

JLX1602G-390-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	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 1602G-390 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。

1602G-390 最大显示 RAM 的大小可以在 1 行显示或 2 行显示 40 个字符或者 80 个字符。一个单一的 ST7032 可以显示长达 16 个字符的一行或两行 16 个字符的行。

2. 1602G-390 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。

2.4 IC 采用 ST7032, 功能强大，稳定性好

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

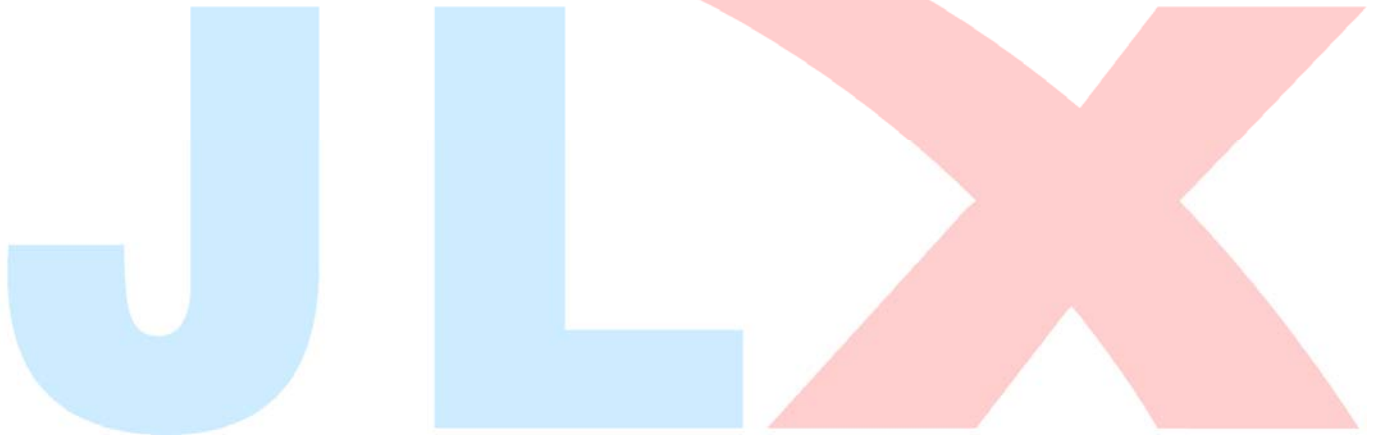
2.5 显示内容:256 种字符(5×8 点字型)；

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）；

2.5 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：采用 IIC 接口。

2.7 工作温度宽:-10℃ - 60℃；



3. 外形尺寸及接口引脚功能

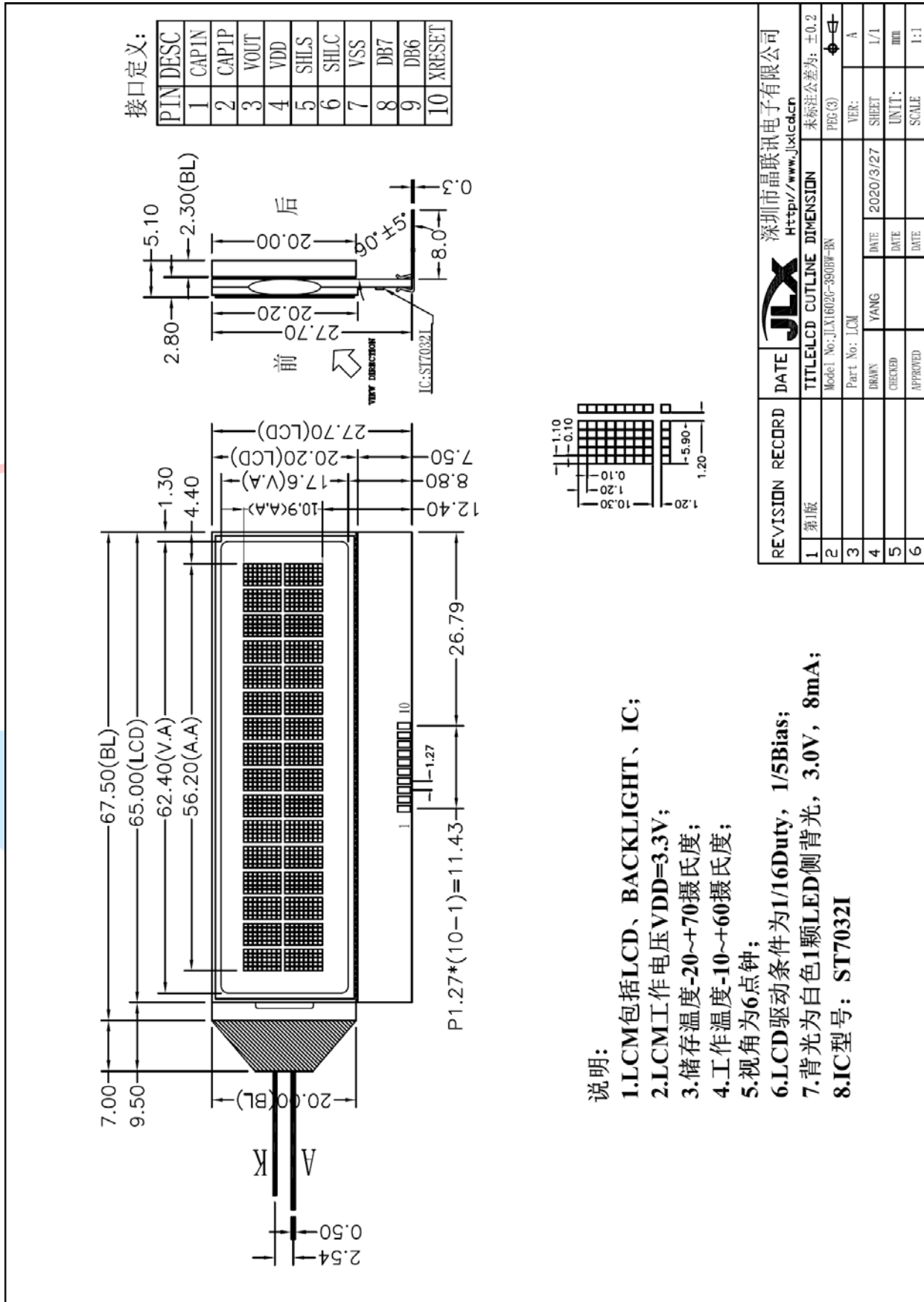


图 1. 外形尺寸

模块的接口引脚功能

引线号	符号	名称	功能
1	CAP1N	偏置电压	两个引脚之间接 1uF 电容
2	CAP1P	偏置电压	
3	VOUT	偏置电压	与 VSS 之间串 1uF 电容
4	VDD	供电电源正极	供电电源正极
5	SHLS	指令	SHLS=1 (这里接高电平, 可以根据需要选择)
6	SHLC	指令	SHLC=1 (这里接高电平, 可以根据需要选择)
7	VSS	接地	0V
8	D6 (SCLK)	I/O	串行时钟 (SCLK)
9	D7 (SDA)	I/O	串行数据 (SDA)
10	XRESET	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

■ LCD and ST7032 Connection

SHLC/SHLS ITO option pin can select at different direction for LCD panel

- Com normal direction/Seg normal direction



2Line x 16 Characters, SHLC=1, SHLS=1

- Com normal direction/Seg reverse direction



2Line x 16 Characters, SHLC=1, SHLS=0

- Com reverse direction/Seg normal direction



2Line x 16 Characters, SHLC=0, SHLS=1

- Com reverse direction/Seg reverse direction



2Line x 16 Characters, SHLC=0, SHLS=0

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:

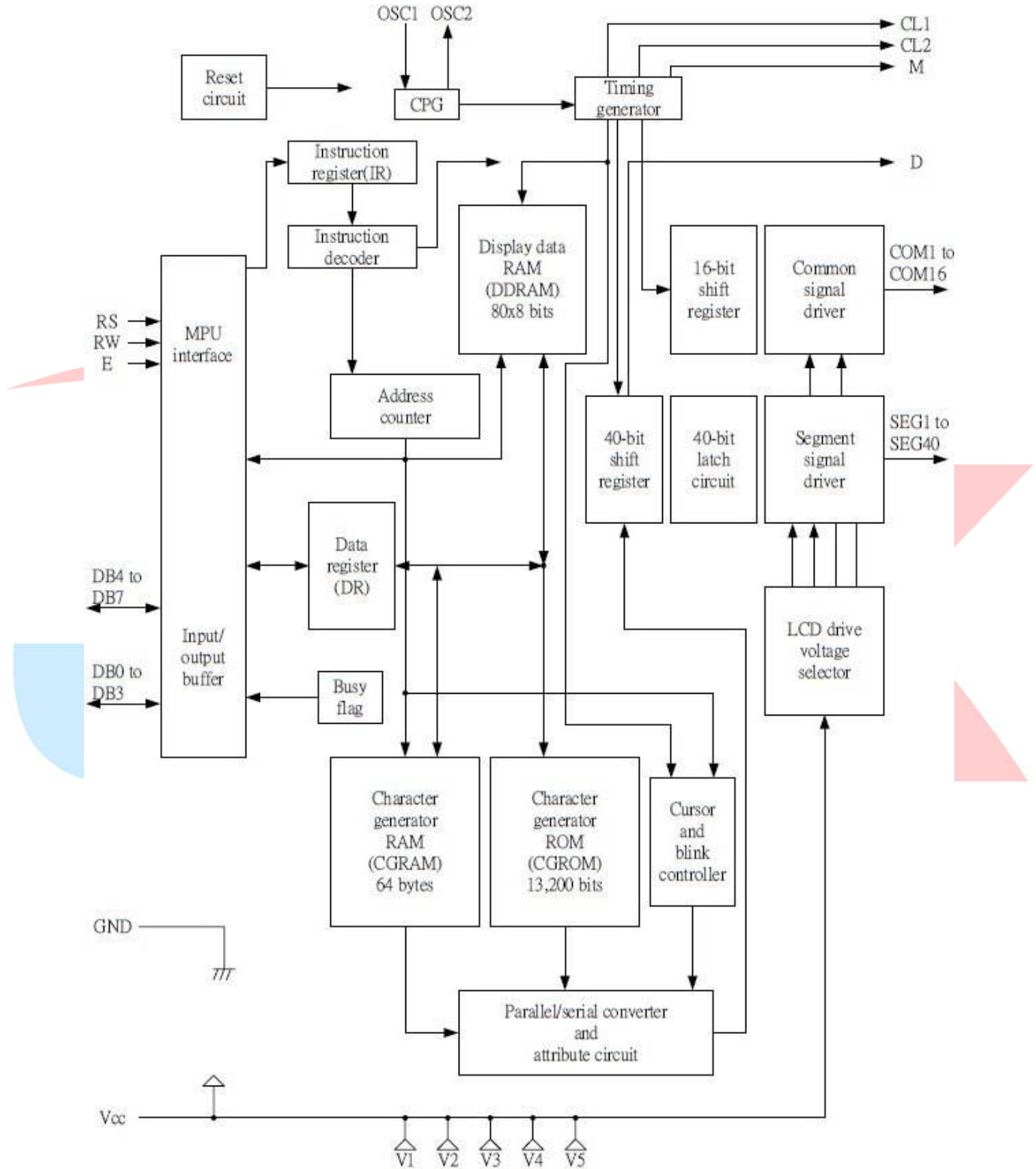


图 2: IC ST7032 内部电路框图

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:

背光板选择白色。

正常工作电流为: 8~20mA (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	
电路电源	VDD - VSS	-0.3		3.6	V
LCD 驱动电压	V0、VOUT	-0.3		13.5	V
LCD 驱动电压	V1\V2\V3\V4	-0.3		V0	V
工作温度		-10		+60	°C
储存温度		-20		+70	°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	8	15	20	mA

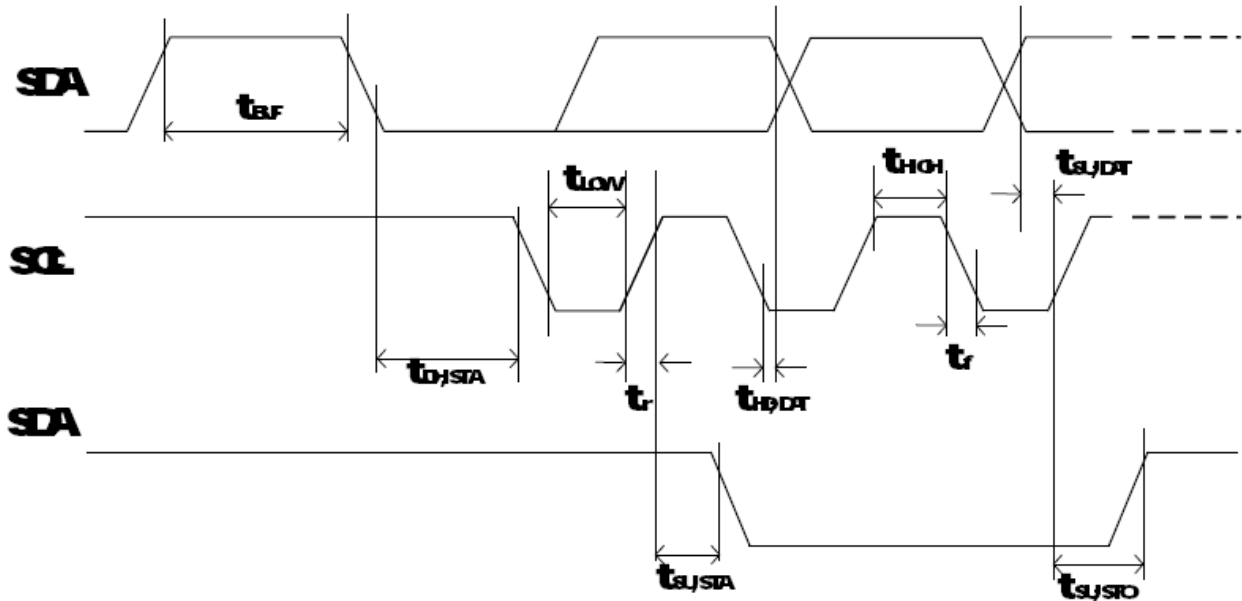
表 3: 直流 (DC) 参数

6. 读写时序特性

6.1 IIC 接口:

从 CPU 写到 ST7032 (Writing Data from CPU to ST7032)

