

JLX1602G-391-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	5
5	技术参数	5~6
6	时序特性	6~9
7	指令功能及硬件接口与编程案例	10~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 JLX1602G-391 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。

JLX1602G-391 可以显示 2 行，每行 16 个英文、数字、符号，并可以自编字符（每显示一个界面最多可以达到 8 个 5*8 点阵自编字符）。

2. JLX1602G-391 图像型点阵液晶模块的特性

2.1 重量轻;

2.2 体积小;

2.3 结构牢: 背光带有双面胶, 针脚式连接, PCB 板;

2.4 IC 采用 ST7032I, 功能强大, 稳定性好;

2.5 功耗低: 1~100mW (关掉背光: [0.3mA@3.3V](#), 打开背光不大于 100mW);

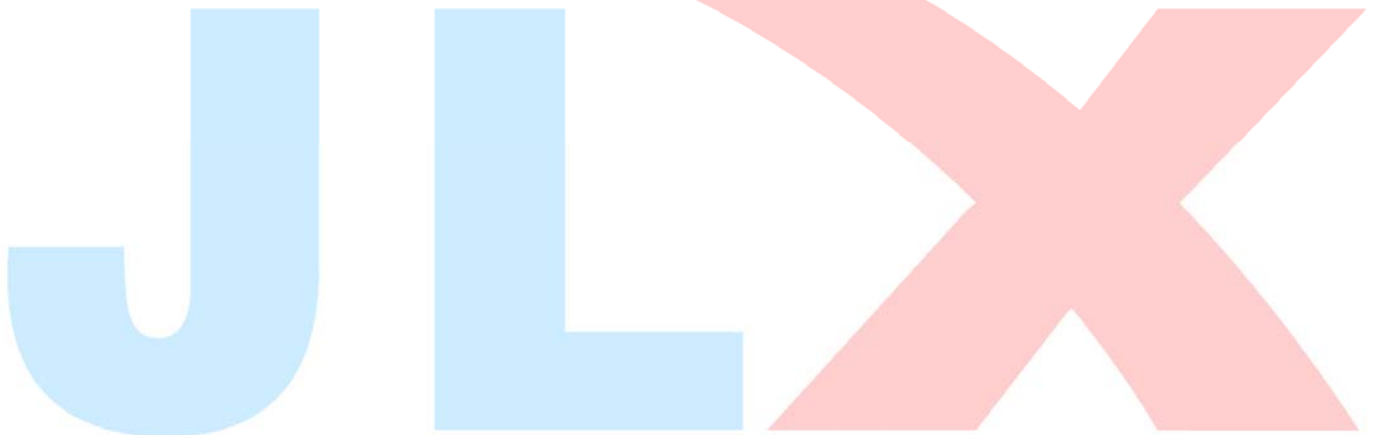
2.6 显示内容: 256 种字符 (5×8 点字型);

可自编 8 种 (5×8) 种字符, (注每显示一个界面最多可以达到 8 个自编字符, 但更换显示界面后可再编);

2.7 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;

2.8 接口简单方便: IIC 串行接口;

