x04, 0x04, 0x00,

/\*-- 文字： > --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x08, 0x10, 0x20, 0x40, 0x80, 0x00, 0x00, 0x00, 0x20, 0x10, 0x08, 0x04, 0x02, 0x01, 0x00,

/\*-- 文字： ? --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x70, 0x48, 0x08, 0x08, 0x08, 0xF0, 0x00, 0x00, 0x00, 0x00, 0x30, 0x36, 0x01, 0x00, 0x00,

/\*-- 文字： @ --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0xC0, 0x30, 0xC8, 0x28, 0xE8, 0x10, 0xE0, 0x00, 0x07, 0x18, 0x27, 0x24, 0x23, 0x14, 0x0B, 0x00,

/\*-- 文字： A --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16 --\*/  
 0x00, 0x00, 0xC0, 0x38, 0xE0, 0x00, 0x00, 0x00, 0x20, 0x3C, 0x23, 0x02, 0x02, 0x27, 0x38, 0x20,

/\*-- 文字: B --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/

0x08, 0xF8, 0x88, 0x88, 0x88, 0x70, 0x00, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x11, 0x0E, 0x00,

/\*-- 文字: C --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/

0xC0, 0x30, 0x08, 0x08, 0x08, 0x08, 0x38, 0x00, 0x07, 0x18, 0x20, 0x20, 0x20, 0x10, 0x08, 0x00,

/\*-- 文字: D --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/

0x08, 0xF8, 0x08, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x10, 0x0F, 0x00,

/\*-- 文字: E --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/

0x08, 0xF8, 0x88, 0x88, 0xE8, 0x08, 0x10, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x23, 0x20, 0x18, 0x00,

/\*-- 文字: F --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/

0x08, 0xF8, 0x88, 0x88, 0xE8, 0x08, 0x10, 0x00, 0x20, 0x3F, 0x20, 0x00, 0x03, 0x00, 0x00, 0x00,

/\*-- 文字: G --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/

0xC0, 0x30, 0x08, 0x08, 0x08, 0x38, 0x00, 0x00, 0x07, 0x18, 0x20, 0x20, 0x22, 0x1E, 0x02, 0x00,

/\*-- 文字: H --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/

0x08, 0xF8, 0x08, 0x00, 0x00, 0x08, 0xF8, 0x08, 0x20, 0x3F, 0x21, 0x01, 0x01, 0x21, 0x3F, 0x20,

/\*-- 文字: I --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/

0x00, 0x08, 0x08, 0xF8, 0x08, 0x08, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,

/\*-- 文字: J --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/

0x00, 0x00, 0x08, 0x08, 0xF8, 0x08, 0x08, 0x00, 0xC0, 0x80, 0x80, 0x80, 0x7F, 0x00, 0x00, 0x00,

/\*-- 文字: K --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/

0x08, 0xF8, 0x88, 0xC0, 0x28, 0x18, 0x08, 0x00, 0x20, 0x3F, 0x20, 0x01, 0x26, 0x38, 0x20, 0x00,

/\*-- 文字: L --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/

0x08, 0xF8, 0x08, 0x00, 0x00, 0x00, 0x00, 0x20, 0x3F, 0x20, 0x20, 0x20, 0x20, 0x30, 0x00,

/\*-- 文字: M --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x08, 0xF8, 0xF8, 0x00, 0xF8, 0xF8, 0x08, 0x00, 0x20, 0x3F, 0x00, 0x3F, 0x00, 0x3F, 0x20, 0x00,

/\*-- 文字: N --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x08, 0xF8, 0x30, 0xC0, 0x00, 0x08, 0xF8, 0x08, 0x20, 0x3F, 0x20, 0x00, 0x07, 0x18, 0x3F, 0x00,

/\*-- 文字: O --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0xE0, 0x10, 0x08, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x0F, 0x10, 0x20, 0x20, 0x20, 0x10, 0x0F, 0x00,

