

# JLX1602G-935-BN 使用说明书

## (焊接式 FPC)

### 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	背光参数	4
5	技术参数	4~5
6	时序特性	5~6
7	指令功能及硬件接口与编程案例	6~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-935-BN 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-935-BN 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 点阵或 4 个 5\*8 自编字符）。

## 2. JLX1602G-935-BN 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：背光带有挡墙，焊接式 FPC。

2.4 IC 采用 ST7032, 功能强大，稳定性好

2.4 功耗低：1~200mW（关掉背光：[0.3mA@3.3V](#), 打开背光不大于 200mW）；

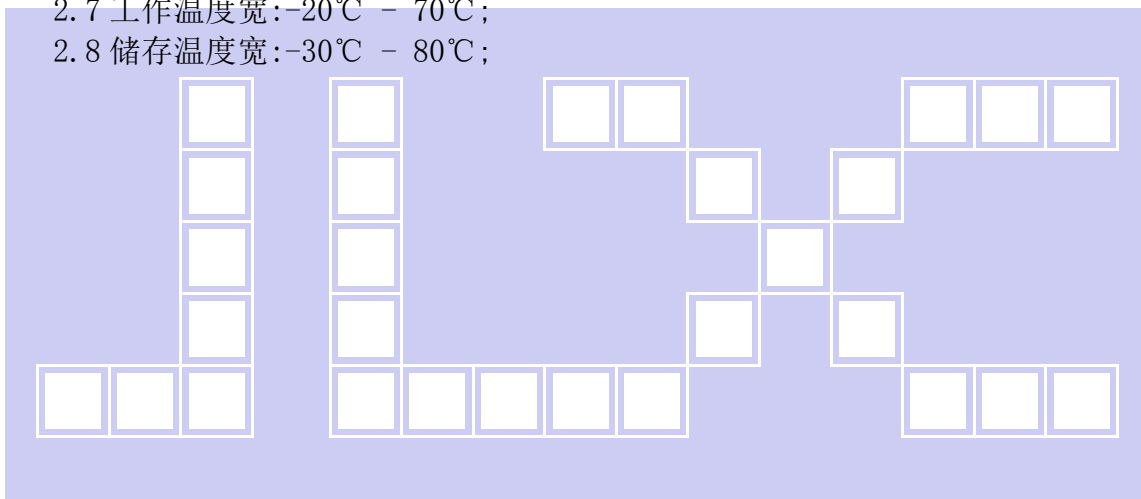
2.5 显示内容:256 种字符(5×8 点字型)；

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）；

2.6 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℃；



## 3. 外形尺寸及接口引脚功能

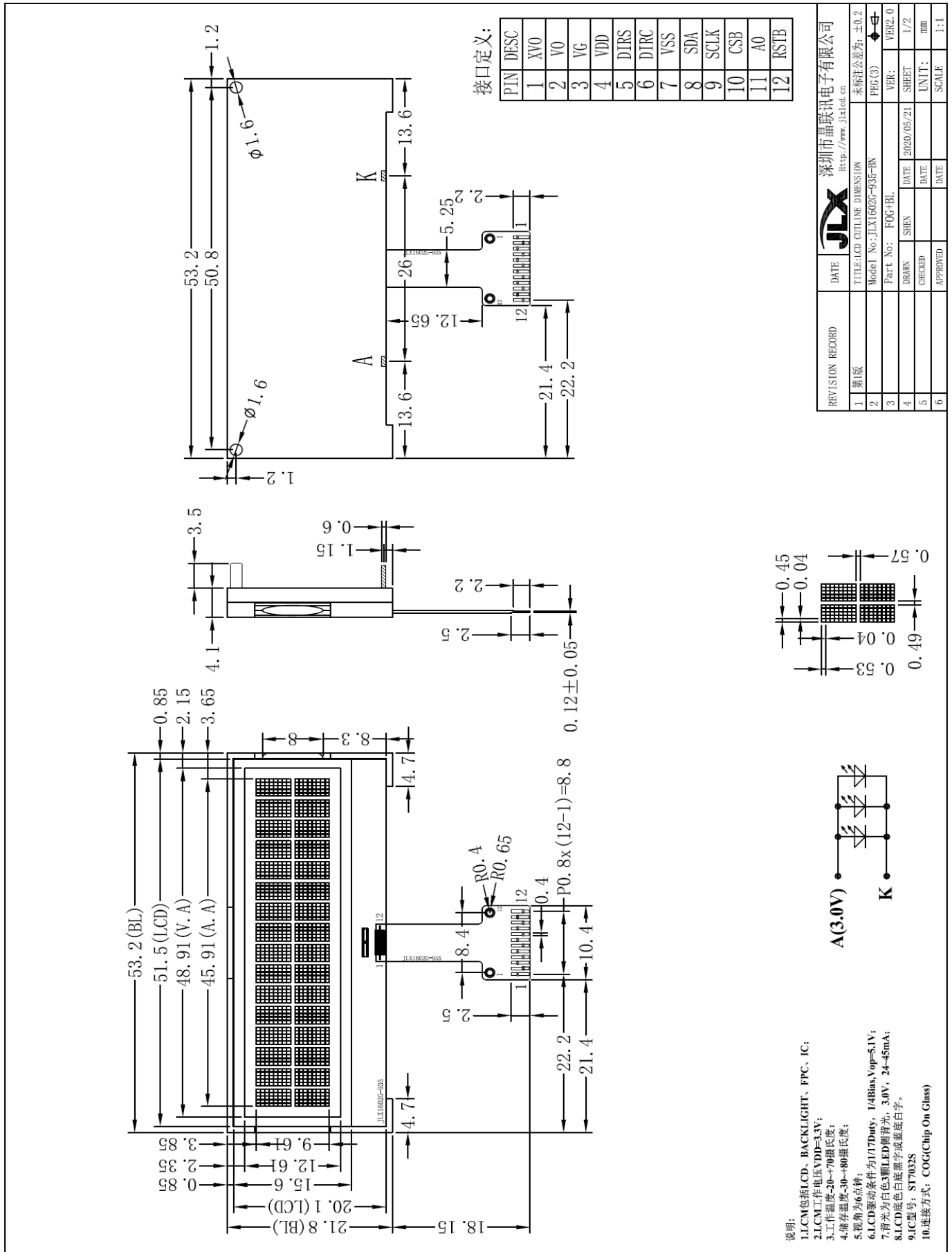


图 1. 外形尺寸

## 模块的接口引脚功能

引线号	符号	名称	功能
1	XV0	偏置电压	
2	V0	偏置电压	
3	VG	LCD 倍压输出	
4	VDD	供电电源正极	供电电源正极, , 3.3V
5	DIRS	指令	DIRS=1; DIRC=0;六点钟方向显示 DIRS=0; DIRC=1;十二点钟方向显示