2.9 工作温度宽: -20℃ - 70℃。



3. 外形尺寸及接口引脚功能

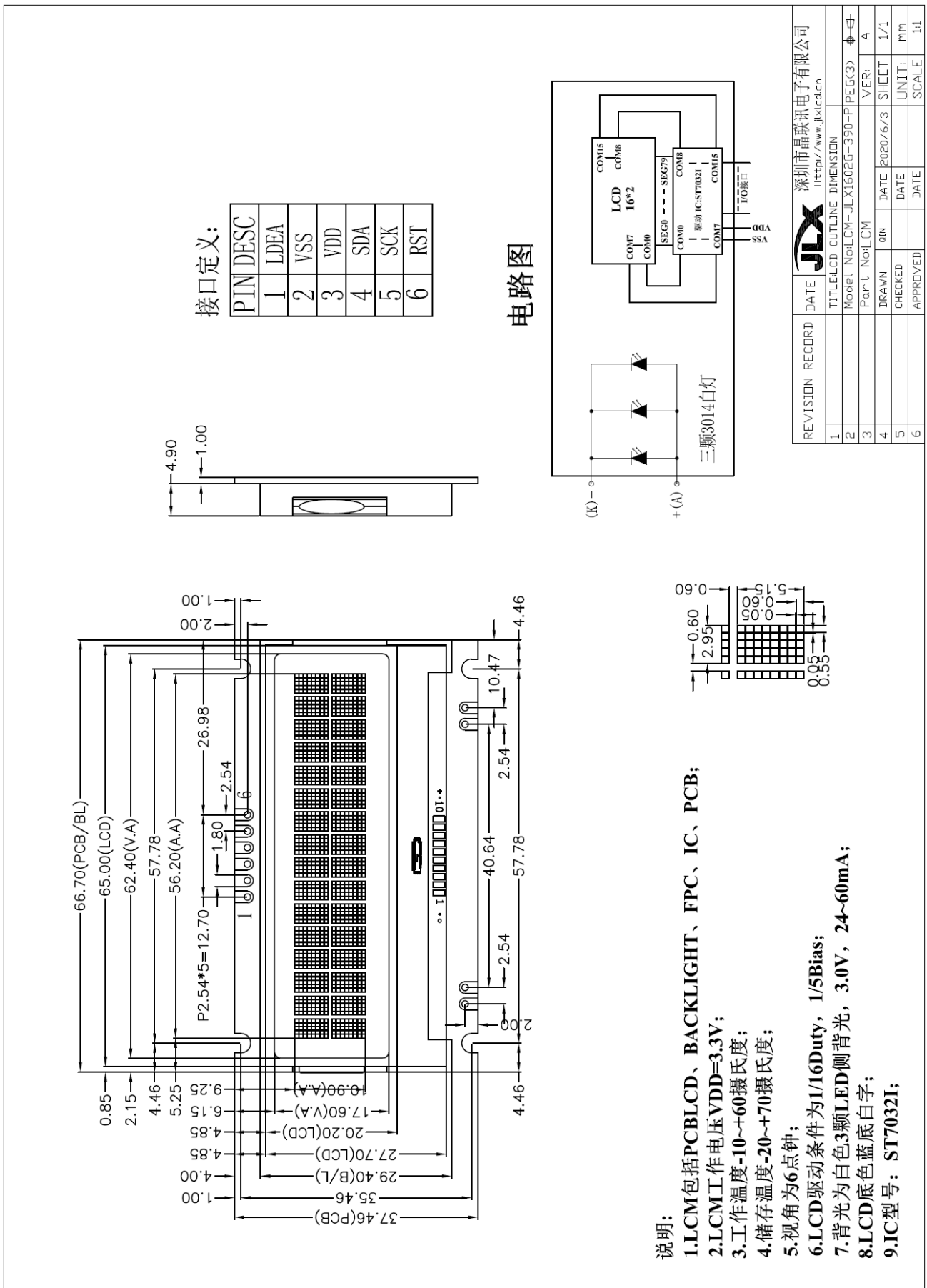


图 1. 外形尺寸

模块的接口引脚功能

引线号	符号	名称	功能
1	LEDA	背光电源	背光电源正极, 同 VDD 电压 (5V 或 3.3V)
2	VSS	接地	0V`
3	VDD	电路电源	5V 或 3.3V
4	SDA	I/O	串行数据
5	SCK	I/O	串行时钟
6	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

表 1: 模块的接口引脚功能

4. 基本原理

4.1 液晶屏 (LCD)

在液晶板上排列着若干 5×8 点阵的字符显示位, 每个显示位可显示 1 个字符, 本产品每行 16 个显示位, 共两行。

4.2 内部电路框图:

电路图

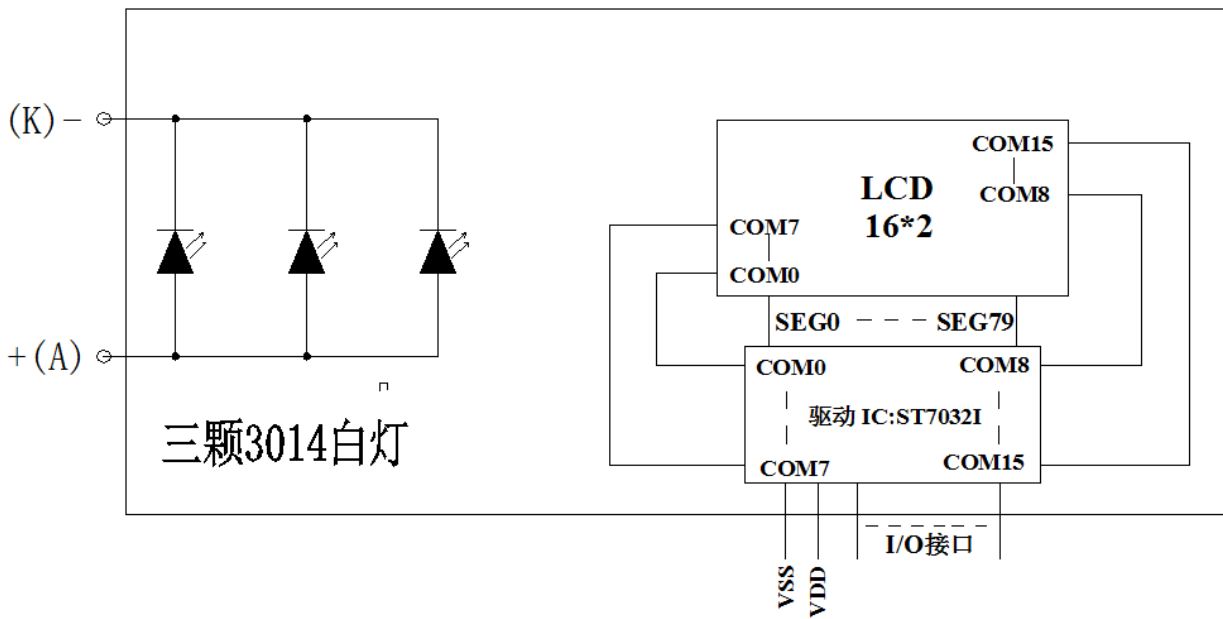


图 2: 内部电路框图

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:

背光 LED 灯颜色: 白色。

正常工作电流为: 24~45mA (LED 灯数共 3 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	
电路电源	VDD - VSS	-0.3	3.3	3.6	V
LCD 驱动电压	V0-XV0	-0.3	5.1	7	V
LCD 驱动电压	VG	-0.3		4	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		2.7	3.3	4.5	V
背光工作电压	VLED		2.9	3.0	3.1	V
输入高电平	VIHC	-	0.8xVDD	-	VDD	V
输入低电平	VILC	-	VSS	-	0.2xVDD	V
输出高电平	VOHC	IOH = 0.2mA	0.8xVDD	-	VDD	V
输出低电平	VOHC	I00 = 1.2mA	VSS	-	0.2xVDD	V
模块工作电流	I _{DD}	VDD = 3.0V	-		0.3	mA
背光工作电流	I _{LED}	VLED=3.0V	24	30	45	mA