● I2C interface

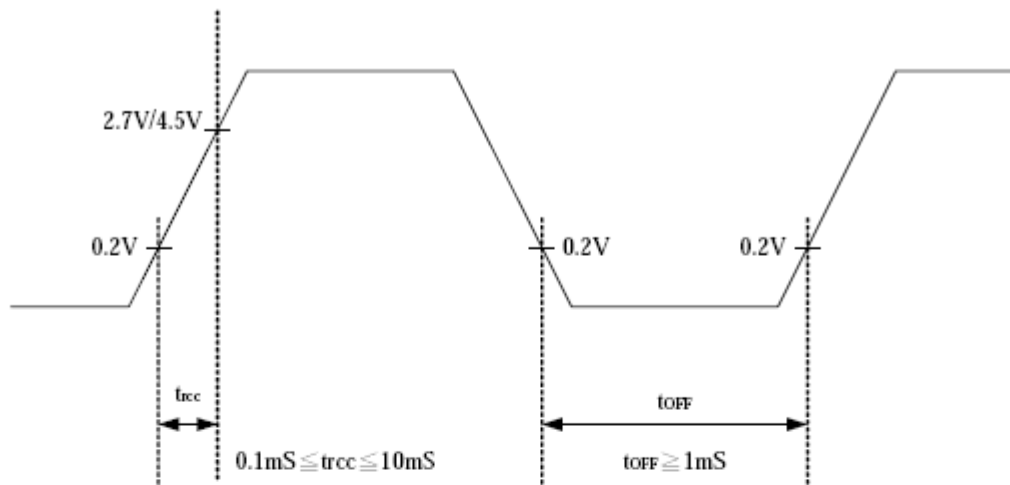


(Ta = 25°C)

Item	Signal	Symbol	Condition	VDD=2.7 to 4.5V Rating		VDD=4.5 to 5.5V Rating		Units
				Min.	Max.	Min.	Max.	
SCL clock frequency	SCL	f _{SCLK}	—	DC	400	DC	400	KHz
SCL clock low period		t _{LOW}		1.3	—	1.3	—	
SCL clock high period		t _{HIGH}		0.6	—	0.6	—	
Data set-up time	SI	t _{SU,DAT}	—	180	—	100	—	ns
Data hold time		t _{HD,DAT}		0	0.9	0	0.9	
SCL,SDA rise time	SCL, SDA	t _r	—	20+0.1Cb	300	20+0.1Cb	300	ns
SCL,SDA fall time		t _f		20+0.1Cb	300	20+0.1Cb	300	
Capacitive load represent by each bus line		C _b	—	—	400	—	400	pf
Setup time for a repeated START condition	SI	t _{SU,STA}	—	0.6	—	0.6	—	us
Start condition hold time		t _{HD,STA}	—	0.6	—	0.6	—	
Setup time for STOP condition		t _{SU,STO}	—	0.6	—	0.6	—	us
Bus free time between a Stop and START condition	SCL	t _{BUF}	—	1.3	—	1.3	—	us

6.2 复位

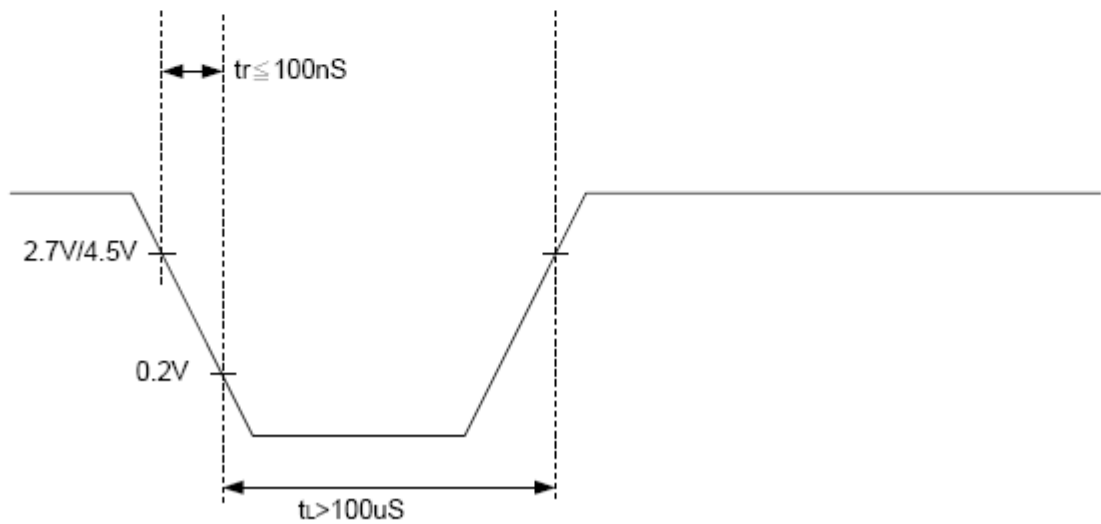
● Internal Power Supply Reset



Notes:

- t_{off} compensates for the power oscillation period caused by momentary power supply oscillations.
- Specified at 4.5V for 5V operation, and at 2.7V for 3V operation.
- If 2.7V/4.5V is not reached during 3V/5V operation, internal reset circuit will not operate normally.