6	DIRC	指令	
7	VSS	接地	0V
8	SDA	串行数据	串行数据 SDA
9	SCLK	串行时钟	串行时钟 SCK
10	CSB	片选	低电平片选
11	A0 (RS)	寄存器选择信号	H:数据寄存器 0:指令寄存器
12	RSTB (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

表 1: 模块的接口引脚功能

## 4. 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:  $-30 \sim +80^{\circ}\text{C}$ ;

背光白色;

正常工作电流为:  $16 \sim 40\text{mA}$  (LED 灯数共 3 颗);

工作电压: 3.0V; (接 3.3V 串 20 欧电阻, 接 5.0V 串 120 欧电阻)

## 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		4.0	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	4.0	V
背光工作电压	VLED		2.9	3.0	3.1	V
输入高电平	V <sub>IH</sub>	-	0.8VDD	-	VDD	V
输入低电平	V <sub>IO</sub>	-	0	-	0.2VDD	V
输出高电平	V <sub>OH</sub>	I <sub>OH</sub> = 0.2mA	0.8VDD		VDD	V
输出低电平	V <sub>OO</sub>	I <sub>OO</sub> = 1.2mA	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 3 颗 LED 灯并联)	24	45	60	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 ST7032S (Writing Data from CPU to ST7032S)