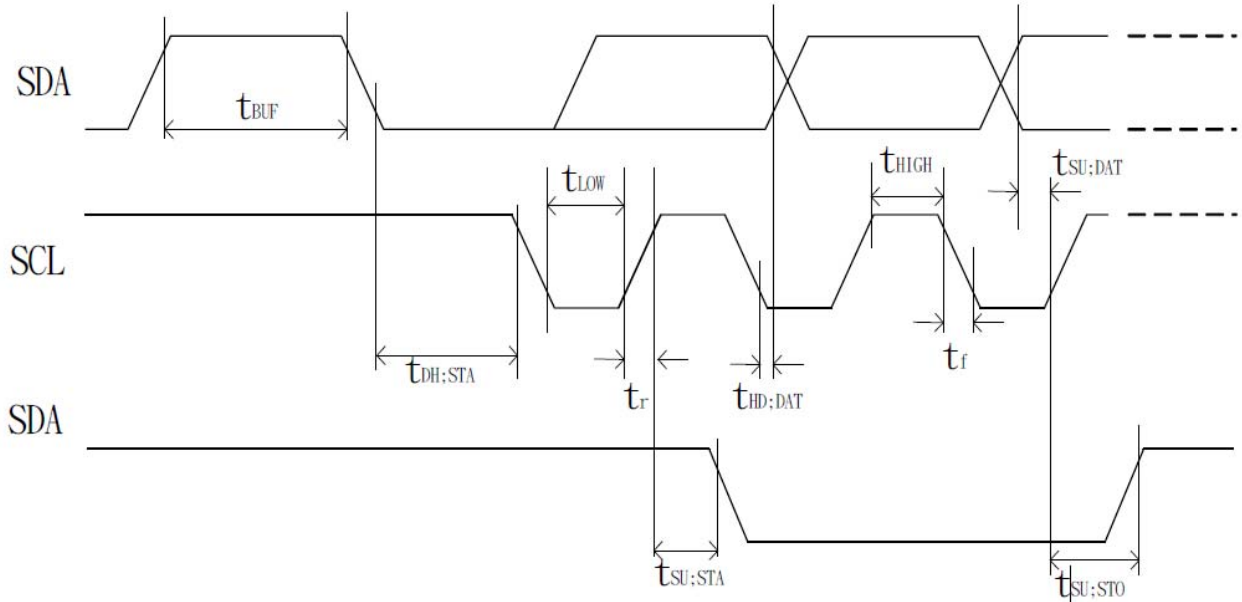
表 3: 直流 (DC) 参数

6. 读写时序特性

6.1 IIC 行接口:

从 CPU 写到 ST7032I (Writing Data from CPU to ST7032I)

● I2C interface



项目	符号	名称	极限值			单位
			MIN	TYPE	MAX	
SCL时钟频率	CSL	FSCLK	DC		400	kUZ
SCL时钟的低周期	CSL	TLOW	1.3		--	us
SCL时钟周期	CSL	THIGH	0.6		--	us
数据保持时间	SDA	TSU;Data	180		--	ns
数据建立时间	SDA	THD;Data	0		0.9	us
SCL, SDA 的上升时间	SCL	TR	20+0.1Cb		300	ns
SCL, SDA 下降时间	SCL	TF	20+0.1Cb		300	ns
每个总线为代表的电容性负载		Cb	--		400	pF
一个重复起始条件设置时间	SDA	TSU;SUA	0.6		--	us
启动条件的保持时间	SDA	THD;STA	0.6		--	us
为停止条件建立时间		TSU;STO	0.6		--	us
开始和停止条件之间的总线空闲时间	SCL	TBUF	1.3			us