/\*-- 文字: P --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x08, 0xF8, 0x08, 0x08, 0x08, 0x08, 0xF0, 0x00, 0x20, 0x3F, 0x21, 0x01, 0x01, 0x01, 0x00, 0x00,

/\*-- 文字: Q --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0xE0, 0x10, 0x08, 0x08, 0x08, 0x10, 0xE0, 0x00, 0x0F, 0x18, 0x24, 0x24, 0x38, 0x50, 0x4F, 0x00,

/\*-- 文字: R --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x08, 0xF8, 0x88, 0x88, 0x88, 0x88, 0x70, 0x00, 0x20, 0x3F, 0x20, 0x00, 0x03, 0x0C, 0x30, 0x20,

/\*-- 文字: S --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x70, 0x88, 0x08, 0x08, 0x08, 0x38, 0x00, 0x00, 0x38, 0x20, 0x21, 0x21, 0x22, 0x1C, 0x00,

/\*-- 文字: T --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x18, 0x08, 0x08, 0xF8, 0x08, 0x08, 0x18, 0x00, 0x00, 0x00, 0x20, 0x3F, 0x20, 0x00, 0x00, 0x00,

/\*-- 文字: U --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x08, 0xF8, 0x08, 0x00, 0x00, 0x08, 0xF8, 0x08, 0x00, 0x1F, 0x20, 0x20, 0x20, 0x20, 0x1F, 0x00,

/\*-- 文字: V --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x08, 0x78, 0x88, 0x00, 0x00, 0xC8, 0x38, 0x08, 0x00, 0x00, 0x07, 0x38, 0x0E, 0x01, 0x00, 0x00,

/\*-- 文字: W --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0xF8, 0x08, 0x00, 0xF8, 0x00, 0x08, 0xF8, 0x00, 0x03, 0x3C, 0x07, 0x00, 0x07, 0x3C, 0x03, 0x00,

/\*-- 文字: X --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x08, 0x18, 0x68, 0x80, 0x80, 0x68, 0x18, 0x08, 0x20, 0x30, 0x2C, 0x03, 0x03, 0x2C, 0x30, 0x20,

/\*-- 文字: Y --\*/  
 /\*-- Comic Sans MS12: 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x08, 0x38, 0xC8, 0x00, 0xC8, 0x38, 0x08, 0x00, 0x00, 0x00, 0x20, 0x3F, 0x20, 0x00, 0x00, 0x00,

/\*-- 文字: Z --\*/  
 /\*-- Comic Sans MS12: 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x10, 0x08, 0x08, 0x08, 0xC8, 0x38, 0x08, 0x00, 0x20, 0x38, 0x26, 0x21, 0x20, 0x20, 0x18, 0x00,

/\*-- 文字: [ --\*/  
 /\*-- Comic Sans MS12: 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0xFE, 0x02, 0x02, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7F, 0x40, 0x40, 0x40, 0x00,

/\*-- 文字: \ --\*/  
 /\*-- Comic Sans MS12: 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x0C, 0x30, 0xC0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x06, 0x38, 0xC0, 0x00,

/\*-- 文字: ] --\*/  
 /\*-- Comic Sans MS12: 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x02, 0x02, 0x02, 0xFE, 0x00, 0x00, 0x00, 0x00, 0x40, 0x40, 0x40, 0x7F, 0x00, 0x00, 0x00,

/\*-- 文字: ^ --\*/  
 /\*-- Comic Sans MS12: 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x04, 0x02, 0x02, 0x02, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

/\*-- 文字: \_ --\*/  
 /\*-- Comic Sans MS12: 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80,

/\*-- 文字: ` --\*/  
 /\*-- Comic Sans MS12: 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x02, 0x02, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

/\*-- 文字: a --\*/  
 /\*-- Comic Sans MS12: 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x19, 0x24, 0x22, 0x22, 0x22, 0x3F, 0x20,