#### System Bus Timing for 4-Line Serial Interface

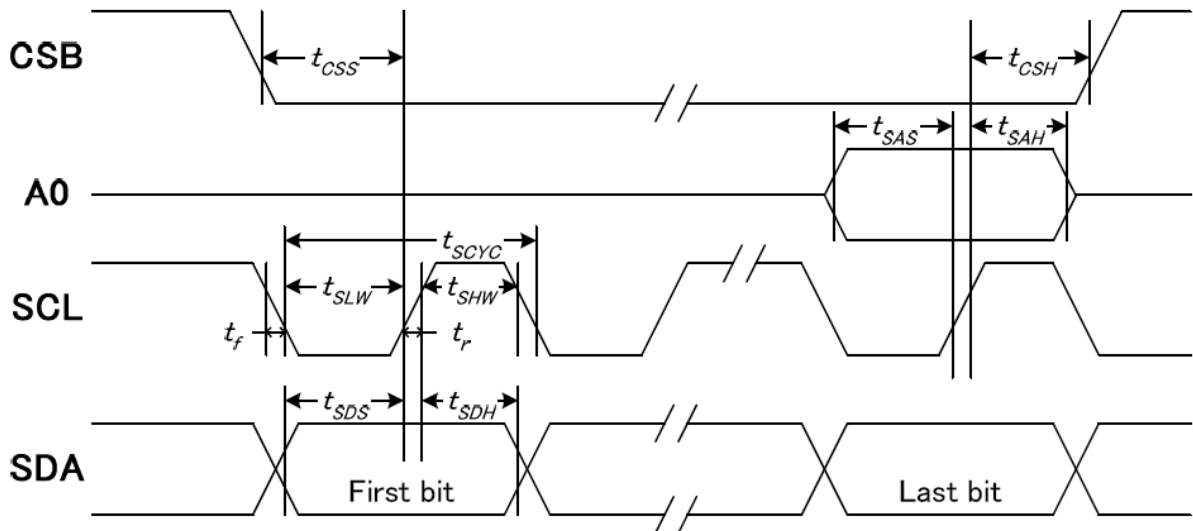


图 2. 从 CPU 写到 ST7032S (Writing Data from CPU to ST7032S)

### 6.2 串行接口: 时序要求 (AC 参数):

写数据到 ST7032S 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 <sub>scyc</sub>	引脚: SCK	150	—	—	ns
保持SCK高电平脉宽 (SCK "H" pulse width)	T <sub>shw</sub>	引脚: SCK	20	—	—	ns
保持SCK低电平脉宽 (SCK "L" pulse width)	T <sub>slw</sub>	引脚: SCK	120	—	—	ns
地址建立时间 (Address setup time)	T <sub>sas</sub>	引脚: RS	10	—	—	ns
地址保持时间 (Address hold time)	T <sub>sah</sub>	引脚: RS	150	—	—	ns
数据建立时间 (Data setup time)	T <sub>sds</sub>	引脚: SI	10	—	—	ns
数据保持时间 (Data hold time)	T <sub>sdh</sub>	引脚: SI	10	—	—	ns
片选信号建立时间 (CS-SCL time)	T <sub>css</sub>	引脚: CS	20	—	—	ns

片选信号保持时间 (CS-SCL time)	$T_{csh}$	引脚: CS	200	—	—	ns
---------------------------	-----------	--------	-----	---	---	----