● Hardware reset(XRESET)



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	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	写“20H”到DDRAM和集 DDRAM地址为“00H”从AC DDRAM地址设置为“00H”
(2)返回家 (Return Home)	0	0	0	0	0	0	0	0	1	x	交流电源, 并返回到其原始的光标 如果转移的位置内容 DDRAM 不会改变
(3)进入模式集 (Entry Mode Set)	0	0	0	0	0	0	0	1	I/D	s	设置光标移动方向和指定显示移位。 这些操作的过程中进行数据的写入 和读出
(4) 显示 (ON/OFF)	0	0	0	0	0	0	1	D	C	B	D=1;这个屏幕上 C=1;光标上 B=1;光标位置
(5) 功能设置 (Function Set)	0	0	0	0	1	DL	N	DH	0	IS	DL;数据接口是 8/4 位 N;行数是 2/1, DH;双倍高度字体 方法是: 指令表中选择
(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)读忙标志和地址 (Read Busy flay and address)	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	无论是在内部操作或可以被称为阅 读 BF.地址计数器的内容也可以被读 取
(7)写数据到ROM (Write data to RAM)	1	0	D7	D6	D5	D4	D3	D2	D1	D0	写数据到内部 RAM (DDRAM/CGRAM/ICONRAM 的)
读取数据从ROM (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)
*注:该位是测试命令, 必须设置“0”											
指令表 0 (0)											
(8)光标或显示位移 (Cursor or Display Shift)	0	0	0	0	0	1	S/C	R/L	X	X	S/C 和 R/L 将光标移动和显示移控制位, 其方向 不改变 DDRAM 的数据情况
(9)设置CGROM (Set CGRAM)	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	设置 CGRAM 地址地址对抗
指令表 1 (1)											
内部OSC频率 (internal OSC frequency)	0	0	0	0	0	1	BS	F2	F1	F0	BS=1:1/4 偏压 BS=0:1/5 偏压 F2~0:FR 频率调整内部振荡频率
设置图标地址 (Set ICON address)	0	0	0	1	0	0	AC3	AC2	AC1	AC0	设置 ICON 地址在地址计数器
电源/ICON控制/CONTR	0	0	0	1	0	1	Lon	Bon	C5	C4	离子:ICON 显示开/关

AST集 (Power/ICON/control/Contrast set)											苯教:设置升压电路开/关 C5, C4 内部跟随模式下的对比度设置
信号控制 (Follower control)	0	0	0	1	1	0	Fon	Rab 2	Rab 1	Rab 0	丰: 设置/关闭输出电路 RAB2~0: 选择放大的比例
对比度设置 (Contrast set)	0	0	0	1	1	1	C3	C2	C1	C0	内部跟随模式下的对比度设置

