



COG12832C3G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	7~11
7	指令功能及硬件接口与编程案例	11~37

1. 概述

科飞研科技专注于液晶屏及液晶模块的研发、制造。所生产COG12832C3G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。

COG12832C3G可以显示128列*32行点阵单色图片，或显示8个/行*2行16*16点阵的汉字，或显示16个/行*4行8*8点阵的英文、数字、符号。

2. COG12832C3G 型点阵液晶模块的特性

- 2.1 结构牢：背光带有挡墙。焊接式 FPC，省去使用者购买连接座的成本
- 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*32 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*2 行。按照 12*12 点阵汉字来计算可显示 10 字/行*2 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℃；
- 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

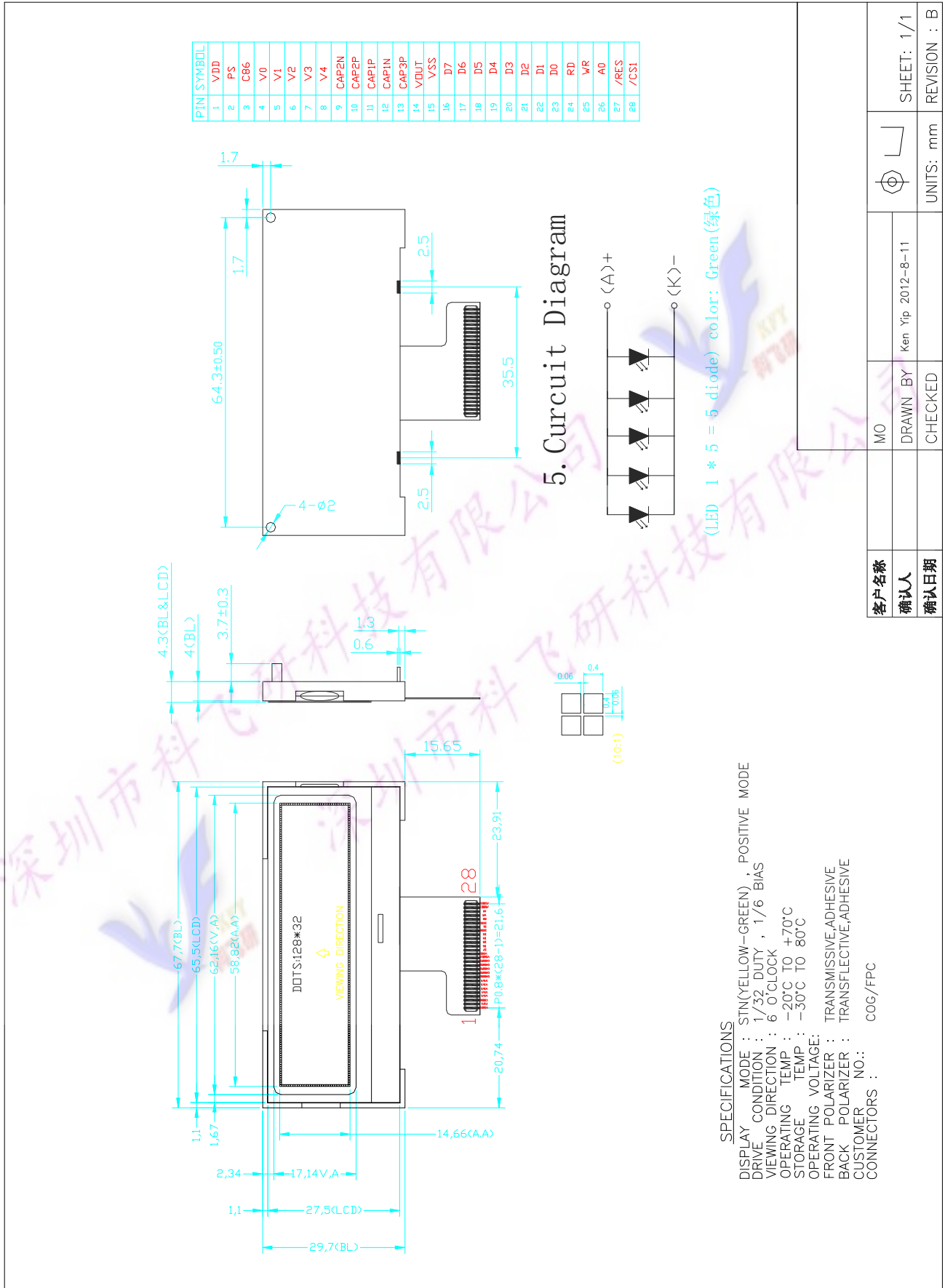


图 1. 外形尺寸

模块的接口引脚功能

引线号	符号	名称	功能
1	VDD	电路电源	3.3V
2	PS	选串并控制接口	H: 并行接口, 0: 串行接口
3	C86	选择 6800 或 8080	H: 6800 系统, 0: 是 8080 系统
4	V0	偏置电压	LCD 驱动偏置电压
5	V1	偏置电压	LCD 驱动偏置电压
6	V2	偏置电压	LCD 驱动偏置电压
7	V3	偏置电压	LCD 驱动偏置电压
8	V4	偏置电压	LCD 驱动偏置电压
9	CAP2N	升压电容	倍压电路
10	CAP2P	升压电容	倍压电路
11	CAP1P	升压电容	倍压电路
12	CAP1N	升压电容	倍压电路
13	CAP3P	升压电容	倍压电路
14	VOUT	分压电阻	LCD 倍压输出
15	VSS	接地	0V
16-23	D7-D0	I/O	数据总线
24	RD	使能信号	读/写模式
25	WR	读/写	H: 读数据 0: 写数据
26	A0	寄存器选择信号	H: 数据寄存器 0: 指令寄存器
27	/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
28	/CS1	片选	低电平片选

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 128×32 点阵, 128 个列信号与驱动 IC 相连, 32 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COG12832C3G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。

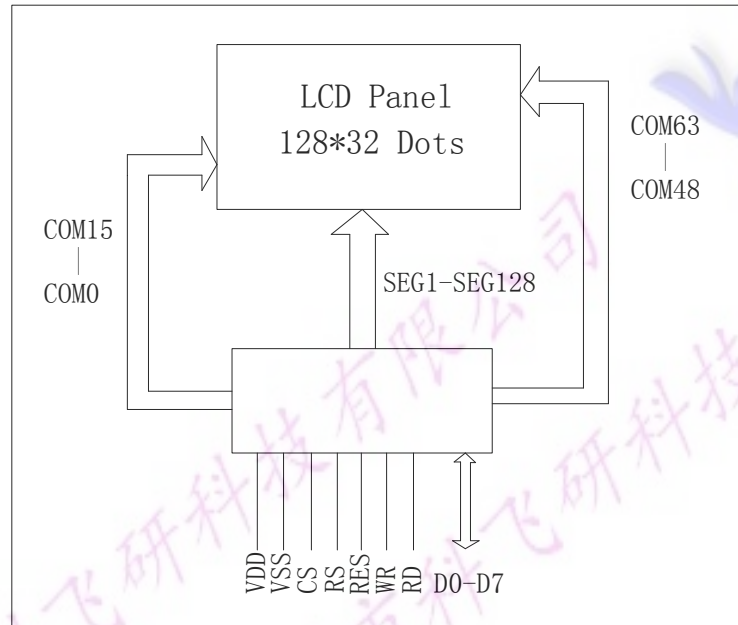


图 2: COG12832C3G 图像点阵型液晶模块的电路框图

4.2 升压电路图:

