

JLX19232G-907-BN 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	时序特性	5~7
7	指令功能及硬件接口与编程案例	7~末页

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 JLX19232G-907 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。

JLX19232G-907 可以显示 192 列*32 行点阵单色图片，或显示 16*16 点阵的汉字 12 字*2 行或者 2 字 12 行，或显示 8*16 点阵的英文、数字、符号 24 个*2 行或者 4 个 12 行。或显示 5*8 点阵的英文、数字、符号 32 个*4 行或者 4 个 24 行。

2. JLX19232G-907 图像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。

2.2 IC 采用 UC1604c, 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 51mW (3.3V*15mA)；

2.4 显示内容：

(1) 192*32 点阵单色图片或者 32*192 点阵单色图片，或其它小于 192*32 点阵或 32*192 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 12 字*2 行或 2 字 12 行；

(3) 按照 8*16 点阵汉字来计算可显示 24 个*2 行或 4 个 12 行；

(4) 按照 5*8 点阵汉字来计算可显示 32 个*4 行或 4 个 24 行；

2.5 指令功能强；

2.6 接口简单方便：采用 4 线 SPI 串行接口。

2.7 工作温度宽：-20℃ - 70℃；

2.8 储存温度宽：-30℃ - 80℃；

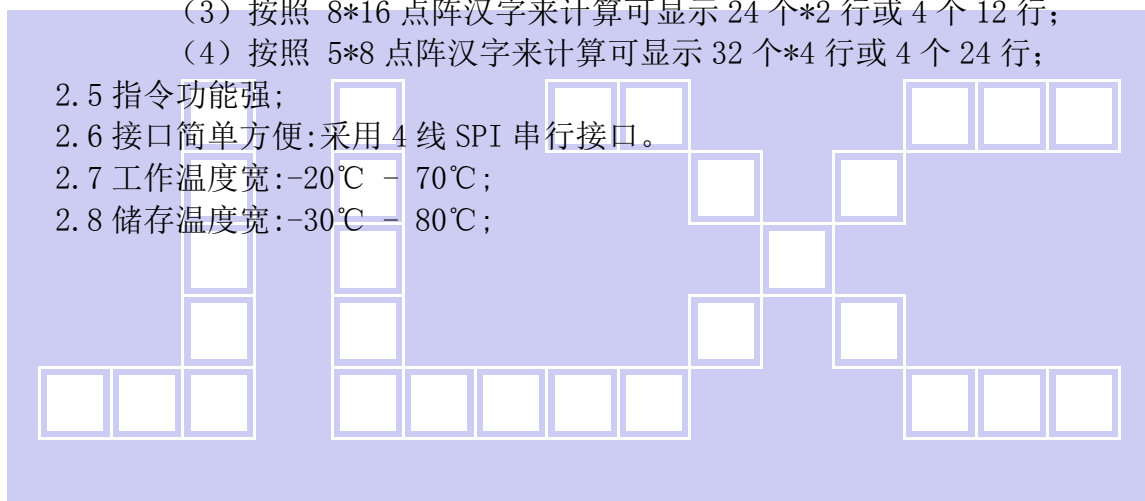


图 1. 外形尺寸

模块的接口引脚功能

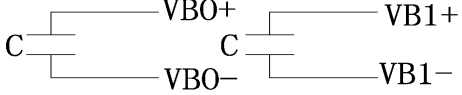
引 线 号	符 号	名 称	功 能
1	LEDK	背光电源负极	接 VSS
2	LEDA	背光电源正极	3.0V
3	VLCD	升压输出	VLCD $\xrightarrow{C}$ VSS
4	VB1+	升压电容	
5	VB1-	升压电容	
6	VBO+	升压电容	
7	VBO-	升压电容	
8	VSS	接地	0V
9	VDD	电路电源	3.3V
10	SDA	串行数据	数据传输
11	SCK	串行时钟	串行时钟
12	CD(即 RS)	寄存器选择信号	H: 数据寄存器 0: 指令寄存器
13	RST	复位	低电平复位, 复位完成后, 回到高电平, 液晶模块开始工作
14	CS0	片选	低电平片选

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 192×32 点阵, 192 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 工作电路框图:

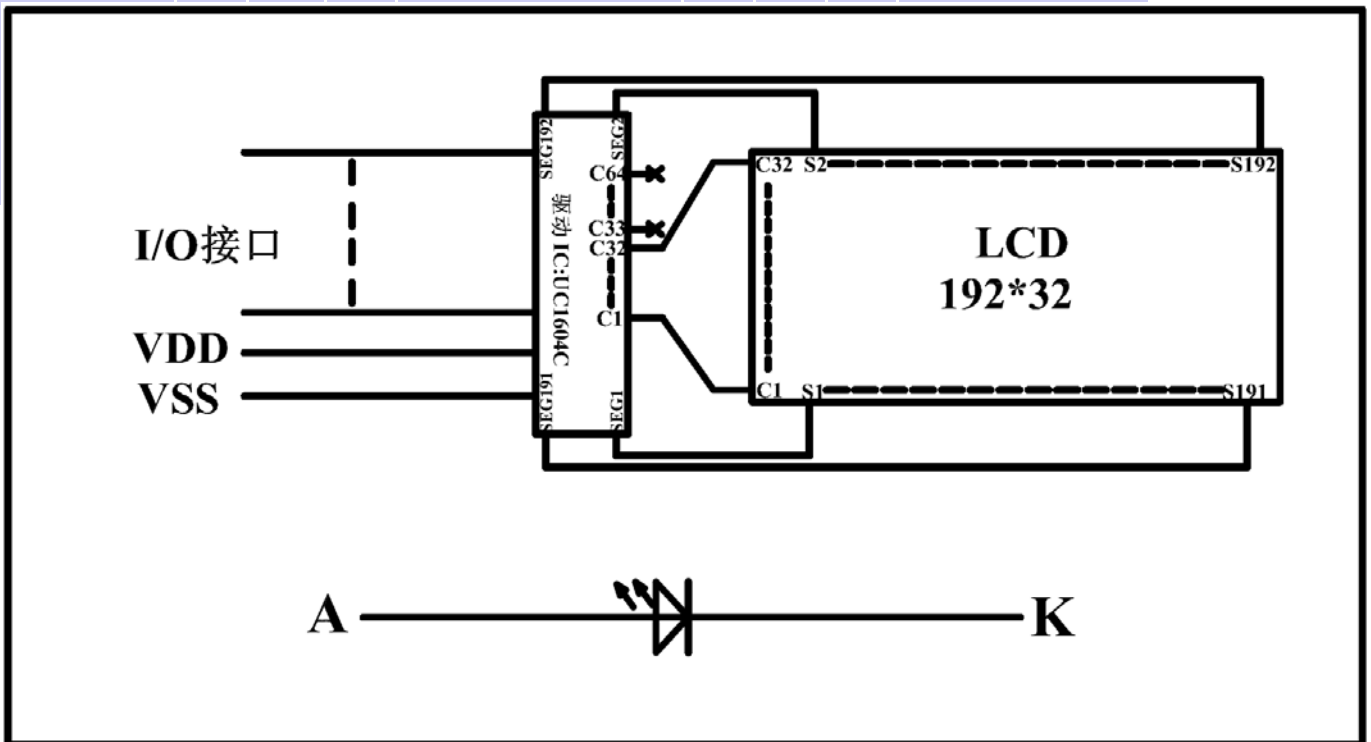


图 2: JLX19232G-907 图像点阵型液晶模块的电路框图

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

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

背光白色;

正常工作电流为: $8 \sim 20\text{mA}$ (LED 灯数共 1 颗);

工作电压: 3.0V ;

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	3.3	3.6	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	VIL	-	0	-	0.2VDD	V
输出高电平	VOH	$\text{IOH} = 0.2\text{mA}$	0.8VDD		VDD	V
输出低电平	VOO	$\text{IOO} = 1.2\text{mA}$	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 (共 2 颗 LED 灯并联)	8	15	20	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 UC1604c (Writing Data from CPU to UC1604c)