6.2 复位

Hardware Reset (RSTB) Hardware Reset Timing

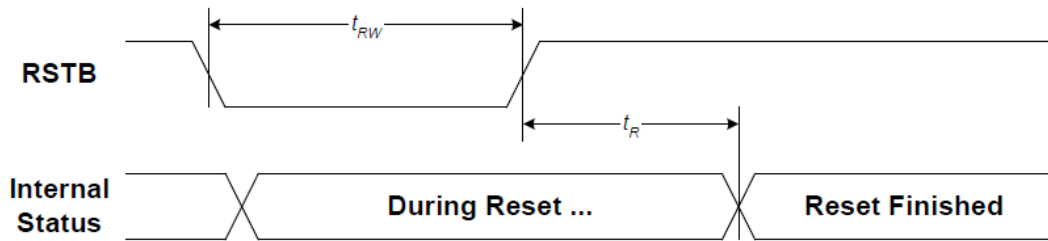
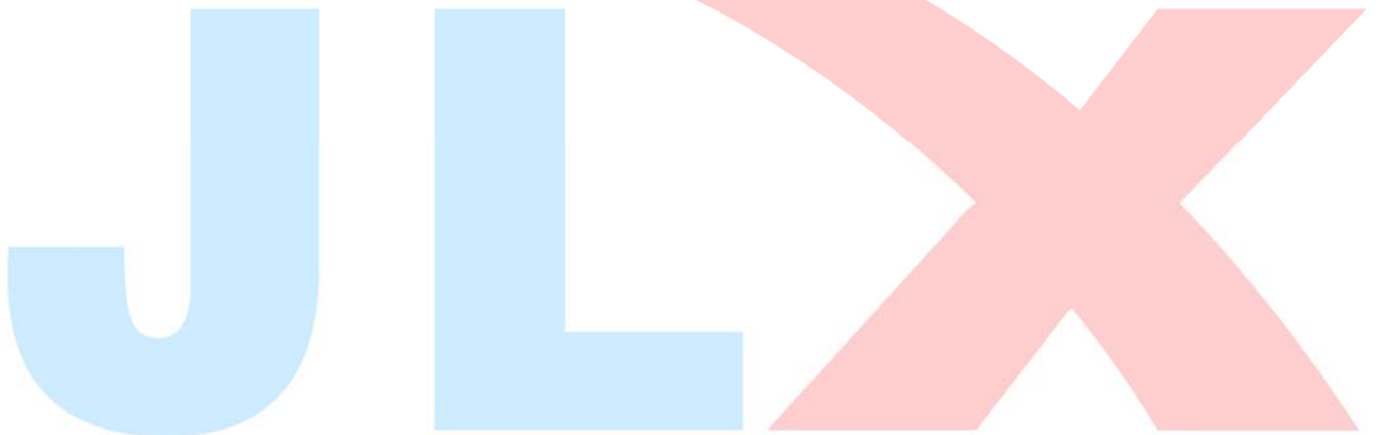


图 7: 电源启动后复位的时序

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	
复位保持低电平的时间	t_{RW}	引脚: RSTB	2.0	-	-	us
复位延时	t_R	引脚: RSTB	-	-	2.0	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
	R S	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
(1)清除显示 (Clear Display)	0	0	0	0	0	0	0	0	0	1	
(2)返回 (Return Home)	0	0	0	0	0	0	0	0	1	x	如果字符正在移动,此指令将 DDRAM 的地址设为“00H”,并且光标和屏幕回到原点。但 DDRAM 里面的内容不变。
(3)进入模式设置 (Entry Mode Set)	0	0	0	0	0	0	0	1	I/D	S	设置光标移动方向和控制显示移位。这些操作在数据的读写过程中生效。 S=1:移动;S=0:不移动 I/D=1:往右移动;I/D=0:往左移动;
(4)显示开关 (Display ON/OFF)	0	0	0	0	0	0	1	D	C	B	D=1:开显示;D=0:关显示。 C=1:打开光标;C=0:关掉光标。 B=1:开光标闪烁;B=0:关光标闪烁。
(5)功能设置 (Function Set)	0	0	0	0	1	DL	N	DH	0	IS	DL:并行接口时,选择4位/8位接口(DL="0":4位,DL="1":8位) N:选择显示行数(N="0":1行,N="1":2行) DH:双倍高度字体,DH=1:开,DH=0:关) IS:指令表选择
(6)设置 DDRAM 地址 (Set DDRAM address)	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	设置 DDRAM 地址
(6)读状态	0	1	0	AC6	AC5	AC4	AC3	AC2	AC1	AC0	读取地址计数器的内容
(7)写数据到RAM (Write data to RAM)	1	0	D7	D6	D5	D4	D3	D2	D1	D0	写数据到内部 RAM (DDRAM/CGRAM/ICONRAM)
(8)从RAM读取数据 (Write data to RAM)	1	1	D7	D6	D5	D4	D3	D2	D1	D0	从内部 RAM 中读取数据 (DDRAM/CGRAM/ICONRAM 的)

指令表 0 (当“IS”=0时)

(1)光标或显示移动 (Cursor or Display Shift)	0	0	0	0	0	1	S/C	R/L	X	X	S/C 和 R/L: 设置光标移动和显示移动及移动方向。但不改变 DDRAM 的数据。 S/C:=1:移动字符;S/C:=0:移动光标。 R/L=1:向右移动;R/L=0:向左移动。
(2)设置CGRAM (Set CGRAM)	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	设置地址计数器里的 CGRAM 地址