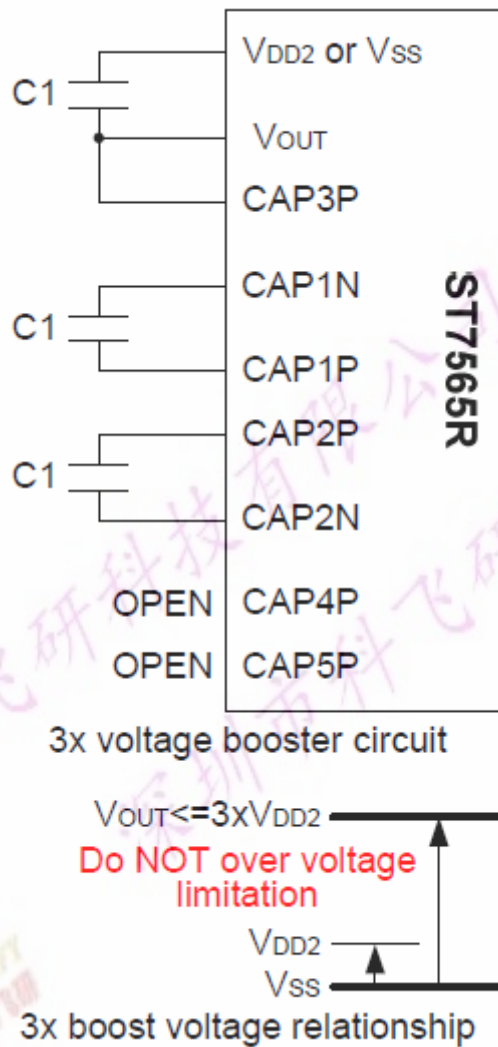


图 3: 3 倍升压电路图

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: $50 \sim 100 \text{mA}$ (LED 灯数共 5 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	50	75	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 ST7565R (Writing Data from CPU to ST7565R)

The 4-line SPI Interface

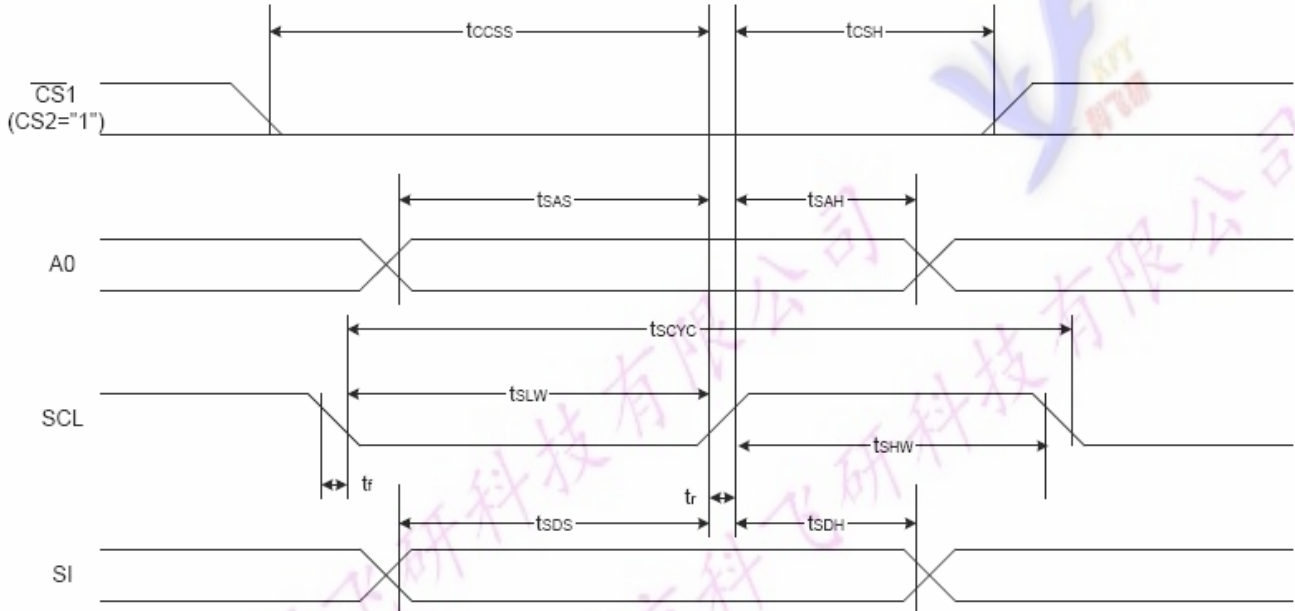


图 4. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

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)