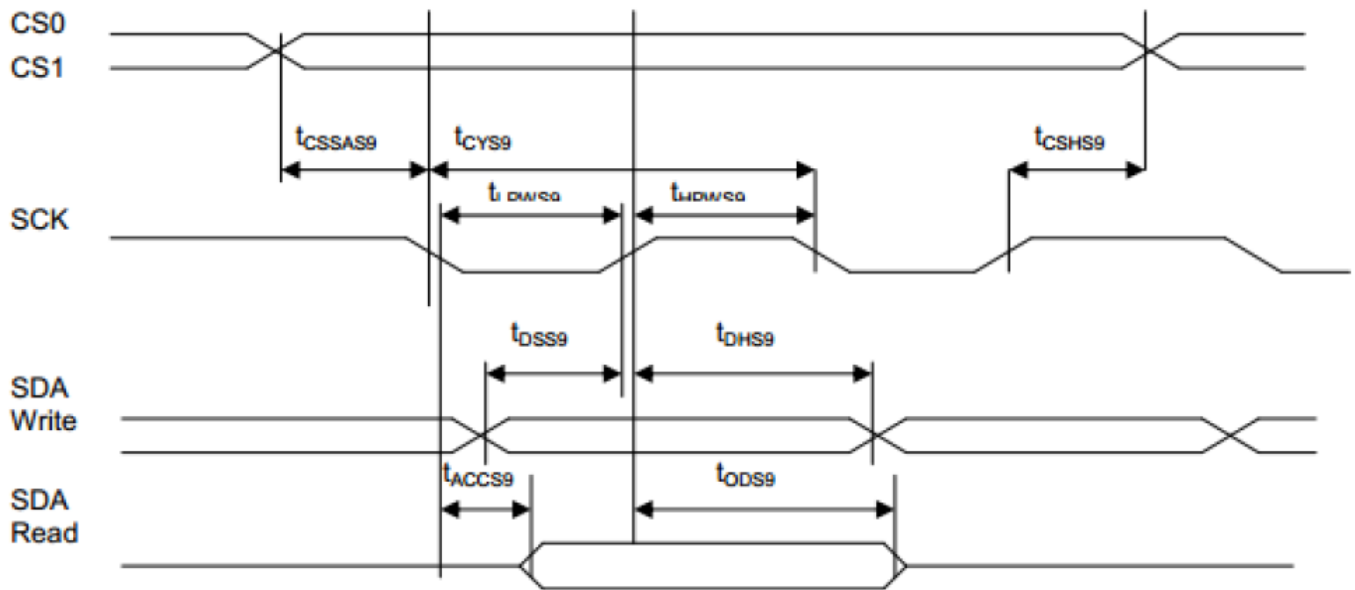


FIGURE 18: Serial Bus Timing Characteristics (for S9)

图 3. 从 CPU 写到 UC1604c (Writing Data from CPU to UC1604c)

6.2 串行接口：时序要求（AC 参数）：

写数据到 UC1604c 的时序要求：

Symbol	Signal	Description	Condition	Min.	Max.	Unit
(2.5V ≤ V _{DD} ≤ 3.6V, Ta= −30 to +85°C)						
(Read / Write)						
t _{CSSAS9}	CS1, CS0	Chip select setup time		5	–	nS
t _{CSSHS9}		Chip select hold time		5		
t _{CYS9}	SCK	System cycle time		190 / 70		
t _{LPWS9}		Low pulse width		80 / 20	–	nS
t _{HPWS9}		High pulse width		80 / 20		
t _{DSS9}	SDA	Data setup time		20	–	nS
t _{DHS9}	(Write)	Data hold time		10		
t _{ACC9}	SDA	Read access time	C _L = 100pF	–	80	nS
t _{OD9}	(Read)	Output disable time		–	30	
(1.7V ≤ V _{DD} < 2.5V, Ta= −30 to +85°C)						
(Read / Write)						
t _{CSSAS9}	CS1, CS0	Chip select setup time		10	–	nS
t _{CSSHS9}		Chip select hold time		10		
t _{CYS9}	SCK	System cycle time		230 / 110		
t _{LPWS9}		Low pulse width		100 / 40	–	nS
t _{HPWS9}		High pulse width		100 / 40		
t _{DSS9}	SDA	Data setup time		24	–	nS
t _{DHS9}	(Write)	Data hold time		15		
t _{ACC9}	SDA	Read access time	C _L = 100pF	–	100	nS
t _{OD9}	(Read)	Output disable time		–	60	