温馨提示: 请详细参考 IC 资料”ST7032.PDF”的第 20 页。

7.3 字符库及对应关系

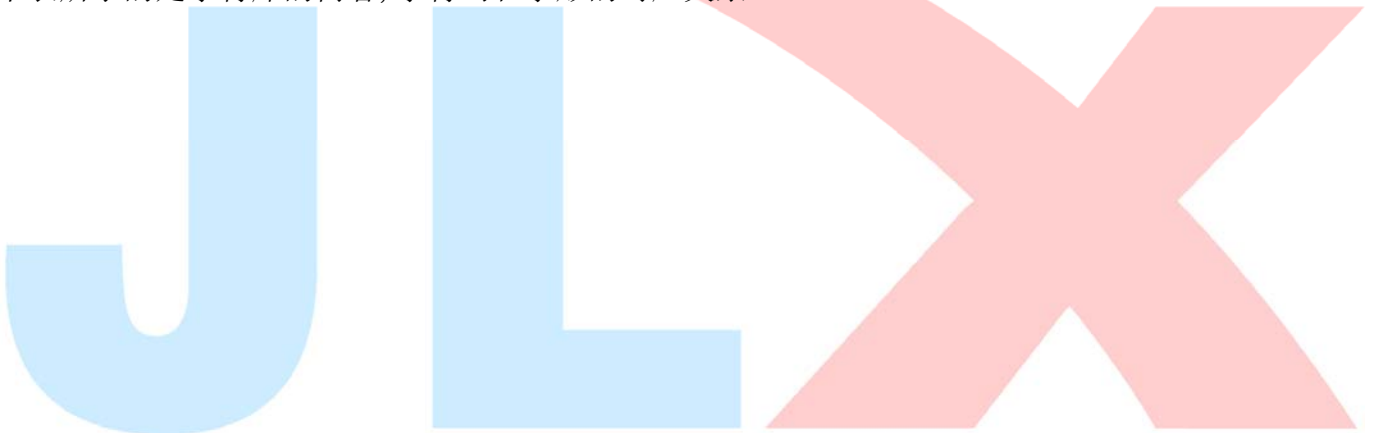
7.3.1 显示位与 DD RAM 地址的对应关系

表 9.

显示位序号		DD RAM 地址										
DD RAM 地址(HEX)	第一行	00	01	02	03	04	27				
	第二行	40	41	42	43	44	67				

7.3.2 标准字符库

下表所示的是字符库的内容, 字符码和字形的对应关系。



Product Number	OPR1	OPR2	Support Character
ST7032-0D	1	1	English/Japan/European

Table A1. Correspondence between Character Codes and Character Patterns
ST7032-0D (ITO option OPR1=1, OPR2=1)

b7-b4 b3-b0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6
0001	!	@	#	\$	%	&	'	()	*	+	=	>	<	?	0
0010	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
0011	q	r	s	t	u	v	w	x	y	z	[\]	^	_	~
0100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
0101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
0110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
0111	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ
1000	€	£	¥	¤	¢	©	®	™	®	™	®	™	®	™	®	™
1001	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1010	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
1011	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
1100	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3
1101	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1110	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1111	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1

表 10. ST7032-0D 字库表

Table A2. Select display pattern in CGRAM or CGROM (use OPR1, OPR2)

b7-b4 b3-b0	0000	0001	...	b7-b4 b3-b0	0000	0001	...	b7-b4 b3-b0	0000	0001	...	b7-b4 b3-b0	0000	0001	...					
0000	Replaced By CGRAM Pattern	0000	...	0000	Replaced By CGRAM Pattern	0000	...	0000	Replaced By CGRAM Pattern	0000	...	0000	0000	0001	...					
0001		0001	...	0001		0001	...	0001		0001	0001	0001	...	0001	0001	0001	...			
0010		0010	...	0010		0010	...	0010		0010	0010	0010	...	0010	0010	0010	...			
0011		0011	...	0011		0011	...	0011		0011	0011	0011	...	0011	0011	0011	...			
0100		0100	...	0100		0100	...	0100		0100	0100	0100	...	0100	0100	0100	...			
0101		0101	...	0101		0101	...	0101		0101	0101	0101	...	0101	0101	0101	...			
0110		0110	...	0110		0110	...	0110		0110	0110	0110	...	0110	0110	0110	...			
0111		0111	...	0111		0111	...	0111		0111	0111	0111	...	0111	0111	0111	...			
1000		Replaced By CGRAM Pattern	1000	...		1000	Replaced By CGRAM Pattern	1000		...	1000	Replaced By CGRAM Pattern	1000	...	1000	1000	1001	...		
1001			1001	...		1001		1001		...	1001		1001	1001	1001	...	1001	1001	1001	...
1010			1010	...		1010		1010		...	1010		1010	1010	1010	...	1010	1010	1010	...
1011	1011		...	1011	1011	...		1011	1011	1011	1011		...	1011	1011	1011	...			
1100	1100		...	1100	1100	...		1100	1100	1100	1100		...	1100	1100	1100	...			
1101	1101		...	1101	1101	...		1101	1101	1101	1101		...	1101	1101	1101	...			
1110	1110		...	1110	1110	...		1110	1110	1110	1110		...	1110	1110	1110	...			
1111	1111		...	1111	1111	...		1111	1111	1111	1111		...	1111	1111	1111	...			