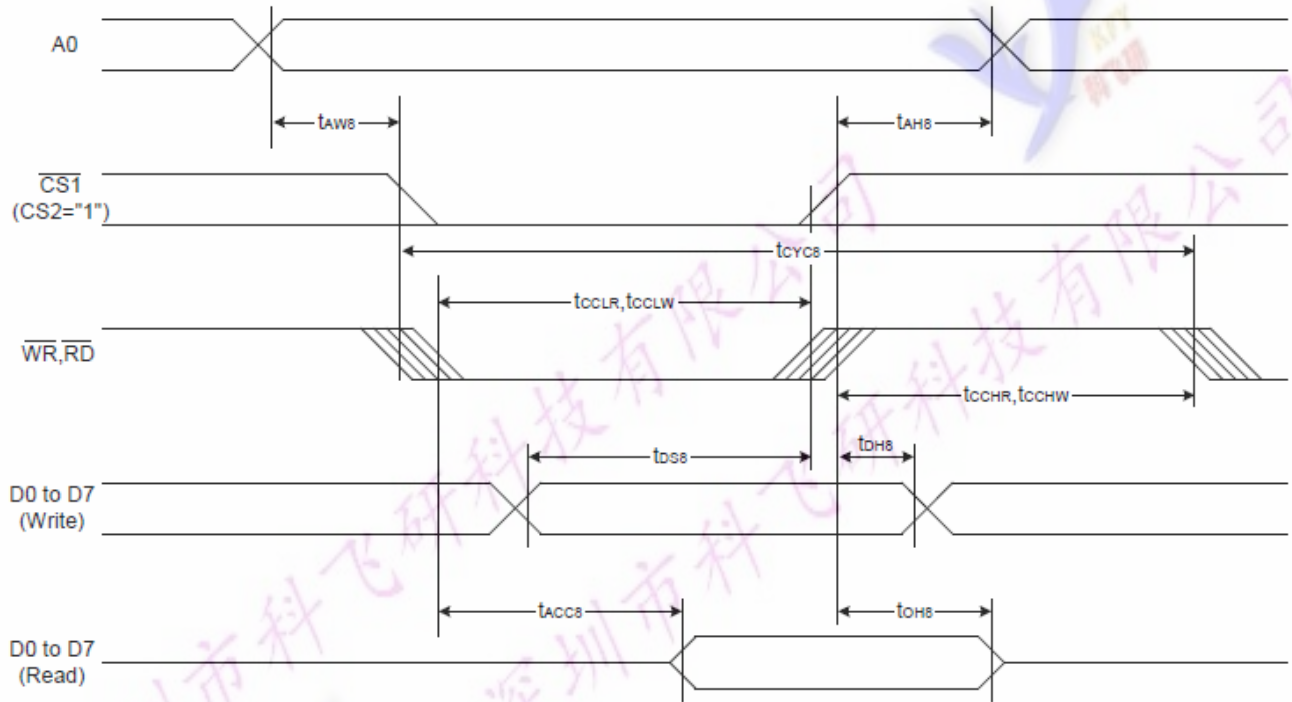


图 5. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

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

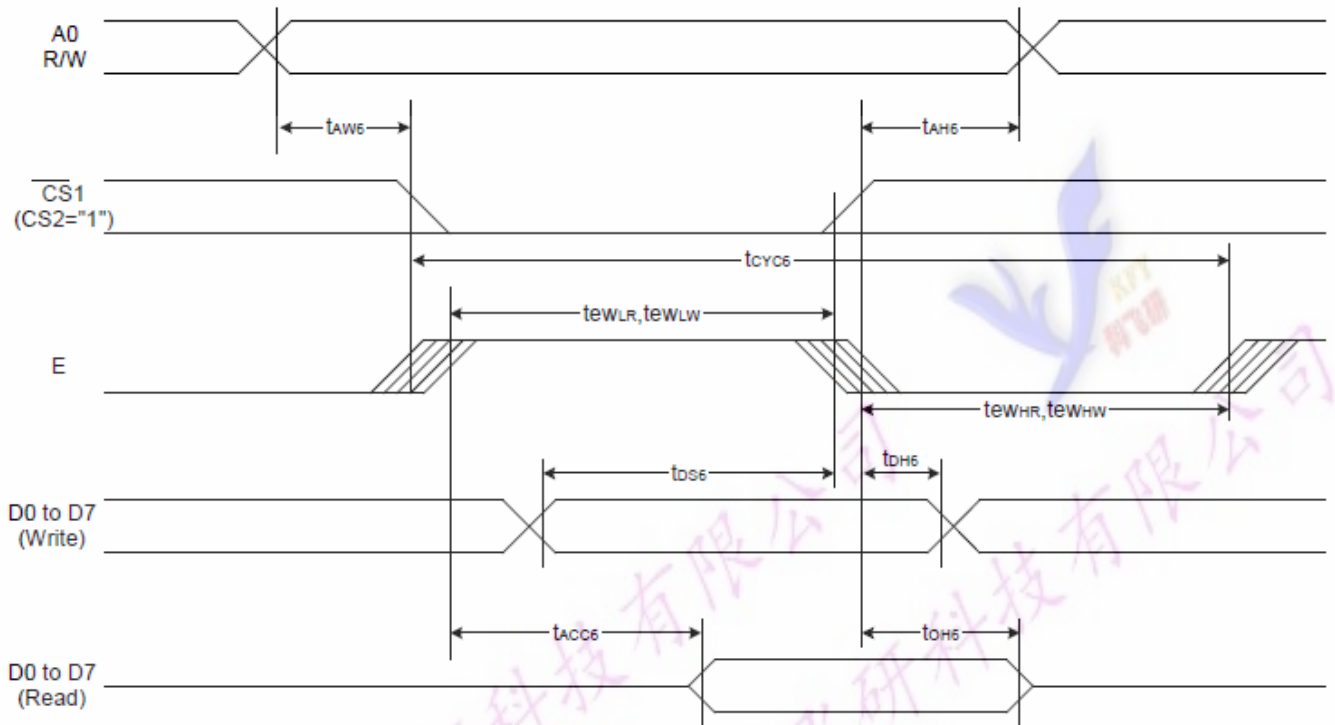


图 6. 从 CPU 写到 ST7565R (Writing Data from CPU to ST7565R)

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		--	
读时间		tACC8	--		70	
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）:

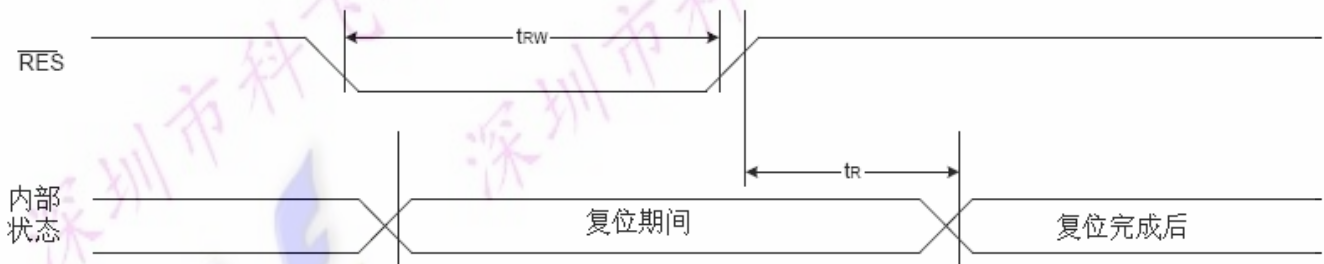


图 7：电源启动后复位的时序

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

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 00000001, 所以此指令表达为: 0x10, 0x01	
		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)		1	1	0	0	0	0	0	0 1	行扫描顺序选择: 0: 普通顺序

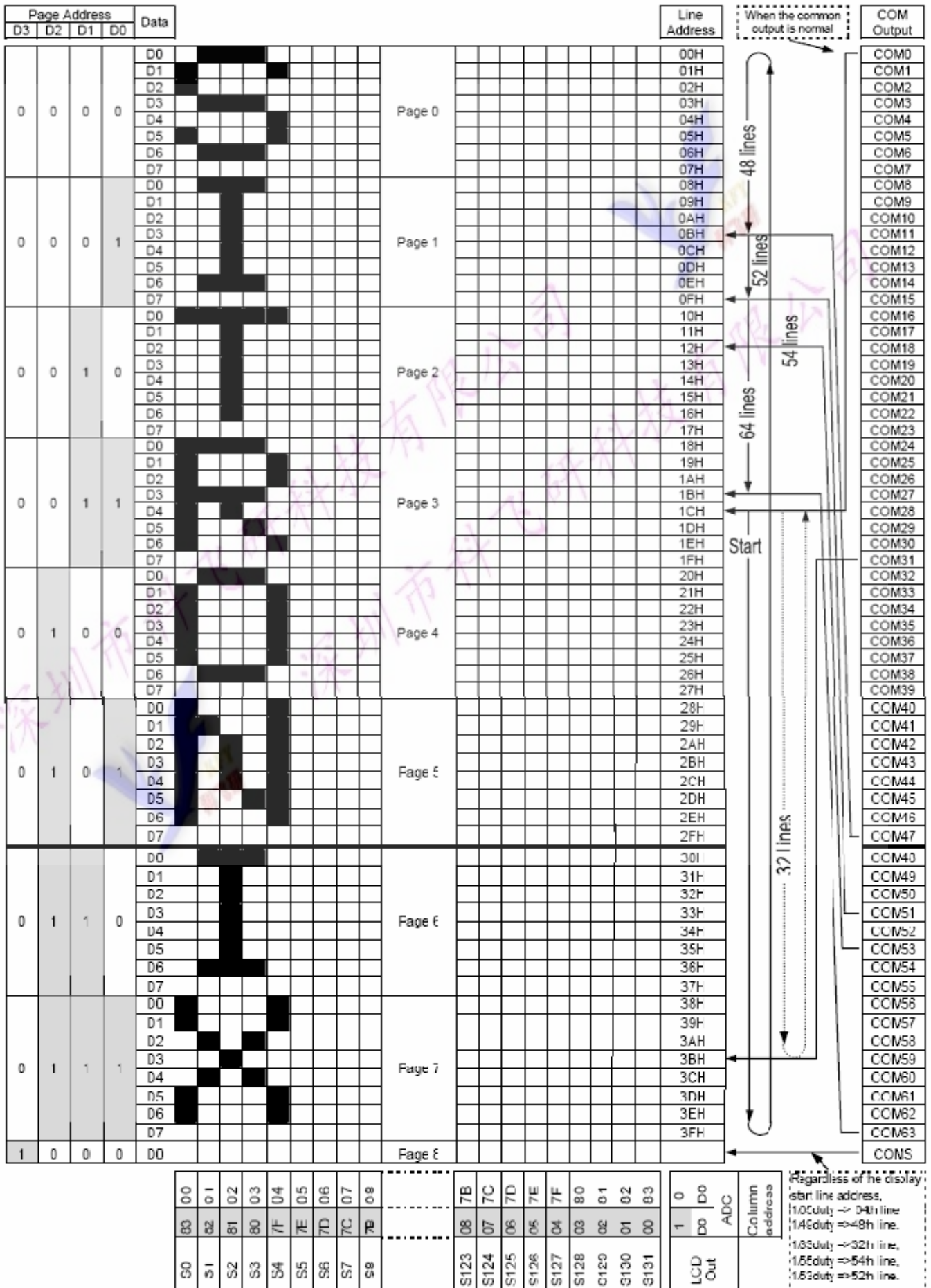
output mode select)										1: 反向扫描	
(16) 电源控制 (Power control set)		0	0	1	0	1	电压操作模式选择, 共 3 位			选择内部电压供应操作模式	
(17) 选择内部电阻比例		0	0	0	1	0	0	内部电压值电阻设置		选择内部电阻比例 (Rb/Ra), 本液晶模块通过外置电阻设置, 此指令失效	
(18)	内部设置液晶电压模式	0	1	0	0	0	0	0	1	设置内部电阻微调, 以设置液晶电压, 此两个指令需紧接着使用	
	设置的电压值		0	0	6 位电压值数据, 0~63 共 64 级						
(19)静态图标显示: 开/关		0	1	0	1	0	1	1	0	0 1	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	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”。
(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:

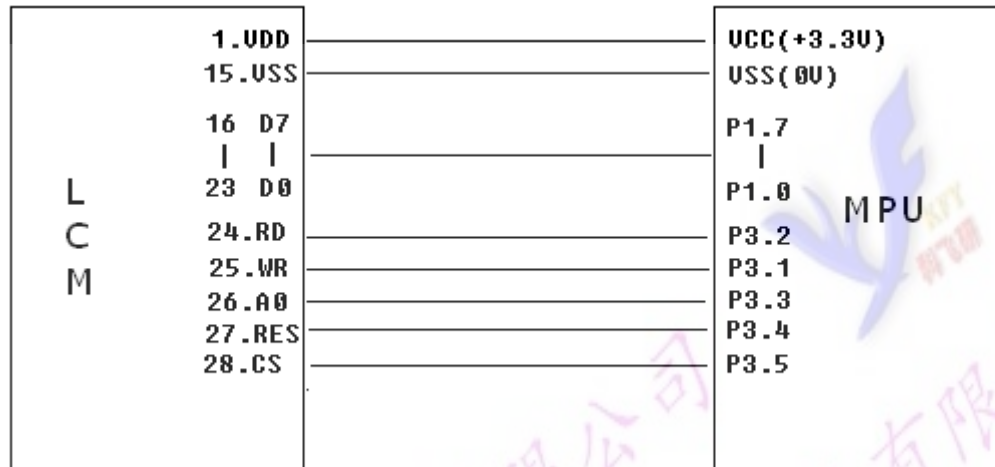
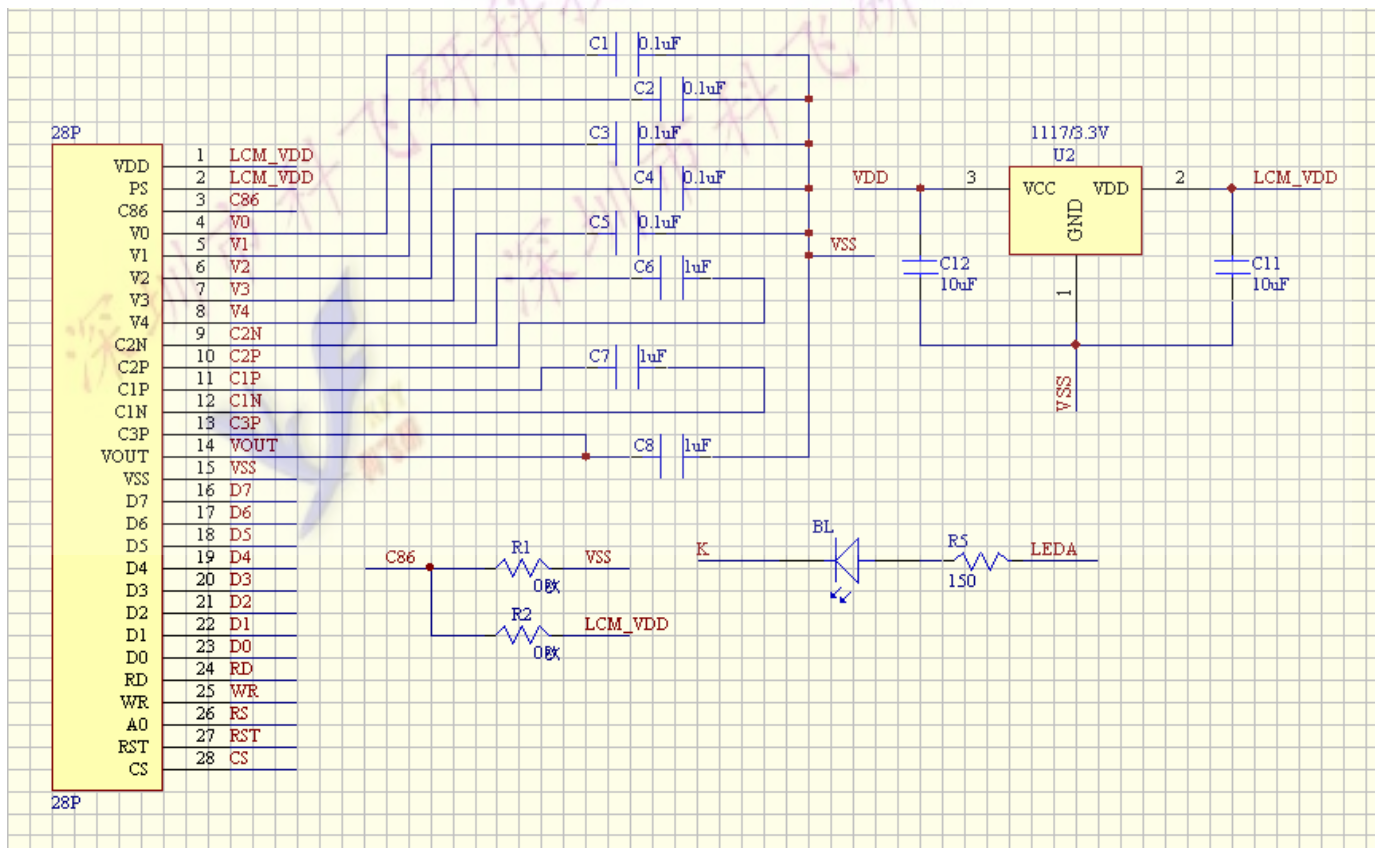