$VDD = 3.3V, T_a = 25^\circ C$

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

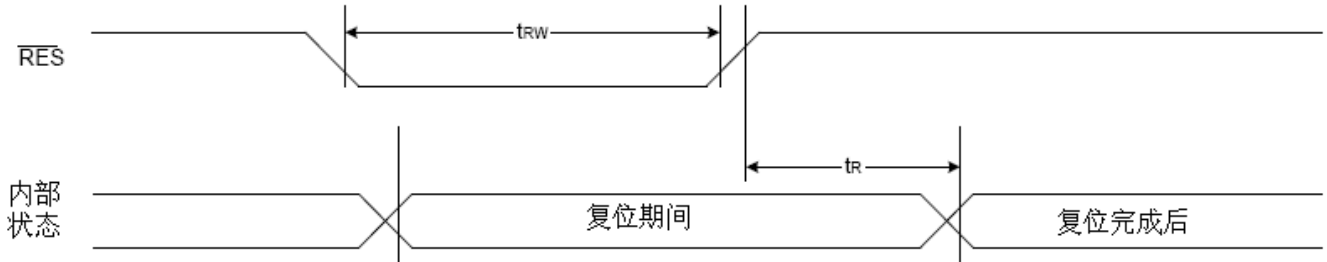


图 3: 电源启动后复位的时序

表 5: 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	$trw$	引脚: RES	1.0	—	—	us
复位到内部状态延时	$tR$	引脚: RES 及 IC 内部状态	—	—	1.0	

## 7. 指令功能:

### 7.1 指令表

表 6

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	<b>x</b>	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	<b>s</b>	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>B</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)
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 AST集 (Power/ICON/control/Contrast set)	0	0	0	1	0	1	Lon	Bon	C5	C4		离子:ICON 显示开/关 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

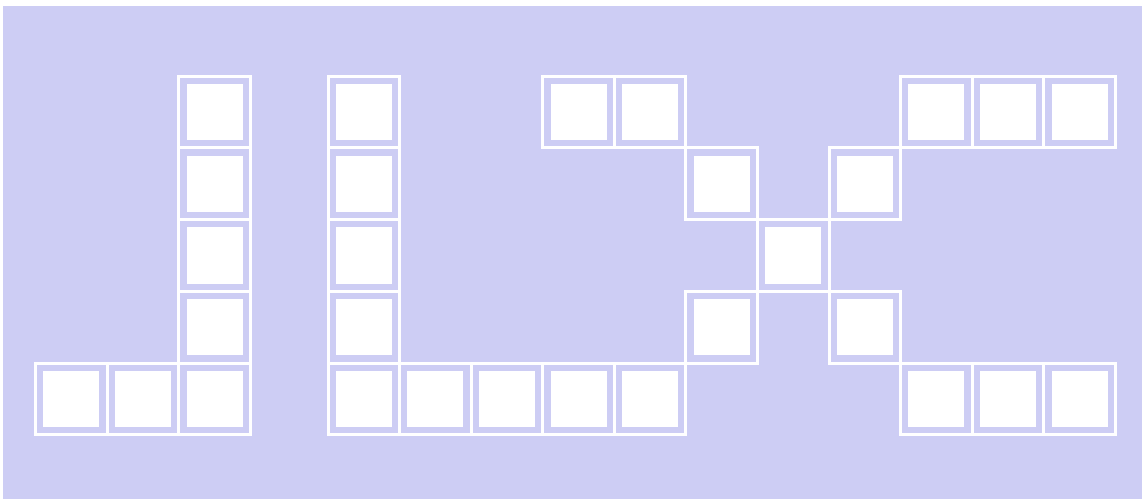
请详细参考 IC 资料”ST7032S\_V1.1.PDF”的第 20 页。

## 7.2 字符库及对应关系

### 7.2.1 显示位与 DD RAM 地址的对应关系

表 7.

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





### 7.2.2 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。

The available character fonts in CGROM are shown below:

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

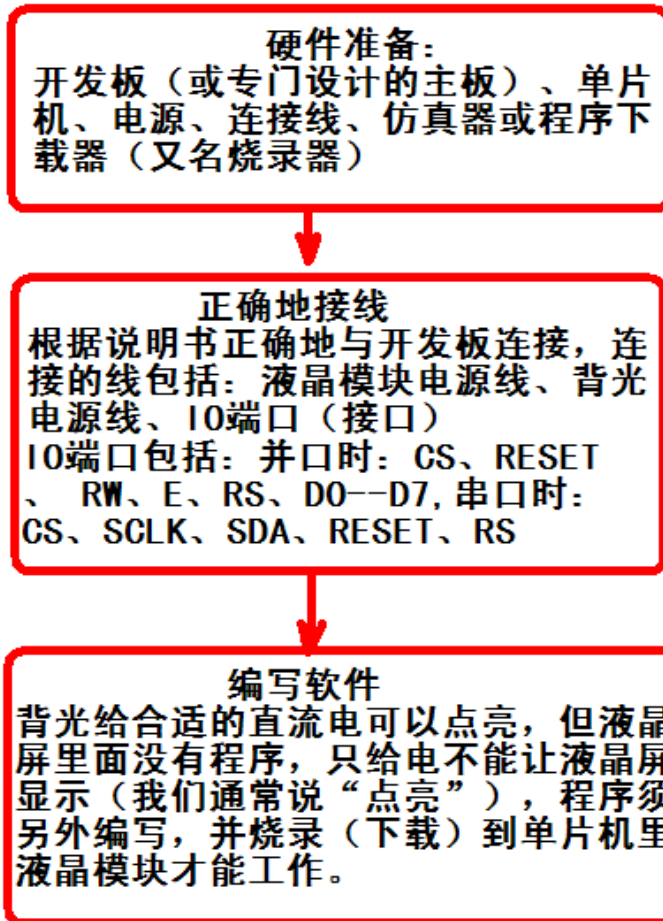
### ST7032S-1A

b7-b4 b3-b0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0001	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
0010	W	X	Y	Z	[	\	]	^	_	`	a	b	c	d	e	f
0011	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
0100	w	x	y	z	{		}	~	`	a	b	c	d	e	f	g
0101	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
0110	x	y	z	{		}	~	`	a	b	c	d	e	f	g	h
0111	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x
1000	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t
1001	u	v	w	x	y	z	{		}	~	`	a	b	c	d	e
1010	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u
1011	v	w	x	y	z	{		}	~	`	a	b	c	d	e	f
1100	w	x	y	z	{		}	~	`	a	b	c	d	e	f	g
1101	x	y	z	{		}	~	`	a	b	c	d	e	f	g	h
1110	y	z	{		}	~	`	a	b	c	d	e	f	g	h	i
1111	z	{		}	~	`	a	b	c	d	e	f	g	h	i	j

### 7.3 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



### 7.4 程序举例：

液晶模块与 MPU(以 8051 系列单片机为例)接口图如下：

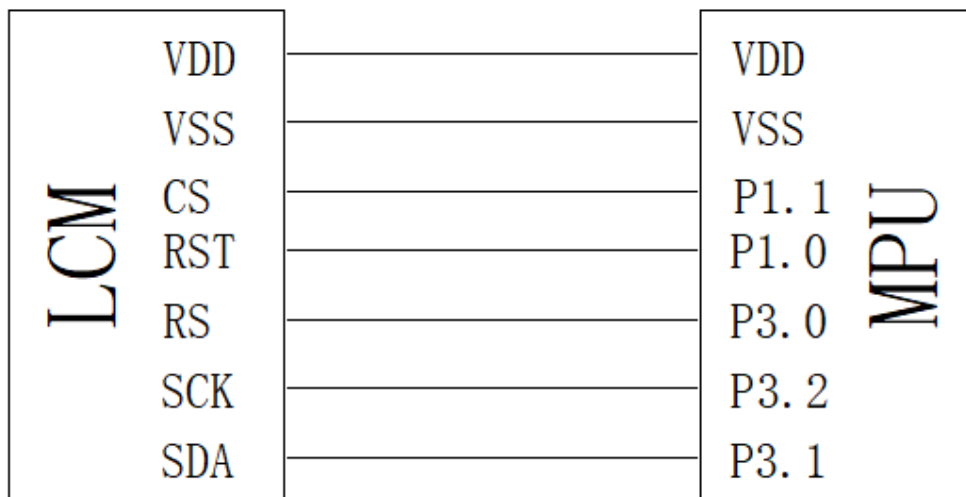
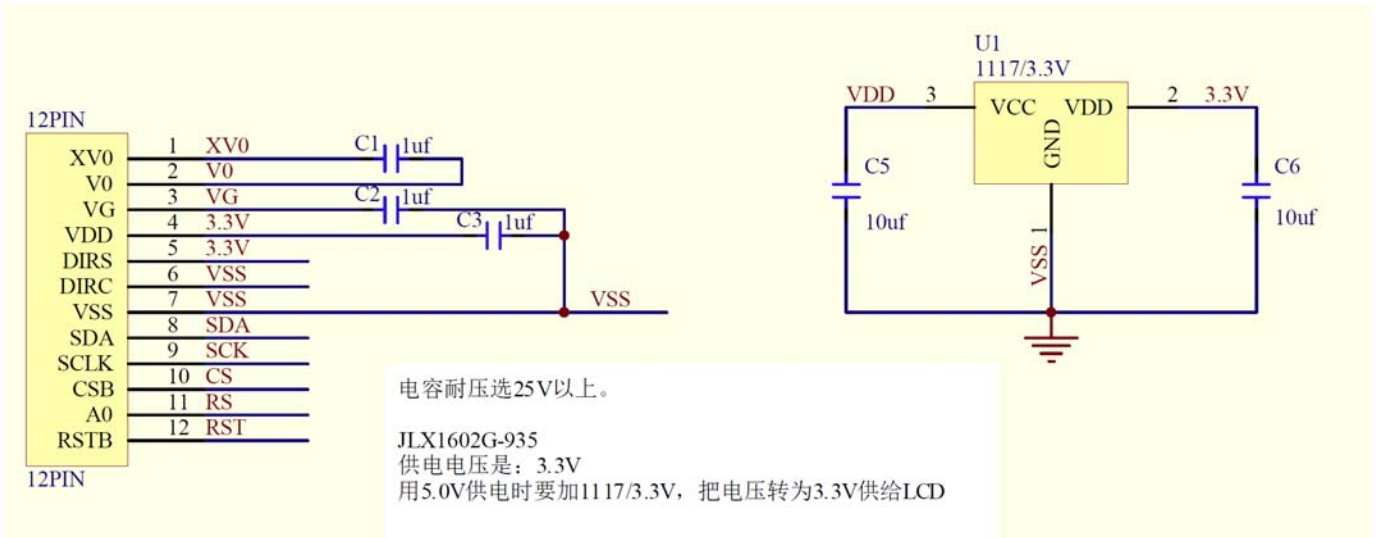