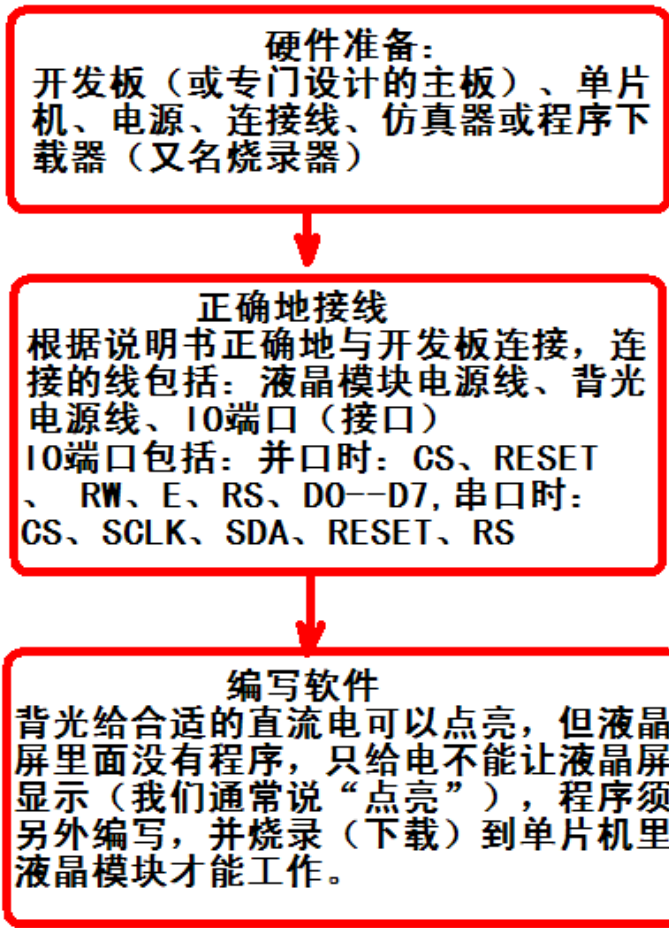
OPR2,OPR1=(0,0) OPR2,OPR1=(0,1) OPR2,OPR1=(1,0) OPR2,OPR1=(1,1)

表 11. ST7032-0D 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

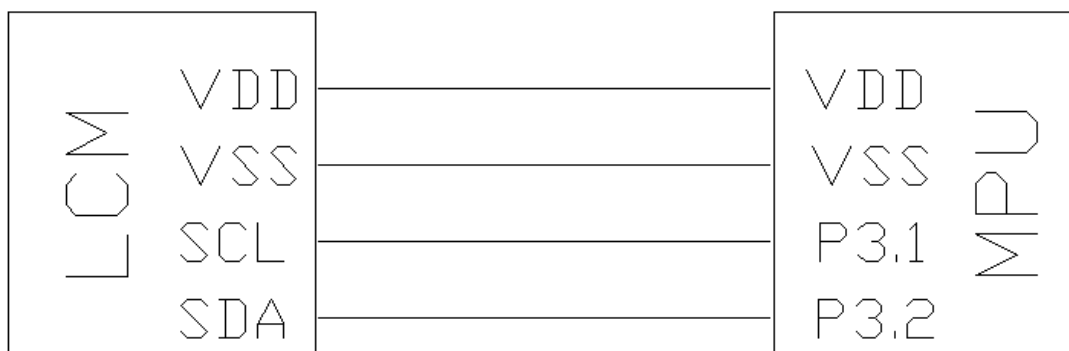
点亮液晶模块的步骤



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-390**"); /*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-390"); /*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();
    }
}
```