图 8. 并行接口





并程序:

```
/* COG12832C3G测试程序****/  
/* LCD 驱动 IC:ST7565R*****/  
/* 该程序显示 2 行中文如下: */  
/* 全套液晶解决方案*/  
/* 质量取胜创建口碑*/  
/* 该程序显示 2 行数字如下: */  
/* 0123456789012345*/  
/* 0123456789012345*/
```

```
#include <reg51.h>
```

```
sbit cs1=P3^5; /*接口定义, 按 51 系列单片机, P1 口接 DB0~DB7*/  
sbit reset=P3^4;  
sbit rs=P3^3;  
sbit rd=P3^2;  
sbit wr=P3^1; /*接口定义, 按 51 系列单片机, P1 口接 DB0~DB7*/
```

```
void transfer_data(int data1);  
void transfer_command(int data1);  
char code quan[];  
char code tao[];  
char code ye[];  
char code jing[];  
char code jie[];  
char code jue[];  
char code fang[];  
char code an[];  
char code num0[];  
char code num1[];  
char code num2[];  
char code num3[];  
char code num4[];  
char code num5[];  
char code num6[];  
char code num7[];  
char code num8[];
```




```
char code num9[];

char code graphic1[];
char code graphic1[];
char code graphic2[];
char code graphic3[];
char code graphic4[];

void displaygraphic(char *dp);
void delay(int i);
void delay1(int i);
void disp_grap(int line,int column, char *dp);
void disp_char(int line,int column, 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(1,1,quan); /*在第1行第1列显示一个 16*16 点阵汉字:全*/
        //delay(20);
        disp_grap(1,2,tao); /*在第1行第2列显示一个 16*16 点阵汉字:套*/
        //delay(20);
        disp_grap(1,3,ye); /*显示一个 16*16 点阵汉字:液*/
        //delay(20);
        disp_grap(1,4,jing); /*显示一个 16*16 点阵汉字:晶*/
        //delay(20);
        disp_grap(1,5,jie); /*显示一个 16*16 点阵汉字:解*/
        //delay(20);
        disp_grap(1,6,jue); /*显示一个 16*16 点阵汉字:决*/
        //delay(20);
        disp_grap(1,7,fang); /*显示一个 16*16 点阵汉字:方*/
        //delay(20);
        disp_grap(1,8,an); /*显示一个 16*16 点阵汉字:案*/
        //delay(20);
    }
}
```



```
disp_grap(2,4,ye); /*显示一个 16*16 点阵汉字: 液*/
//delay(20);
disp_grap(2,5,jing); /*显示一个 16*16 点阵汉字: 晶*/
//delay(20);
disp_grap(2,6,mo); /*显示一个 16*16 点阵汉字: 模*/
//delay(20);
disp_grap(2,7,kuai); /*显示一个 16*16 点阵汉字: 块*/
//delay(20);
disp_grap(2,8,chang); /*显示一个 16*16 点阵汉字: 厂*/
waitkey();
clear_screen(); //clear all dots
disp_char(1,1,num0); /*在第 1 行第 1 列显示一个 8*16 点阵数字:0*/
disp_char(1,2,num1); /*在第 1 行第 2 列显示一个 8*16 点阵数字:1*/
disp_char(1,3,num2); /*在第 1 行第 3 列显示一个 8*16 点阵数字:2*/
disp_char(1,4,num3); /*在第 1 行第 4 列显示一个 8*16 点阵数字:3*/
disp_char(1,5,num4); /*在第 1 行第 5 列显示一个 8*16 点阵数字:4*/
disp_char(1,6,num5); /*以下类同*/
disp_char(1,7,num6);
disp_char(1,8,num7);
disp_char(1,9,num8);
disp_char(1,10,num9);
disp_char(1,11,num0);
disp_char(1,12,num1);
disp_char(1,13,num2);
disp_char(1,14,num3);
disp_char(1,15,num4);
disp_char(1,16,num5);
disp_char(2,1,num0); /*在第 2 行第 1 列显示一个 8*16 点阵数字:0*/
disp_char(2,2,num1); /*在第 2 行第 2 列显示一个 8*16 点阵数字:1*/
disp_char(2,3,num2); /*在第 2 行第 3 列显示一个 8*16 点阵数字:2*/
disp_char(2,4,num3); /*在第 2 行第 4 列显示一个 8*16 点阵数字:3*/
disp_char(2,5,num4); /*在第 2 行第 5 列显示一个 8*16 点阵数字:4*/
disp_char(2,6,num5); /*以下类同*/
disp_char(2,7,num6);
disp_char(2,8,num7);
disp_char(2,9,num8);
disp_char(2,10,num9);
disp_char(2,11,num0);
disp_char(2,12,num1);
disp_char(2,13,num2);
disp_char(2,14,num3);
disp_char(2,15,num4);
disp_char(2,16,num5);
```



```
waitkey();
clear_screen(); //clear all dots
displaygraphic(graphic1); //display a picture of 128*64 dots
waitkey();
clear_screen(); //clear all dots
displaygraphic(graphic2); //display a picture of 128*64 dots
waitkey();
clear_screen(); //clear all dots
displaygraphic(graphic3); //display a picture of 128*64 dots
waitkey();
clear_screen(); //clear all dots
displaygraphic(graphic4); //display a picture of 128*64 dots
waitkey();
}
}
```

```
//=====initial
```

```
void initial_lcd()
{
    reset=0; //Reset the chip when reset=0
    delay(20);
    reset=1;
    transfer_command(0xe2); /*软复位*/
    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(0x22); /*粗调对比度, 可设置范围 20~27*/
    transfer_command(0x81); /*微调对比度*/
    transfer_command(0x1b); /*微调对比度的值, 可设置范围 0~63*/
    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: 从第一行开始*/
    transfer_command(0xaf); /*开显示*/
}
```

```
//=====clear all dot martrics=====
```

```
void clear_screen()
{
    unsigned char i, j;
```

```
    for(i=0; i<4; i++)
```

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

/*在指定行和列位置显示指定的汉字（16*16 点阵的汉字）*/

```
void disp_grap(int line,int column,char *dp)
{
    int i,j,k,col_l,col_h;
    for(i=0;i<2;i++)
    {
        cs1=0;
        transfer_command(0xb0+i+2*(line-1)); //set page address,
        k=column-1;
        k=k*0x10;
        col_h=k&0xf0;
        col_h=col_h>>4;
        col_l=k&0x0f;
        transfer_command(0x10+col_h);
        transfer_command(0x00+col_l);
        for(j=0;j<16;j++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}
```

/*在指定行和列位置显示指定的字母、数字（8*16 点阵的）*/

```
void disp_char(int line,int column,char *dp)
{
    int i,j,k,col_l,col_h;
    for(i=0;i<2;i++)
    {
        cs1=0;
        transfer_command(0xb0+i+2*(line-1)); //set page address,
        k=column-1;
        k=k*0x08;
        col_h=k&0xf0;
        col_h=col_h>>4;
```

```
col_l=k&0x0f;
transfer_command(0x10+col_h);
transfer_command(0x00+col_l);
for(j=0;j<8;j++)
{
    transfer_data(*dp);
    dp++;
}
}
```

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

```
void displaygraphic(char *dp)
{
    int i, j;
    for(i=0;i<4;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++;
        }
    }
}
```

/*=====写指令=====*/

```
void transfer_command(int data1)
{
    cs1=0;
    rs=0;
    rd=0;
    wr=0;
    P1=data1;
    rd=1;
    cs1=1;
    rd=0;
}
```