/\*-- 文字: b --\*/  
 /\*-- Comic Sans MS12: 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x08, 0xF8, 0x00, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x11, 0x20, 0x20, 0x11, 0x0E, 0x00,

/\*-- 文字: c --\*/  
 /\*-- Comic Sans MS12: 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x0E, 0x11, 0x20, 0x20, 0x20, 0x11, 0x00,

/\*-- 文字: d --\*/

/\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0x80, 0x80, 0x88, 0xF8, 0x00, 0x00, 0x0E, 0x11, 0x20, 0x20, 0x10, 0x3F, 0x20,

/\*-- 文字: e --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x1F, 0x22, 0x22, 0x22, 0x22, 0x13, 0x00,

/\*-- 文字: f --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x80, 0x80, 0xF0, 0x88, 0x88, 0x88, 0x18, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,

/\*-- 文字: g --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x6B, 0x94, 0x94, 0x94, 0x93, 0x60, 0x00,

/\*-- 文字: h --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x08, 0xF8, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x20, 0x3F, 0x21, 0x00, 0x00, 0x20, 0x3F, 0x20,

/\*-- 文字: i --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x80, 0x98, 0x98, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,

/\*-- 文字: j --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0x80, 0x98, 0x98, 0x00, 0x00, 0x00, 0xC0, 0x80, 0x80, 0x80, 0x7F, 0x00, 0x00,

/\*-- 文字: k --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x08, 0xF8, 0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0x20, 0x3F, 0x24, 0x02, 0x2D, 0x30, 0x20, 0x00,

/\*-- 文字: l --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00,

/\*-- 文字: m --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x20, 0x3F, 0x20, 0x00, 0x3F, 0x20, 0x00, 0x3F,

/\*-- 文字: n --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x80, 0x80, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x20, 0x3F, 0x21, 0x00, 0x00, 0x20, 0x3F, 0x20,

/\*-- 文字: o --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x00, 0x1F, 0x20, 0x20, 0x20, 0x20, 0x1F, 0x00,

/\*-- 文字: p --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x80, 0x80, 0x00, 0x80, 0x80, 0x00, 0x00, 0x00, 0x80, 0xFF, 0xA1, 0x20, 0x20, 0x11, 0x0E, 0x00,

/\*-- 文字: q --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x0E, 0x11, 0x20, 0x20, 0xA0, 0xFF, 0x80,

/\*-- 文字: r --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x80, 0x80, 0x80, 0x00, 0x80, 0x80, 0x80, 0x00, 0x20, 0x20, 0x3F, 0x21, 0x20, 0x00, 0x01, 0x00,

/\*-- 文字: s --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x33, 0x24, 0x24, 0x24, 0x24, 0x19, 0x00,

/\*-- 文字: t --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x80, 0x80, 0xE0, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1F, 0x20, 0x20, 0x00, 0x00,

/\*-- 文字: u --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x80, 0x80, 0x00, 0x00, 0x00, 0x80, 0x80, 0x00, 0x00, 0x1F, 0x20, 0x20, 0x20, 0x10, 0x3F, 0x20,

/\*-- 文字: v --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x80, 0x80, 0x80, 0x00, 0x00, 0x80, 0x80, 0x80, 0x00, 0x01, 0x0E, 0x30, 0x08, 0x06, 0x01, 0x00,

/\*-- 文字: w --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x80, 0x80, 0x00, 0x80, 0x00, 0x80, 0x80, 0x80, 0x0F, 0x30, 0x0C, 0x03, 0x0C, 0x30, 0x0F, 0x00,

/\*-- 文字: x --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x80, 0x80, 0x00, 0x80, 0x80, 0x80, 0x00, 0x00, 0x20, 0x31, 0x2E, 0x0E, 0x31, 0x20, 0x00,

/\*-- 文字: y --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x80, 0x80, 0x80, 0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x81, 0x8E, 0x70, 0x18, 0x06, 0x01, 0x00,