Note: tr (Rising time), tf (falling time) : ≤ 15nS

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

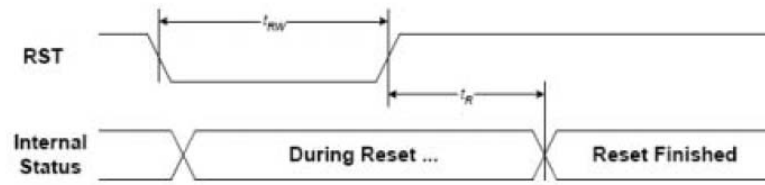


FIGURE 20: Reset Characteristics

(1.7V ≤ V_{DD} ≤ 3.6V, Ta = -30 to +85°C)

Symbol	Signal	Description	Condition	Min.	Max.	Unit
t _{RW}	RST	Reset low pulse width		3	—	μs
t _R	RST, Internal Status	Reset to Internal Status pulse delay		6	—	ms

7. 指令功能:

7.1 指令表

表 4.

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
(2) 显示初始行设置 (Display start line set)		0	0	1	显示初始行地址, 共 6 位					
(3) 页地址设置 (Page address set)		0	1	0	1	1	显示页地址, 共 4 位			
(4)	列地址高4位设置	0	0	0	0	1	列地址的高 4 位			
	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
(6) 写显示数据到液晶屏 (Display data write)		1	8 位显示数据							
(7) 读液晶屏的显示数据 (Display data read)		1	8 位显示数据							
(8) 显示列地址增减 (ADC select)			1	0	1	0	0	MY	MX	0
										1
(9) 显示正显/反显		0	1	0	1	0	0	1	1	0

(Display normal/reverse)									1	0xA6: 常规: 正显 0xA7: 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: 0xA4: 常规 0xA5: 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: 0xA2: BIAS=1/9 (常用) 0xA3: BIAS=1/7
(12) 读-改-写 (Read-modify-write)	0	1	1	1	0	0	0	0	0	0XE0: “读-改-写” 开始。 本液晶模块使用串行接口, 不具备此功能。 详情请参考IC资料
(13) 退出上述“读-改-写”指令(End)	0	1	1	1	0	1	1	1	0	0XEE: 上述“读-改-写”指令结束 本液晶模块使用串行接口, 不具备此功能。 详情请参考 IC 资料
(14) 软件复位 (Reset)	0	1	1	1	0	0	0	1	0	0XE2: 软件复位。

(15) 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: 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 0x2C, 0x2E, 0x2F 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 0x2F, 一次性打开三部分电路。
(16) 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): 可以理解为粗调对比度值。可设置范围为: 0x20~0x27, 数值越大对比度越浓, 越小越淡
(17)	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, 可以理解为微调对比度值, 此两个指令需紧接着使用。上面一条指令 0x81 是不改的, 下面一条指令可设置范围为: 0x00~0x3F, 数值越大对比度越浓, 越小越淡
	设置的电压值		0	0	6 位电压值数据, 0~63 共 64 级						
(18)静态图标显示: 开/关		0	1	0	1	0	1	1	0	0 1	静态图标的开关设置: 0xAC: 关, 0xAD: 开。 此指令在进入及退出睡眠模式时起作用
(19) 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
(20) 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
(21)空指令 (NOP)		0	1	1	1	0	0	0	1	1	空操作