/*-----写数据-----*/

```
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(5);
        if (P2&0x01) goto repeat;
        else;
        delay(40);
}

//-----

char code quan[]={
/*-- 文字: 全 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x00,0x80,0x40,0x60,0x50,0x48,0x44,0xC3,0x44,0x48,0x50,0x70,0x60,0x20,0x00,0x00,
0x00,0x40,0x40,0x44,0x44,0x44,0x44,0x7F,0x44,0x44,0x44,0x46,0x44,0x60,0x40,0x00
};
```



```
char code tao[]={
/*-- 文字：套 --*/
/*-- 宋体 12； 此字体下对应的点阵为：宽 x 高=16x16 --*/
0x40,0x44,0x24,0x24,0xF4,0x5C,0x57,0x54,0x54,0x5C,0x54,0x24,0x64,0xC4,0x44,0x00,
0x04,0x04,0x44,0x64,0x57,0x4D,0x45,0x65,0x25,0x25,0x35,0xE4,0x44,0x04,0x04,0x00
};

char code ye[]={

/*-- 文字：液 --*/
/*-- 宋体 12； 此字体下对应的点阵为：宽 x 高=16x16 --*/
0x10,0x61,0x06,0xE0,0x18,0x84,0xE4,0x1C,0x84,0x65,0xBE,0x24,0xA4,0x64,0x04,0x00,
0x04,0x04,0xFF,0x00,0x01,0x00,0xFF,0x41,0x21,0x12,0x0C,0x1B,0x61,0xC0,0x40,0x00
};

char code jing[]={
/*-- 文字：晶 --*/
/*-- 宋体 12； 此字体下对应的点阵为：宽 x 高=16x16 --*/
0x00,0x00,0x00,0x00,0x7E,0x2A,0x2A,0x2A,0x2A,0x2A,0x2A,0x7E,0x00,0x00,0x00,0x00,
0x00,0x7F,0x25,0x25,0x25,0x25,0x7F,0x00,0x00,0x7F,0x25,0x25,0x25,0x25,0x7F,0x00
};

char code jie[]={
/*-- 文字：解 --*/
/*-- 宋体 12； 此字体下对应的点阵为：宽 x 高=16x16 --*/
0x08,0xF4,0x57,0x54,0xFC,0x54,0xF0,0x42,0xA2,0x1E,0x02,0xA2,0x62,0x3E,0x00,0x00,
0x80,0x7F,0x02,0x02,0x5F,0x82,0x7F,0x0A,0x09,0x09,0x09,0xFF,0x09,0x09,0x09,0x00
};

char code jue[]={
/*-- 文字：决 --*/
/*-- 宋体 12； 此字体下对应的点阵为：宽 x 高=16x16 --*/
0x02,0x04,0xD8,0x08,0x00,0x08,0x08,0x08,0xFF,0x08,0x08,0x08,0xF8,0x00,0x00,0x00,
0x02,0xFE,0x01,0x80,0x41,0x21,0x11,0x0D,0x03,0x05,0x09,0x11,0x31,0x61,0x21,0x00
};

char code fang[]={

/*-- 文字：方 --*/
/*-- 宋体 12； 此字体下对应的点阵为：宽 x 高=16x16 --*/
0x08,0x08,0x08,0x08,0x08,0x08,0xF9,0x4A,0x4C,0x48,0x48,0xC8,0x08,0x08,0x08,0x00,
0x40,0x40,0x20,0x10,0x0C,0x03,0x00,0x00,0x20,0x40,0x40,0x3F,0x00,0x00,0x00,0x00
};
```



```
char code an[]={
/*-- 文字： 案 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x00, 0x20, 0x2C, 0x24, 0x64, 0x74, 0xAD, 0xA6, 0xE4, 0x34, 0x24, 0x24, 0x2C, 0x24, 0x00, 0x00,
0x00, 0x24, 0x24, 0x25, 0x15, 0x15, 0x0D, 0xFE, 0x04, 0x0D, 0x17, 0x14, 0x24, 0x64, 0x24, 0x00
};
```

```
char code mo[]={
/*-- 文字： 模 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x10, 0xD0, 0xFF, 0x50, 0x90, 0x04, 0xF4, 0x54, 0x5F, 0x54, 0x54, 0x5F, 0xF4, 0x04, 0x00, 0x00,
0x03, 0x00, 0xFF, 0x00, 0x00, 0x84, 0x85, 0x45, 0x35, 0x0F, 0x15, 0x25, 0x65, 0xC4, 0x44, 0x00
};
```

```
char code kuai[]={
/*-- 文字： 块 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x10, 0x10, 0xFF, 0x10, 0x10, 0x00, 0x08, 0x08, 0xFF, 0x08, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00,
0x08, 0x18, 0x0F, 0x04, 0x85, 0x41, 0x31, 0x0D, 0x03, 0x05, 0x09, 0x11, 0x31, 0x61, 0x21, 0x00
};
```

```
char code chang[]={
/*-- 文字： 厂 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x00, 0x00, 0xFE, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x00,
0x40, 0x30, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};
```

```
char code num0[]={
/*-- 文字： 0 --*/
/*-- 宋体 12； 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
};
```

```
char code num1[]={
/*-- 文字： 1 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=8x16 --*/
0x00, 0x10, 0x10, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x20, 0x20, 0x3F, 0x20, 0x20, 0x00, 0x00
};
```




```
char code num2[]={
/*-- 文字: 2 --*/
/*-- 宋体 12; 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
};
char code num3[]={
/*-- 文字: 3 --*/
/*-- 宋体 12; 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
};
char code num4[]={
/*-- 文字: 4 --*/
/*-- 宋体 12; 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
};

char code num5[]={
/*-- 文字: 5 --*/
/*-- 宋体 12; 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
};

char code num6[]={
/*-- 文字: 6 --*/
/*-- 宋体 12; 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
};
char code num7[]={
/*-- 文字: 7 --*/
/*-- 宋体 12; 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
};
char code num8[]={
/*-- 文字: 8 --*/
/*-- 宋体 12; 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
};

char code num9[]={
/*-- 文字: 9 --*/
/*-- 宋体 12; 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
};
```

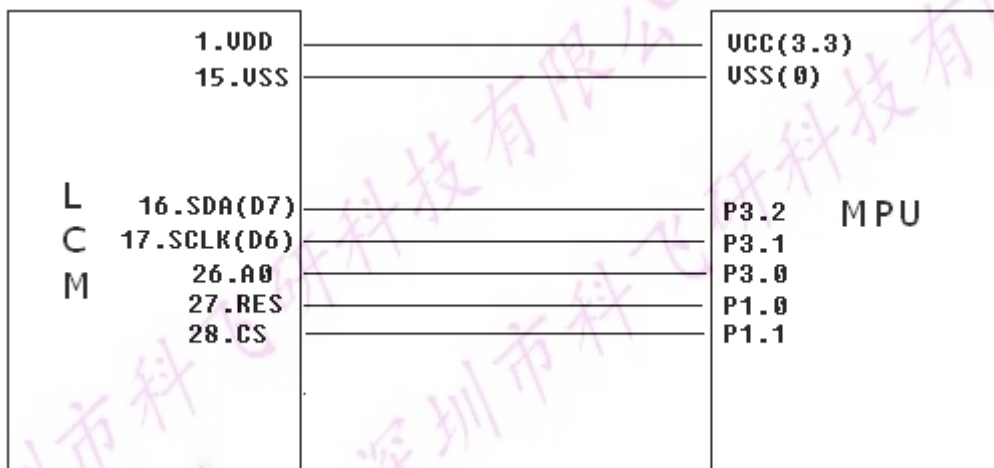
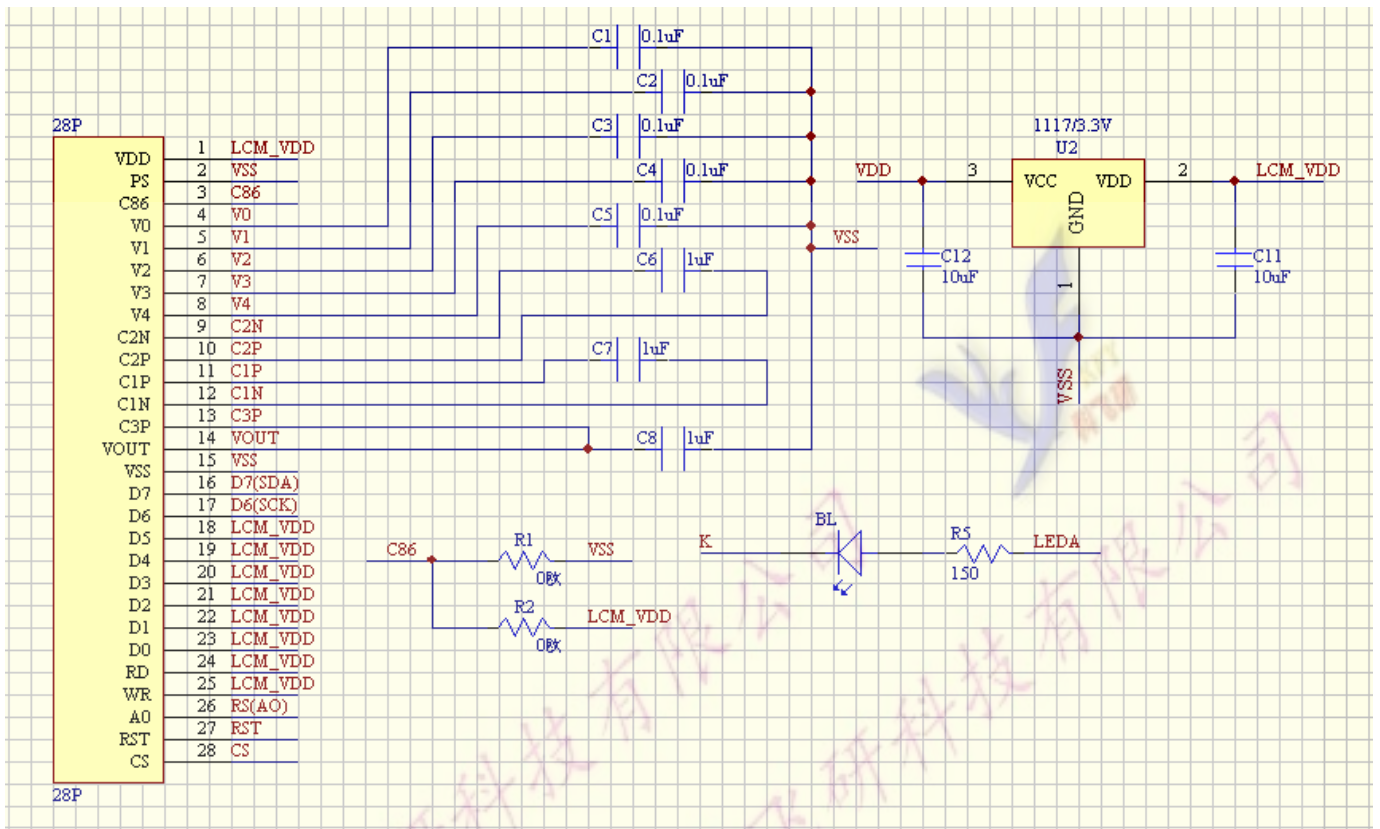