/\*-- 文字: z --\*/  
 /\*-- Comic Sans MS12; 此字体下对应的点阵为: 宽 x 高=8x16 --\*/  
 0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x00, 0x00, 0x21, 0x30, 0x2C, 0x22, 0x21, 0x30, 0x00,

/\*-- 文字: { --\*/

```

/*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16  --*/
0x00, 0x00, 0x00, 0x00, 0x80, 0x7C, 0x02, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3F, 0x40, 0x40,

/*-- 文字： |  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16  --*/
0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00,

/*-- 文字： }  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16  --*/
0x00, 0x02, 0x02, 0x7C, 0x80, 0x00, 0x00, 0x00, 0x00, 0x40, 0x40, 0x3F, 0x00, 0x00, 0x00, 0x00,

/*-- 文字： ~  --*/
/*-- Comic Sans MS12; 此字体下对应的点阵为：宽 x 高=8x16  --*/
0x00, 0x06, 0x01, 0x01, 0x02, 0x02, 0x04, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

};

```

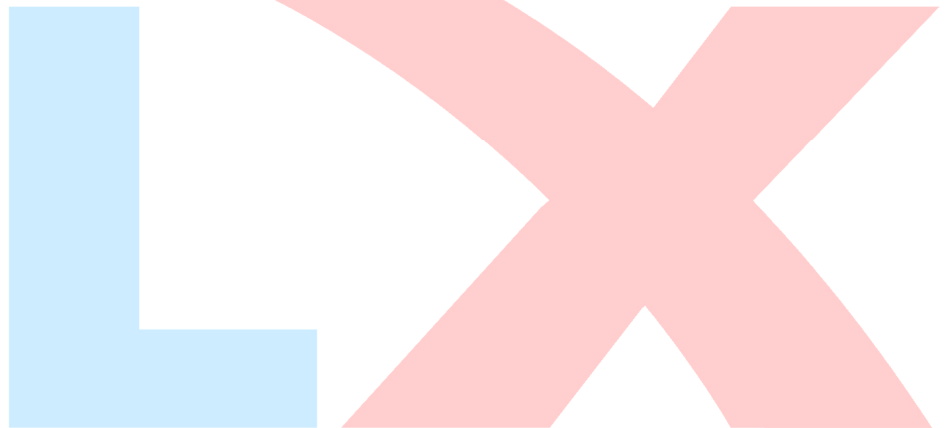
```

uchar code ascii_table_5x8[95][5]={
/*全体 ASCII 列表:5x8 点阵*/
0x00, 0x00, 0x00, 0x00, 0x00, //space
0x00, 0x00, 0x4f, 0x00, 0x00, //!
0x00, 0x07, 0x00, 0x07, 0x00, //"
0x14, 0x7f, 0x14, 0x7f, 0x14, //#
0x24, 0x2a, 0x7f, 0x2a, 0x12, //$
0x23, 0x13, 0x08, 0x64, 0x62, //%
0x36, 0x49, 0x55, 0x22, 0x50, //&
0x00, 0x05, 0x07, 0x00, 0x00, //]
0x00, 0x1c, 0x22, 0x41, 0x00, //(
0x00, 0x41, 0x22, 0x1c, 0x00, //)
0x14, 0x08, 0x3e, 0x08, 0x14, //*
0x08, 0x08, 0x3e, 0x08, 0x08, //+
0x00, 0x50, 0x30, 0x00, 0x00, //,
0x08, 0x08, 0x08, 0x08, 0x08, //-
0x00, 0x60, 0x60, 0x00, 0x00, //.
0x20, 0x10, 0x08, 0x04, 0x02, ///
0x3e, 0x51, 0x49, 0x45, 0x3e, //0
0x00, 0x42, 0x7f, 0x40, 0x00, //1
0x42, 0x61, 0x51, 0x49, 0x46, //2
0x21, 0x41, 0x45, 0x4b, 0x31, //3
0x18, 0x14, 0x12, 0x7f, 0x10, //4
0x27, 0x45, 0x45, 0x45, 0x39, //5
0x3c, 0x4a, 0x49, 0x49, 0x30, //6
0x01, 0x71, 0x09, 0x05, 0x03, //7
0x36, 0x49, 0x49, 0x49, 0x36, //8
0x06, 0x49, 0x49, 0x29, 0x1e, //9
0x00, 0x36, 0x36, 0x00, 0x00, //:

```



0x00, 0x56, 0x36, 0x00, 0x00, //;  
 0x08, 0x14, 0x22, 0x41, 0x00, //<  
 0x14, 0x14, 0x14, 0x14, 0x14, //=  
 0x00, 0x41, 0x22, 0x14, 0x08, //>  
 0x02, 0x01, 0x51, 0x09, 0x06, //?  
 0x32, 0x49, 0x79, 0x41, 0x3e, //@  
 0x7e, 0x11, 0x11, 0x11, 0x7e, //A  
 0x7f, 0x49, 0x49, 0x49, 0x36, //B  
 0x3e, 0x41, 0x41, 0x41, 0x22, //C  
 0x7f, 0x41, 0x41, 0x22, 0x1c, //D  
 0x7f, 0x49, 0x49, 0x49, 0x41, //E  
 0x7f, 0x09, 0x09, 0x09, 0x01, //F  
 0x3e, 0x41, 0x49, 0x49, 0x7a, //G  
 0x7f, 0x08, 0x08, 0x08, 0x7f, //H  
 0x00, 0x41, 0x7f, 0x41, 0x00, //I  
 0x20, 0x40, 0x41, 0x3f, 0x01, //J  
 0x7f, 0x08, 0x14, 0x22, 0x41, //K  
 0x7f, 0x40, 0x40, 0x40, 0x40, //L  
 0x7f, 0x02, 0x0c, 0x02, 0x7f, //M  
 0x7f, 0x04, 0x08, 0x10, 0x7f, //N  
 0x3e, 0x41, 0x41, 0x41, 0x3e, //O  
 0x7f, 0x09, 0x09, 0x09, 0x06, //P  
 0x3e, 0x41, 0x51, 0x21, 0x5e, //Q  
 0x7f, 0x09, 0x19, 0x29, 0x46, //R  
 0x46, 0x49, 0x49, 0x49, 0x31, //S  
 0x01, 0x01, 0x7f, 0x01, 0x01, //T  
 0x3f, 0x40, 0x40, 0x40, 0x3f, //U  
 0x1f, 0x20, 0x40, 0x20, 0x1f, //V  
 0x3f, 0x40, 0x38, 0x40, 0x3f, //W  
 0x63, 0x14, 0x08, 0x14, 0x63, //X  
 0x07, 0x08, 0x70, 0x08, 0x07, //Y  
 0x61, 0x51, 0x49, 0x45, 0x43, //Z  
 0x00, 0x7f, 0x41, 0x41, 0x00, //[  
 0x02, 0x04, 0x08, 0x10, 0x20, //\  
 0x00, 0x41, 0x41, 0x7f, 0x00, //]  
 0x04, 0x02, 0x01, 0x02, 0x04, //^  
 0x40, 0x40, 0x40, 0x40, 0x40, //\_  
 0x01, 0x02, 0x04, 0x00, 0x00, //`  
 0x20, 0x54, 0x54, 0x54, 0x78, //a  
 0x7f, 0x48, 0x48, 0x48, 0x30, //b  
 0x38, 0x44, 0x44, 0x44, 0x44, //c  
 0x30, 0x48, 0x48, 0x48, 0x7f, //d  
 0x38, 0x54, 0x54, 0x54, 0x58, //e  
 0x00, 0x08, 0x7e, 0x09, 0x02, //f  
 0x48, 0x54, 0x54, 0x54, 0x3c, //g  
 0x7f, 0x08, 0x08, 0x08, 0x70, //h





```

0x00, 0x00, 0x7a, 0x00, 0x00, //i
0x20, 0x40, 0x40, 0x3d, 0x00, //j
0x7f, 0x20, 0x28, 0x44, 0x00, //k
0x00, 0x41, 0x7f, 0x40, 0x00, //l
0x7c, 0x04, 0x38, 0x04, 0x7c, //m
0x7c, 0x08, 0x04, 0x04, 0x78, //n
0x38, 0x44, 0x44, 0x44, 0x38, //o
0x7c, 0x14, 0x14, 0x14, 0x08, //p
0x08, 0x14, 0x14, 0x14, 0x7c, //q
0x7c, 0x08, 0x04, 0x04, 0x08, //r
0x48, 0x54, 0x54, 0x54, 0x24, //s
0x04, 0x04, 0x3f, 0x44, 0x24, //t
0x3c, 0x40, 0x40, 0x40, 0x3c, //u
0x1c, 0x20, 0x40, 0x20, 0x1c, //v
0x3c, 0x40, 0x30, 0x40, 0x3c, //w
0x44, 0x28, 0x10, 0x28, 0x44, //x
0x04, 0x48, 0x30, 0x08, 0x04, //y
0x44, 0x64, 0x54, 0x4c, 0x44, //z
0x08, 0x36, 0x41, 0x41, 0x00, //{
0x00, 0x00, 0x77, 0x00, 0x00, //|
0x00, 0x41, 0x41, 0x36, 0x08, //}
0x04, 0x02, 0x02, 0x02, 0x01, //~
};
    
```

```

uchar code cheng1[]={
/*-- 文字: 成 --*/
/*-- 宋体 23; 此字体下对应的点阵为: 宽 x 高=31x31 --*/
/*-- 高度不是 8 的倍数, 现调整为: 宽度 x 高度=32x32 --*/
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0C,