指令表 1 (当“IS”=1 时)

(1)内部晶振频率/Bias设置 (Internal OSC frequency/Bias Selection)	0	0	0	0	0	1	BS	F2	F1	F0	BS="1" : 1/4 偏压 BS="0" : 1/5 偏压 F[2:0]:设置内部振荡频率
(2)设置图标地址 (Set ICON address)	0	0	0	1	0	0	AC3	AC2	AC1	AC0	设置地址计数器里的图标地址
(3)电源/ICON控制/对比度设置 (Power/ICON control/Contrast set)	0	0	0	1	0	1	Ion	Bon	C5	C4	Ion: ICON 显示开/关 Bon: 升压电路开/关 C5,C4: 对比度设置
(4)电源控制 (Follower control)	0	0	0	1	1	0	Fon	0	0	0b	Fon: 内部跟随器电路开/关
(5)对比度设置 (Contrast set)	0	0	0	1	1	1	C3	C2	C1	C0	对比度设置

温馨提示: 请详细参考 IC 资料”ST7032I.PDF”的第 20-26 页。

7.3 字符库及对应关系

7.3.1 显示位与 DD RAM 地址的对应关系

表 9.

显示位序号	DD RAM 地址
DD RAM 地址 (HEX)	第一行 00 01 02 03 04 0F
	第二行 40 41 42 43 44 4F

7.3.2 标准字符库

下表所示的是字符库的内容, 字符码和字符的对应关系。

The available character fonts in CGROM are shown below:

ROM Code	Character generator ROM Size	Support Character
1A	256	English / European / Japanese

ST7032S-1A

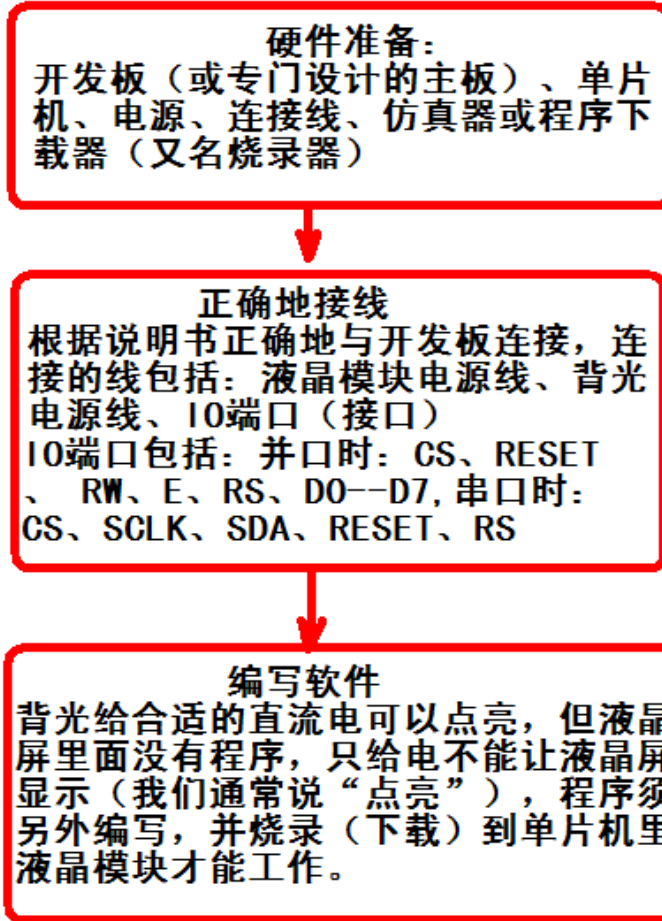
b7-b4 b3-b0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	!	@	#	\$	%	&	'	()	*	+	,	-	.	/	:
0001	;	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J
0010	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0011	[\]	^	_	`	a	b	c	d	e	f	g	h	i	j
0100	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0101	{		}	~												
0110																
0111																
1000																
1001																
1010																
1011																
1100																
1101																
1110																
1111																