图 4. 串行接口



### 点亮液晶模块的编程步骤



```
/*=====*/
/* ST7032S 测试程序 */
/* JLX1602G-935 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 CS=P1^1;
sbit RST=P1^0;
sbit RS=P3^0;
sbit SCK=P3^2;
sbit SDA=P3^1;

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(2500);
}

/*写指令到LCD模块*/
void transfer_command(int data1)
{
char i;
CS=0;
RS=0;
for(i=0;i<8;i++)
{
SCK=0;
```



```

        if(data1&0x80) SDA=1;
        else SDA=0;
        SCK=1;
        delay_us(5);
        data1=data1<<=1;
    }
    CS=1;
}

```

```

/*写数据到LCD 模块*/
void transfer_data(int data1)
{
    char i;
    CS=0;
    RS=1;
    for(i=0;i<8;i++)
    {
        SCK=0;
        if(data1&0x80) SDA=1;
        else SDA=0;
        SCK=1;
        delay_us(5);
        data1=data1<<=1;
    }
    CS=1;
}

```

```

/*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(0x6d);          //
    delay(5);
    transfer_command(0x55);          //粗调对比度, 范围 0x54-0x57
    delay(5);
    transfer_command(0x7b);          //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)
{
    /****12点阵显示***/
    // DIRS=0;
    // DIRC=1;
    /****6点阵显示***/
    DIRS=1;
    DIRC=0;
    /*****/
    RST =0;
    delay(20);
    RST =1;
    delay(100);
    initial_lcd();
    write_CGRAM();

    while(1)
    {
        disp_char(1,1,"**JLX1602G-935**"); /*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-935"); /*在第1行，第1列，显示字符... */
        disp_char(2,1,"*2020/03/26 cog**"); /*在第2行，第1列，显示字符... */
        waitkey();
        disp_char(1,1,"*0123456789ABCD*"); /*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,"*xxxxxxxxxxxx*"); /*在第1行，第1列，显示字符... */
        disp_char(2,1,"*cccccccccccc*"); /*在第2行，第1列，显示字符... */
        waitkey();
        disp_CGRAM();
        waitkey();
    }
}

```



**-END-**