0xFC, 0xFC, 0x88, 0x00, 0x00, 0x1C, 0x78, 0xF0, 0xE0, 0x00, 0x80, 0x80, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x83, 0x83, 0x83, 0x83, 0x83, 0x83, 0xC3, 0xC3, 0x03, 0x1F,
0xFF, 0xFF, 0x83, 0x03, 0x03, 0x03, 0xC3, 0xF3, 0xF3, 0x63, 0x03, 0x03, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0xFC, 0xFF, 0x3F, 0x00, 0x80, 0x00, 0x00, 0x80, 0xFF, 0xFF, 0x03, 0x00, 0x00, 0x03,
0x9F, 0xFF, 0xF8, 0xF8, 0xBE, 0x1F, 0x07, 0x01, 0x00, 0x00, 0xE0, 0x20, 0x00, 0x00, 0x20, 0x38,
0x1F, 0x07, 0x01, 0x00, 0x00, 0x01, 0x01, 0x07, 0x07, 0x23, 0x31, 0x18, 0x0C, 0x0E, 0x07, 0x03,
0x01, 0x01, 0x01, 0x03, 0x07, 0x0F, 0x0E, 0x1C, 0x1F, 0x3F, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00};
    
```

```

uchar code gong1[]={
/*-- 文字: 功 --*/
/*-- 宋体 23; 此字体下对应的点阵为: 宽 x 高=31x31 --*/
/*-- 高度不是 8 的倍数, 现调整为: 宽度 x 高度=32x32 --*/
0x00, 0x00, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0x80, 0xC0, 0xC0, 0xC0, 0x00,
0x00, 0x00, 0x00, 0xFE, 0xFC, 0x0C, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x0C, 0x04, 0x04,
0x04, 0x84, 0xFF, 0xFF, 0x04, 0x04, 0x04, 0x04, 0x04, 0xFE, 0xFE, 0x04, 0x00, 0x00, 0x00, 0x00,
0xC0, 0xC0, 0xC0, 0xC0, 0xE0, 0x60, 0x7F, 0x3F, 0x30, 0x30, 0x10, 0x18, 0x18, 0x88, 0xC0, 0xF8,
    
```

```
0x7F, 0x1F, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01,
0x01, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x30, 0x18, 0x0C, 0x0C, 0x07, 0x03, 0x01, 0x00, 0x04,
0x04, 0x0C, 0x0C, 0x1C, 0x38, 0x3C, 0x1F, 0x0F, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
```

```
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 tai1[]={
/*-- 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 shi1[]={
/*-- 文字: 使 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x40, 0x20, 0xF0, 0x1C, 0x07, 0xF2, 0x94, 0x94, 0x94, 0xFF, 0x94, 0x94, 0x94, 0xF4, 0x04, 0x00,
0x00, 0x00, 0x7F, 0x00, 0x40, 0x41, 0x22, 0x14, 0x0C, 0x13, 0x10, 0x30, 0x20, 0x61, 0x20, 0x00};
```

```
uchar code yong1[]={
/*-- 文字: 用 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x00, 0x00, 0x00, 0xFE, 0x22, 0x22, 0x22, 0x22, 0xFE, 0x22, 0x22, 0x22, 0x22, 0xFE, 0x00, 0x00,
0x80, 0x40, 0x30, 0x0F, 0x02, 0x02, 0x02, 0x02, 0xFF, 0x02, 0x02, 0x42, 0x82, 0x7F, 0x00, 0x00};
```

```
uchar code bmp_12864_1[]={
/*-- 调入了一幅图像: D:\e\新开发部\显示图案收藏\12864G-202 回字框.bmp --*/
/*-- 宽度 x 高度=128x64 --*/
0xFF, 0x01, 0x01, 0x01, 0x01, 0xF9, 0x09, 0x09, 0x09, 0x09, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89,
0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89,
0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89,
0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89,
0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89, 0x89,
.
.
.
}
```