表 10. ST7032I-1A 字库表

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

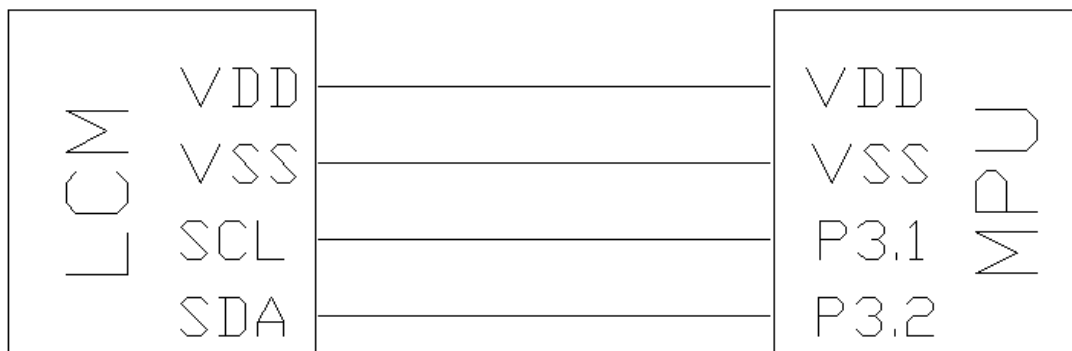
点亮液晶模块的步骤



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. IIC 接口



7.51、程序

IIC 程序:

```
/*=====*/
/* ST7032i 测试程序                               */
/* JLX1602G-390 测试程序                           */
/* 晶联讯电子: 公司网址: http://www.jlxlcd.cn;  阿里巴巴网址: http://www.jlxlcd.com.cn */
/*=====*/
#include <reg51.H>
sbit  reset=P3^0;
sbit   scl=P3^1;
sbit   sda=P3^2;
sbit   key=P2^0;

#define uchar unsigned char
#define uint unsigned int

/*=====*/
char code CGRAM_code[]={
0x1f,0x1f,0x1f,0x1f,0x1f,0x1f,0x1f,0x1f,
0x1f,0x11,0x11,0x11,0x11,0x11,0x11,0x1f,
0x1f,0x00,0x1f,0x00,0x1f,0x00,0x1f,0x00,
0x00,0x1f,0x00,0x1f,0x00,0x1f,0x00,0x1f,
0x0a,0x0a,0x0a,0x0a,0x0a,0x0a,0x0a,0x0a,
0x15,0x15,0x15,0x15,0x15,0x15,0x15,0x15,
0x0f,0x09,0x0f,0x09,0x0f,0x09,0x11,0x00,
0x08,0x0f,0x12,0x0f,0x0a,0x1f,0x02,0x02,
};
char code CGRAM[]={
{0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07};
char code japanese1[]={
{0xb0,0xb1,0xb2,0xb3,0xb4,0xb5,};
char code japanese2[]={
{0xb6,0xb7,0xb8,0xb9,0xba,0xbb,};

/*长一点的延时*/
void delay(int i)
{
    int j,k;
    for(j=0;j<i;j++)
        for(k=0;k<110;k++);
}

/*短一点的延时*/
void delay_us(int i)
{
    int j,k;
    for(j=0;j<i;j++)
        for(k=0;k<1;k++);
}

/*等待一个按键 (P2.0 口与 GND 之间) */
void waitkey()
{
    repeat:
```