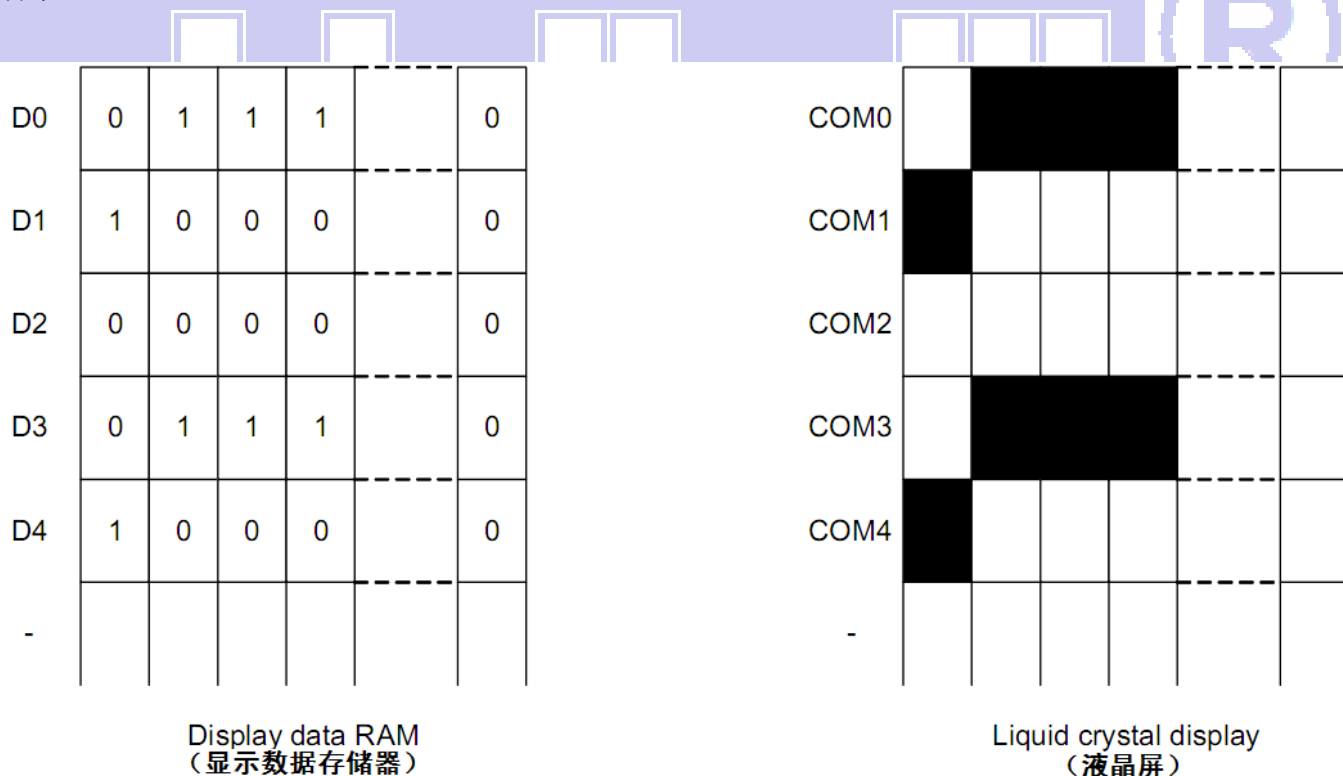
(22) 测试 (Test)	0	1	1	1	1	*	*	*	*	内部测试用, 千万别用!
----------------	---	---	---	---	---	---	---	---	---	--------------

温馨提示: 请详细参考 IC 资料 “UC1604c_V1.1.PDF”第 12~19 页的指令表及指令详解。

7.3 点阵与 DD RAM(显示数据存储器)地址的对应关系

请留意页的定义: PAGE, 与平时所讲的“页”并不是一个意思, 在此表示 8 个行就是一个“页”, 一个 192*32 点阵的屏分为 4 个“页”, 从第 0“页”到第 3“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:



下图摘自 UC1604c IC 资料, 可通过 “UC1604c_a0_1.1.PDF” 之第 36 页获取最佳效果。

10

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:

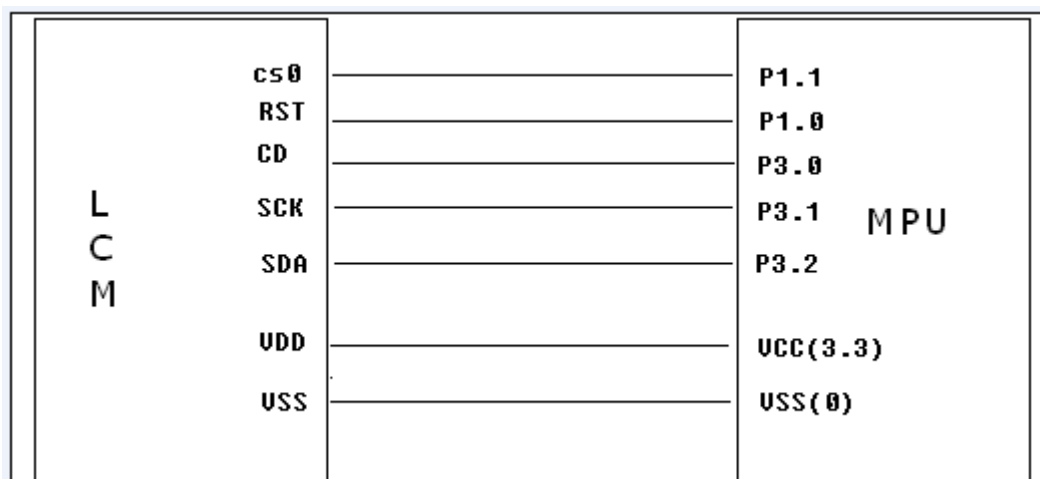
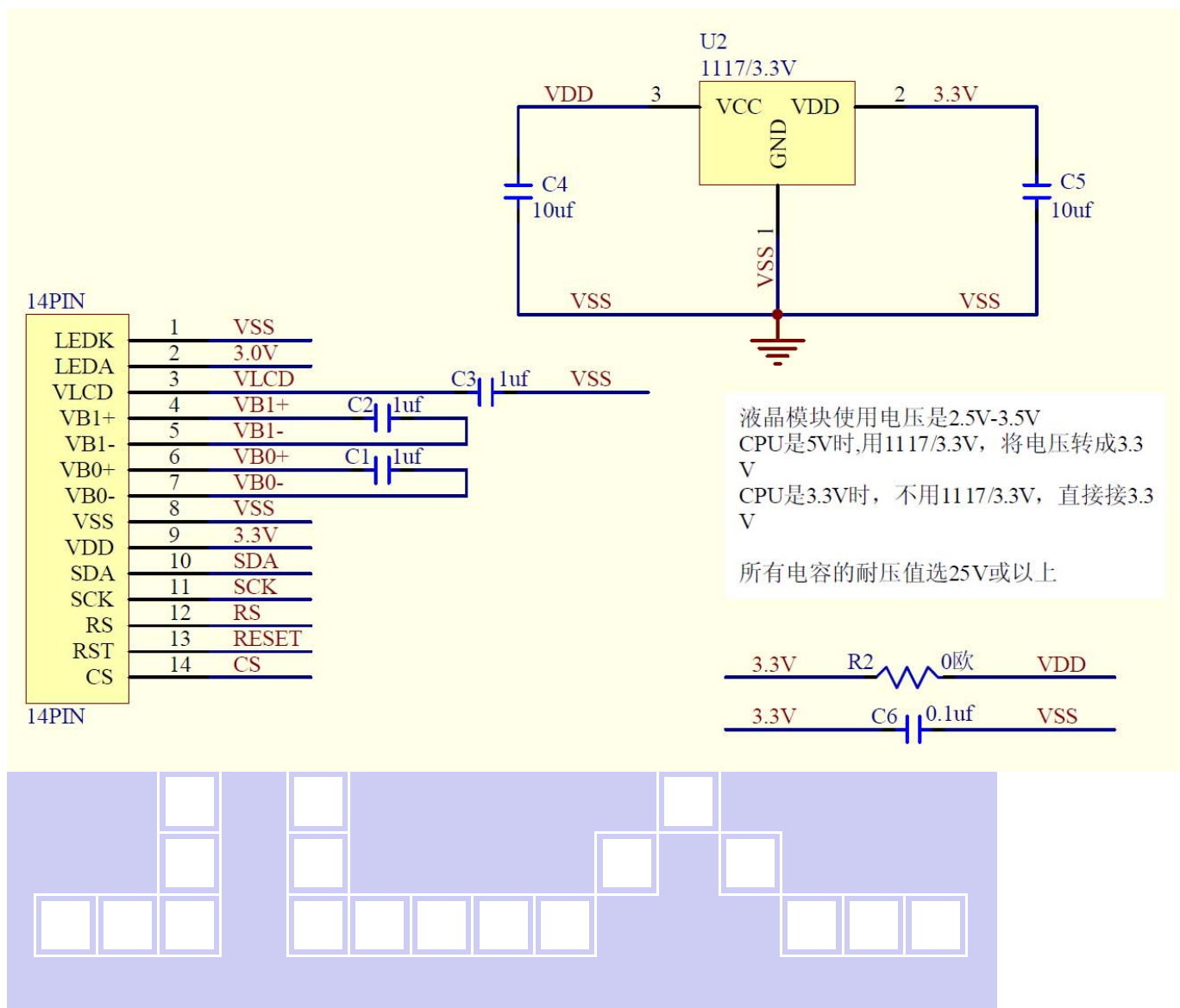