图 9. 串行接口



串程序序:

```

/* COG12832C3G测试程序****/
/* LCD 驱动 IC:ST7565R*****/
/* 串行接口*/
/* 该程序显示 2 行中文如下: */
/* 全套液晶解决方案*/
/* 质量取胜创建口碑*/
/* 该程序显示 2 行数字如下: */
/* 0123456789012345*/
/* 0123456789012345*/
    
```

```
#include <reg51.h>
```

```

sbit cs1=P1^1;    //接口定义, 按 51 系列单片机
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 quan[];
char code tao[];
char code ye[];
char code jing[];
char code jie[];
char code jue[];
char code fang[];
char code an[];
char code mo[];
char code kuai[];
char code chang[];
char code num0[];
char code num1[];
char code num2[];
char code num3[];
char code num4[];
char code num5[];
char code num6[];
char code num7[];
char code num8[];
char code num9[];

void delay(int i);
void delay1(int i);
void disp_grap(int line,int column,char *dp);
void disp_char(int line,int column,char *dp);
void initial_lcd();
void clear_screen();
void disp_chinese();
void disp_numeric();
void disp_ascill();
void Switch();

//=====main program=====
void main(void)
{
    int i,j,k;
    initial_lcd();          /*初始化 LCD*/
    while(1)
    {

        clear_screen();    /*清屏*/
        disp_chinese();    /*显示两行汉字*/
```



```
        delay(200);    /*延时*/
clear_screen();      /*清屏*/
disp_numeric();     /*显示两行数字*/
        delay(200);
        clear_screen();
        disp_asciil();
    }
}

void disp_chinese()
{
    disp_grap(1, 1, quan); /*在第 1 行第 1 列显示一个 16*16 点阵汉字:全*/
    //delay(20);
    disp_grap(1, 2, tao); /*在第 1 行第 2 列显示一个 16*16 点阵汉字:套*/
    //delay(20);
    disp_grap(1, 3, ye); /*显示一个 16*16 点阵汉字:液*/
    //delay(20);
    disp_grap(1, 4, jing); /*显示一个 16*16 点阵汉字:晶*/
    //delay(20);
    disp_grap(1, 5, jie); /*显示一个 16*16 点阵汉字:解*/
    //delay(20);
    disp_grap(1, 6, jue); /*显示一个 16*16 点阵汉字:决*/
    //delay(20);
    disp_grap(1, 7, fang); /*显示一个 16*16 点阵汉字:方*/
    //delay(20);
    disp_grap(1, 8, an); /*显示一个 16*16 点阵汉字:案*/
    //delay(20);

    disp_grap(2, 4, ye); /*显示一个 16*16 点阵汉字:液*/
    //delay(20);
    disp_grap(2, 5, jing); /*显示一个 16*16 点阵汉字:晶*/
    //delay(20);
    disp_grap(2, 6, mo); /*显示一个 16*16 点阵汉字:模*/
    //delay(20);
    disp_grap(2, 7, kuai); /*显示一个 16*16 点阵汉字:块*/
    //delay(20);
    disp_grap(2, 8, chang); /*显示一个 16*16 点阵汉字:厂*/
    //delay(20);
}

void disp_numeric()
{
```

```
disp_char(1, 1, num0); /*在第 1 行第 1 列显示一个 8*16 点阵数字:0*/
disp_char(1, 2, num1); /*在第 1 行第 2 列显示一个 8*16 点阵数字:1*/
disp_char(1, 3, num2); /*在第 1 行第 3 列显示一个 8*16 点阵数字:2*/
disp_char(1, 4, num3); /*在第 1 行第 4 列显示一个 8*16 点阵数字:3*/
disp_char(1, 5, num4); /*在第 1 行第 5 列显示一个 8*16 点阵数字:4*/
disp_char(1, 6, num5); /*以下类同*/
disp_char(1, 7, num6);
disp_char(1, 8, num7);
disp_char(1, 9, num8);
disp_char(1, 10, num9);
disp_char(1, 11, num0);
disp_char(1, 12, num1);
disp_char(1, 13, num2);
disp_char(1, 14, num3);
disp_char(1, 15, num4);
disp_char(1, 16, num5);
disp_char(2, 1, num0); /*在第 2 行第 1 列显示一个 8*16 点阵数字:0*/
disp_char(2, 2, num1); /*在第 2 行第 2 列显示一个 8*16 点阵数字:1*/
disp_char(2, 3, num2); /*在第 2 行第 3 列显示一个 8*16 点阵数字:2*/
disp_char(2, 4, num3); /*在第 2 行第 4 列显示一个 8*16 点阵数字:3*/
disp_char(2, 5, num4); /*在第 2 行第 5 列显示一个 8*16 点阵数字:4*/
disp_char(2, 6, num5); /*以下类同*/
disp_char(2, 7, num6);
disp_char(2, 8, num7);
disp_char(2, 9, num8);
disp_char(2, 10, num9);
disp_char(2, 11, num0);
disp_char(2, 12, num1);
disp_char(2, 13, num2);
disp_char(2, 14, num3);
disp_char(2, 15, num4);
disp_char(2, 16, num5);
}
void disp_ascill()
{
    disp_char(2, 16, num5);
}

/*初始化*/
void initial_lcd()
{
    reset=0;          //Reset the chip when reset=0
    delay(20);
    reset=1;
    transfer_command(0xe2); /*软复位*/
}
```

```

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(0x22);    /*粗调对比度, 可设置范围 20~27*/
transfer_command(0x81);    /*微调对比度*/
transfer_command(0x19);    /*微调对比度的值, 可设置范围 0~63*/
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: 从第一行开始*/
transfer_command(0xaf);    /*开显示*/
}

```

```
//=====clear all dot martrics=====
```

```
void clear_screen()
```

```

{
unsigned char i, j;

    for(i=0;i<4;i++)
    {
        cs1=0;
        transfer_command(0xb0+i);
        transfer_command(0x10);
        transfer_command(0x00);
        for(j=0;j<132;j++)
        {
            transfer_data(0x00);
        }
    }
}