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

```
void transfer(uchar data1)
```

```
{
    int i;
    for(i=0;i<8;i++)
    {
        scl=0;
        if(data1&0x80) sda=1;
        else sda=0;
        scl=1;
        delay_us(15);
        scl=0;
        data1=data1<<1;
    }
    sda=0;
    scl=1;
    delay_us(15);
    scl=0;
}
```

```
void start_flag()
```

```
{
    scl=1;    /*START FLAG*/
    sda=1;    /*START FLAG*/
    sda=0;    /*START FLAG*/
}
```

```
void stop_flag()
```

```
{
    scl=1;    /*STOP FLAG*/
    sda=0;    /*STOP FLAG*/
    sda=1;    /*STOP FLAG*/
}
```

```
void transfer_command(uchar com)
```

```
{
    start_flag();
    transfer(0x7c);
    transfer(0x80);
    transfer(com);
    stop_flag();
}
```

```
//写数据到 OLED 显示模块
```

```
void transfer_data(uchar dat)
```

```
{
    start_flag();
    transfer(0x7c);
    transfer(0xC0);
    transfer(dat);
    stop_flag();
}
```

```
}
```

```
/*初始化 LCD MODULE*/
```

```
void initial_lcd()
```

```
{
```

```
    transfer_command(0x38);    /*function select*/
```

```
    transfer_command(0x01);    /*clear screen*/
```

```
    delay(5);
```

```
    transfer_command(0x06);    /*setdisplay mode*/
```

```
    delay(5);
```

```
    transfer_command(0x0c);    /*turn on display*/
```

```
    delay(5);
```

```
    transfer_command(0x39);    /*extension instruction*/
```

```
    transfer_command(0x1c);    //
```

```
    delay(5);
```

```
    transfer_command(0x69);    //
```

```
    delay(5);
```

```
    transfer_command(0x57);    //粗调对比度, 范围 0x54-0x57
```

```
    delay(5);
```

```
    transfer_command(0x7a);    //微调对比度, 范围 0x70-0x7f
```

```
    delay(5);
```

```
}
```

```
void write_CGRAM()
```

```
{
```

```
    int i;
```

```
    transfer_command(0x38);    /*extension instruction*/
```

```
    transfer_command(0x40);    //set position
```

```
    for(i=0;i<64;i++)
```

```
    {
```

```
        transfer_data(CGRAM_code[i]);
```

```
    }
```

```
}
```

```
/*在指定行和列位置显示指定的字母、数字 (5*7 点阵的)*/
```

```
void disp_char(int line, int column, char code *dp)
```

```
{
```

```
    int i;
```

```
    transfer_command(0x80+(line-1)*0x40+(column-1));    //set position
```

```
    for(i=0;i<16;i++)
```

```
    {
```

```
        transfer_data(*dp);
```

```
        dp++;
```

```
    }
```

```
}
```

```
/*在指定行和列位置显示指定的字母、数字 (5*7 点阵的)*/
```

```
void disp_CGRAM()
```

```
{
```

```
int i, j;

for(j=0; j<8; j++)
{
    transfer_command(0x80); //set position
    for(i=0; i<16; i++)
    {
        transfer_data(CGRAM[j]);
    }
    transfer_command(0xc0); //set position
    for(i=0; i<16; i++)
    {
        transfer_data(CGRAM[j]);
    }
    waitkey();
}

}

/*主程序*/
void main(void)
{
    reset =0;
    delay(20);
    reset =1;
    delay(100);
    initial_lcd();
    write_CGRAM();
    while(1)
    {
        disp_char(1,1,"**JLX1602G-391**"); /*在第1行, 第1列, 显示字符.... */
        disp_char(2,1,"*Character COG**"); /*在第2行, 第1列, 显示字符.... */
        waitkey();
        disp_char(1,1,japanese1); /*在第2行, 第1列, 显示字符.... */
        disp_char(2,1,japanese2); /*在第2行, 第1列, 显示字符.... */
        waitkey();
        disp_char(1,1,"jlxlcd 1602g-391"); /*在第1行, 第1列, 显示字符.... */
        disp_char(2,1,"*2015/08/17 cog**"); /*在第2行, 第1列, 显示字符.... */
        waitkey();
        disp_char(1,1,"*0123456789 123*"); /*在第1行, 第1列, 显示字符.... */
        disp_char(2,1,"*Character COg**"); /*在第2行, 第1列, 显示字符.... */
        waitkey();
        disp_char(1,1,"*xxxxxxxxxxxxxx*"); /*在第1行, 第1列, 显示字符.... */
        disp_char(2,1,"*cccccccccccccc*"); /*在第2行, 第1列, 显示字符.... */
        waitkey();
        disp_CGRAM();
        waitkey();
    }
}
```