图 4. 串行接口



点亮液晶模块的编程步骤



// 液晶演示程序 JLX19232G-907, 串行接口!

// 驱动 IC 是: UC1604c

```
#include <reg52.h>
```

```
#include <intrins.h>
```

```
#include <Ctype.h>
```

```
#include <font.h> //此头文件购买后向销售索要
```

```
sbit cs1=P1^1; //LCD 片选引脚 (CS0)
```

```
sbit reset=P1^0; //LCD 复位引脚
```

```
sbit rs=P3^0; //LCD RS (A0) 引脚
```

```
sbit scl=P3^1; //LCD 串行时钟引脚 (SCK)
```

```
sbit sid=P3^2; //LCD 串行数据引脚 (SDA)
```

```
sbit key=P2^0; //我司测试主板, 按键引脚
```

```
void delay_us(int i);
```

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

```
//写指令到LCD 模块
```

```
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;
        sclk=1;
        data1=data1<<=1;
    }
    cs1=1;
```

```
//写数据到LCD 模块
```

```
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;
    }
    cs1=1;
}
```

```
//延时 1
```

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

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

}

//延时 2

void delay_us(int i)

{

int j, k;

for(j=0; j<i; j++)

for(k=0; k<10; k++);

}

void waitkey()

{

repeat:

if(key==1) goto repeat;

else delay(2800);

}

//LCD 模块初始化

void initial_lcd()

{

reset=0; //低电平复位

delay(800);

reset=1; //复位完毕

delay(800);

transfer_command(0xe2); //软复位

delay(500);

transfer_command(0x2f); //打开内部升压

delay(500);

transfer_command(0x81); //微调对比度

transfer_command(0x54); //微调对比度的值, 可设置范围 0x00~0xFF

transfer_command(0xe9); //1/7 偏压比 (bias)

transfer_command(0xc4); //行扫描顺序: 从上到下 0xc2

transfer_command(0xf1);

transfer_command(0x1f);

transfer_command(0xaf); //开显示

}

void lcd_address(uchar page, uchar column)

{

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

电话: 0755-29784961

Http://www.jlxlcd.cn

15

```
void clear_screen()
{
    unsigned char i, j;
    for(i=0; i<8; i++)
    {
        lcd_address(i, 0);
        for(j=0; j<192; j++)
        {
            transfer_data(0x00);
        }
    }
}
```

```
void display_graphic_192x32(uchar *dp)
```

```
{
    uchar i, j;
    for(i=0; i<4; i++)
    {
        lcd_address(i, 0);
        for(j=0; j<192; j++)
        {
            transfer_data(*dp);
            dp++;
        }
    }
}
```

//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;
    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++;
        }
    }
}
```

```
void display_string_8x16(uint page, uint column, uchar *text)
```

```
{
    uint i=0, j, k, n;
    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++;
}
}

```

//显示一串 5x8 点阵的字符串

//括号里的参数分别为 (页, 列, 是否反显, 数据指针)

```
void display_string_5x8(uint page, uint column, uchar reverse, uchar *text)
```

```

{
    uchar i=0, j, k, data1;
    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++)
            {
                if(reverse==1) data1=~ascii_table_5x8[j][k];
                else data1=ascii_table_5x8[j][k];
                transfer_data(data1);
            }
            if(reverse==1) transfer_data(0xff);
            else transfer_data(0x00);
            i++;
            column+=6;
        }
        else
            i++;
    }
}

```



```

}

void display_string_5x8_1(uint page,uint column,uchar *text)
{
    uint i=0,j,k;
    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++;
    }
}