```

```
/*在指定行和列位置显示指定的汉字 (16*16 点阵的汉字) */
```

```
void disp_grap(int line,int column,char *dp)
```

```

{
    int i, j, k, col_l, col_h;
    for(i=0;i<2;i++)
    {
        cs1=0;
        transfer_command(0xb0+i+2*(line-1)); //set page address,
        k=column-1;
        k=k*0x10;
        col_h=k&0xf0;
        col_h=col_h>>4;
        col_l=k&0x0f;
    }
}

```

```
transfer_command(0x10+col_h);
transfer_command(0x00+col_l);
for(j=0;j<16;j++)
{
    transfer_data(*dp);
    dp++;
}
}
```

/*在指定行和列位置显示指定的字母、数字（8*16点阵的）*/

```
void disp_char(int line,int column,char *dp)
{
    int i,j,k,col_l,col_h;
    for(i=0;i<2;i++)
    {
        cs1=0;
        transfer_command(0xb0+i+2*(line-1)); //set page address,
        k=column-1;
        k=k*0x08;
        col_h=k&0xf0;
        col_h=col_h>>4;
        col_l=k&0x0f;
        transfer_command(0x10+col_h);
        transfer_command(0x00+col_l);
        for(j=0;j<8;j++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}
```

/*=====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(5);
    }
}
```



```
sclk=1;
delay1(5);
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;

        }
}

//=====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 Switch()
{
  repeat:
    if (P2&0x01) goto repeat;
    else delay(1);
    if (P2&0x01) goto repeat;
    else;
}
//-----

char code quan[]={
/*-- 文字: 全 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x00, 0x80, 0x40, 0x60, 0x50, 0x48, 0x44, 0xC3, 0x44, 0x48, 0x50, 0x70, 0x60, 0x20, 0x00, 0x00,
0x00, 0x40, 0x40, 0x44, 0x44, 0x44, 0x44, 0x7F, 0x44, 0x44, 0x44, 0x46, 0x44, 0x60, 0x40, 0x00
};

char code tao[]={
/*-- 文字: 套 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x40, 0x44, 0x24, 0x24, 0xF4, 0x5C, 0x57, 0x54, 0x54, 0x5C, 0x54, 0x24, 0x64, 0xC4, 0x44, 0x00,
0x04, 0x04, 0x44, 0x64, 0x57, 0x4D, 0x45, 0x65, 0x25, 0x25, 0x35, 0xE4, 0x44, 0x04, 0x04, 0x00
};

char code ye[]={
/*-- 文字: 液 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x10, 0x61, 0x06, 0xE0, 0x18, 0x84, 0xE4, 0x1C, 0x84, 0x65, 0xBE, 0x24, 0xA4, 0x64, 0x04, 0x00,
0x04, 0x04, 0xFF, 0x00, 0x01, 0x00, 0xFF, 0x41, 0x21, 0x12, 0x0C, 0x1B, 0x61, 0xC0, 0x40, 0x00
};

char code jing[]={
/*-- 文字: 晶 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x00, 0x00, 0x00, 0x00, 0x7E, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0x2A, 0x7E, 0x00, 0x00, 0x00, 0x00,
0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00, 0x00, 0x7F, 0x25, 0x25, 0x25, 0x25, 0x7F, 0x00
};

char code jie[]={
/*-- 文字: 解 --*/
/*-- 宋体 12; 此字体下对应的点阵为: 宽 x 高=16x16 --*/
0x08, 0xF4, 0x57, 0x54, 0xFC, 0x54, 0xF0, 0x42, 0xA2, 0x1E, 0x02, 0xA2, 0x62, 0x3E, 0x00, 0x00,
0x80, 0x7F, 0x02, 0x02, 0x5F, 0x82, 0x7F, 0x0A, 0x09, 0x09, 0x09, 0xFF, 0x09, 0x09, 0x09, 0x00
};

char code jue[]={
```



```
/*-- 文字： 决 --*/  
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/  
0x02, 0x04, 0xD8, 0x08, 0x00, 0x08, 0x08, 0x08, 0xFF, 0x08, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00,  
0x02, 0xFE, 0x01, 0x80, 0x41, 0x21, 0x11, 0x0D, 0x03, 0x05, 0x09, 0x11, 0x31, 0x61, 0x21, 0x00  
};  
char code fang[]={  
  
/*-- 文字： 方 --*/  
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/  
0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0xF9, 0x4A, 0x4C, 0x48, 0x48, 0xC8, 0x08, 0x08, 0x08, 0x00,  
0x40, 0x40, 0x20, 0x10, 0x0C, 0x03, 0x00, 0x00, 0x20, 0x40, 0x40, 0x3F, 0x00, 0x00, 0x00, 0x00  
};  
  
char code an[]={  
/*-- 文字： 案 --*/  
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/  
0x00, 0x20, 0x2C, 0x24, 0x64, 0x74, 0xAD, 0xA6, 0xE4, 0x34, 0x24, 0x24, 0x2C, 0x24, 0x00, 0x00,  
0x00, 0x24, 0x24, 0x25, 0x15, 0x15, 0x0D, 0xFE, 0x04, 0x0D, 0x17, 0x14, 0x24, 0x64, 0x24, 0x00  
};  
char code mo[]={  
/*-- 文字： 模 --*/  
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/  
0x10, 0xD0, 0xFF, 0x50, 0x90, 0x04, 0xF4, 0x54, 0x5F, 0x54, 0x54, 0x5F, 0xF4, 0x04, 0x00, 0x00,  
0x03, 0x00, 0xFF, 0x00, 0x00, 0x84, 0x85, 0x45, 0x35, 0x0F, 0x15, 0x25, 0x65, 0xC4, 0x44, 0x00  
};  
  
char code kuai[]={  
/*-- 文字： 块 --*/  
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/  
0x10, 0x10, 0xFF, 0x10, 0x10, 0x00, 0x08, 0x08, 0xFF, 0x08, 0x08, 0x08, 0xF8, 0x00, 0x00, 0x00,  
0x08, 0x18, 0x0F, 0x04, 0x85, 0x41, 0x31, 0x0D, 0x03, 0x05, 0x09, 0x11, 0x31, 0x61, 0x21, 0x00  
};
```



```
char code chang[]={
/*-- 文字： 厂 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=16x16 --*/
0x00,0x00,0xFE,0x02,0x02,0x02,0x02,0x02,0x02,0x02,0x02,0x02,0x02,0x02,0x02,0x00,
0x40,0x30,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
};
char code num0[]={
/*-- 文字： 0 --*/
/*-- 宋体 12； 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
};
char code num1[]={
/*-- 文字： 1 --*/
/*-- 宋体 12； 此字体下对应的点阵为： 宽 x 高=8x16 --*/
0x00,0x10,0x10,0xF8,0x00,0x00,0x00,0x00,0x00,0x20,0x20,0x3F,0x20,0x20,0x00,0x00
};
char code num2[]={
/*-- 文字： 2 --*/
/*-- 宋体 12； 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
};
char code num3[]={
/*-- 文字： 3 --*/
/*-- 宋体 12； 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
};
char code num4[]={
/*-- 文字： 4 --*/
/*-- 宋体 12； 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
};

char code num5[]={
/*-- 文字： 5 --*/
/*-- 宋体 12； 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
};

char code num6[]={
/*-- 文字： 6 --*/
/*-- 宋体 12； 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
};
char code num7[]={
/*-- 文字： 7 --*/
```



```
/*-- 宋体 12; 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  
};  
char code num8[]={  
/*-- 文字: 8  --*/  
/*-- 宋体 12; 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  
};  
  
char code num9[]={  
/*-- 文字: 9  --*/  
/*-- 宋体 12; 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  
};
```