串行接口:

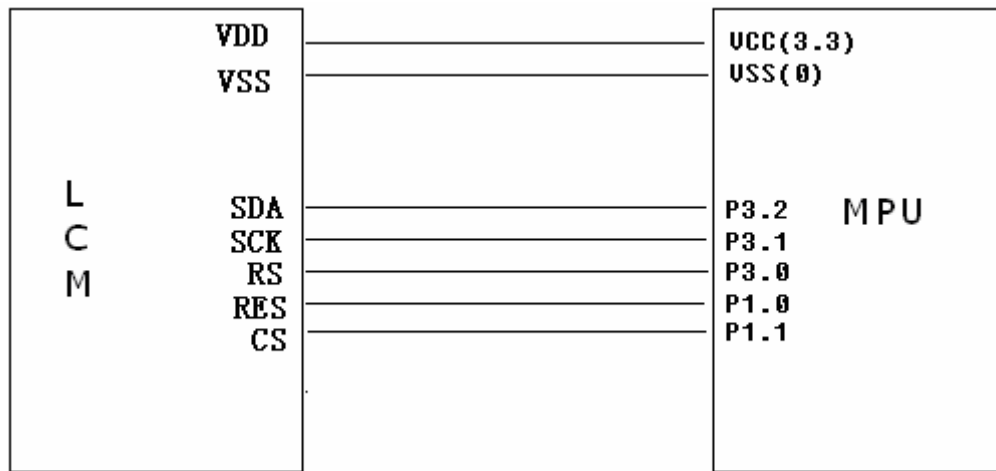
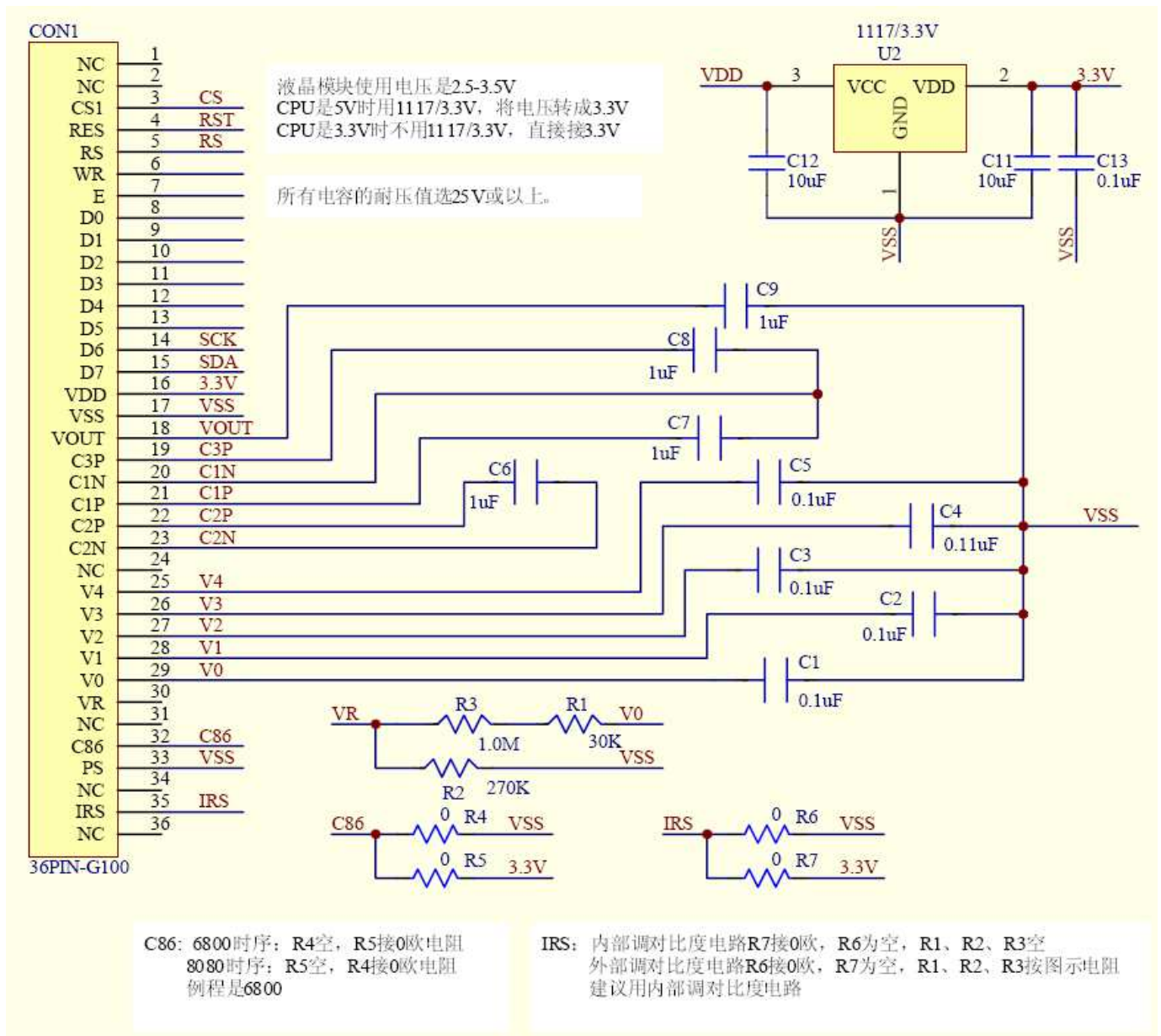


图 9. 串行接口



串行调对比度电路图 (建议用内部调对比度电路, IRS 接 3.3V)



7.52、以下为串行方式例程序

与并行方式相比较, 只需改变传送数据和命令子程序即可

```
#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 graphicl[];
```

```
void delay(int i);
void delay1(int i);
void displaygraphic(char *dp);

void Initial_Lcd();
void clear_screen();
void waitkey();

/*=====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(1);
        sclk=1;
        delay1(1);
        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;
    }
}
```