//写入一组 16x16 点阵的汉字字符串 (字符串表格中需含有此字)
//括号里的参数: (页, 列, 汉字字符串)
void display_string_16x16(uchar page,uchar column ,uchar *text)
{
    uchar i,j,k;
    uint address;
    j=0;
    while(text[j]!='\0')
    {
        i=0;
        address=1;
        while(Chinese_text_16x16[i]> 0x7e)
        {
            if(Chinese_text_16x16[i] == text[j])
            {
                if(Chinese_text_16x16[i+1] == text[j+1])
                {
                    address=i*16;
                    break;
                }
            }
            i +=2;
        }
    }
}

```



```

if(column>191)
{
    column=0;
    page+=2;
}
if(address !=1)
{
    for(k=0;k<2;k++)
    {
        lcd_address(page+k, column);
        for(i=0;i<16;i++)
        {
            transfer_data(Chinese_code_16x16[address]);
            address++;
        }
    }
    j +=2;
}

```

```

else
{

```

```

    lcd_address(page, column);
    for(k=0;k<2;k++)
    {
        for(i=0;i<16;i++)
        {
            transfer_data(0x00);
        }
    }
    j++;
}

```

```

    column+=16;
}
}

```

//竖屏显示 32x32 点阵图像、汉字

//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;
    for(j=0;j<32;j++)
    {
        for (i=0;i<4;i++)
        {
            lcd_address(page+i, column+j);
            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++;
        }
    }
}

```

```

    }
}

void main(void)
{
    while(1)
    {
        initial_lcd();
        clear_screen();
        display_string_5x8(0, 0, 1, "MENU"); //显示 5x8 点阵的字符串, 括号里的参数分别为 (页, 列, 是否反
显, 数据指针)
        display_string_5x8(2, 0, 0, "Select>>>>");
        display_string_5x8(2, 66, 1, "1. Graphic");
        display_string_5x8(3, 66, 0, "2. Chinese");
        waitkey();

        clear_screen(); //clear all dots
        display_graphic_192x32(bmp1);
        waitkey();

        transfer_command(0xc2);
        clear_screen(); //clear all dots
        display_graphic_192x32(bmp1);
        waitkey();
        clear_screen();
        transfer_command(0xc0);
        display_graphic_32x32(0, 32*0, jing); //显示单个汉字"晶"
        display_graphic_32x32(0, 32*1, lian); //显示单个汉字"联"
        display_graphic_32x32(0, 32*2, xun); //显示单个汉字"讯"
        display_graphic_32x32(0, 32*3, ye); //显示单个汉字"液"
        display_graphic_32x32(0, 32*4, jing); //显示单个汉字"晶"
        display_graphic_32x32(0, 32*5, ping); //显示单个汉字"屏"
        waitkey();
        clear_screen();

        transfer_command(0xc6);
        display_graphic_32x32(0, 32*0, jing); //显示单个汉字"晶"
        display_graphic_32x32(0, 32*1, lian); //显示单个汉字"联"
        display_graphic_32x32(0, 32*2, xun); //显示单个汉字"讯"
        display_graphic_32x32(0, 32*3, ye); //显示单个汉字"液"
        display_graphic_32x32(0, 32*4, jing); //显示单个汉字"晶"
        display_graphic_32x32(0, 32*5, ping); //显示单个汉字"屏"
        waitkey();
        transfer_command(0xc4);
        clear_screen(); //clear all dots
        display_string_16x16(1, 0, "欢迎使用晶联讯液晶屏模块");
        waitkey();
        clear_screen(); //clear all dots

```

```
display_string_8x16(0,0,"(<\\"0123456abt~!@##$%^\">)");//在第1页,第1列显示字符串
display_string_8x16(2,0,"{[(<\\" '&*|\\@#_-= ' \")]}");//在第*页,第*列显示字符串
waitkey();
clear_screen();
display_string_5x8_1(0,0,"[!#$%&'()*+,-./0123456789:;<=>?]"");
display_string_5x8_1(1,0,"[ABCDEFGHIJKLMNOPQRSTUVWXYZabcd]"");
display_string_5x8_1(2,0,"(abcdefghijklmnopqrstuvwxyzabcd)"");
display_string_5x8_1(3,0,"{[(<\\" '&*|\\@abcde012#_-= ' \")]}");//
waitkey();
}
}
```

-END-

