

JLX160160G-161-PL 使用说明书

(带字库 IC)

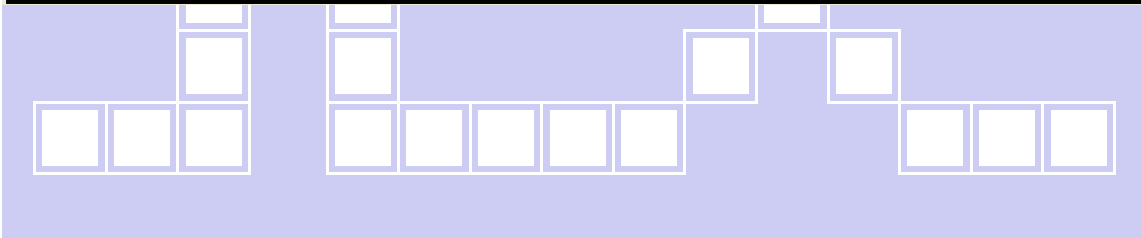
目 录

序号	内 容 标 题	页 码
1	字库	2~3
2	外形及接口引脚功能	4~5
3	基本原理	5~6
4	技术参数	6
5	指令功能及硬件接口与编程案例	6~末页

1. 字库

字库 IC(IC 型号: JLX-GB2312-3205, 此 IC 为可选的配件) 自带字库内容:

分类	字库内容	编码体系 (字符集)	字符数
汉字字符	11X12 点 GB2312 标准点阵字库	GB2312	6763+846
	15X16 点 GB2312 标准点阵字库	GB2312	6763+846
	24X24 点 GB2312 标准点阵字库	GB2312	6763+846
	32X32 点 GB2312 标准点阵字库	GB2312	6763+846
	6X12 点国标扩展字符	GB2312	126
	8X16 点国标扩展字符	GB2312	126
	12X24 点国标扩展字符	GB2312	126
	16X32 点国标扩展字符	GB2312	126
ASCII 字符	5X7 点 ASCII 字符	ASCII	96
	7X8 点 ASCII 字符	ASCII	96
	6X12 点 ASCII 字符	ASCII	96
	8X16 点 ASCII 字符	ASCII	96
	8X16 点粗体 ASCII 字符	ASCII	96
	12 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	96
	12 点阵不等宽 ASCII 白正 (Times New Roman) 字符	ASCII	96
	16 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	96
	16 点阵不等宽 ASCII 白正 (Times New Roman) 字符	ASCII	96
	24 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	96
	24 点阵不等宽 ASCII 白正 (Times New Roman) 字符	ASCII	96
	32 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	96
32 点阵不等宽 ASCII 白正 (Times New Roman) 字符	ASCII	96	
输入法码表	全拼输入法码表	GB2312	



字型样张

11X12 点 GB2312 汉字

啊阿埃挨哎唉哀皑癌蔼矮艾碍爱隘鞍
 氨安俺按暗岸胺案肮盎凹敖熬翱袄
 傲奥懊澳芭捌扒叭吧芭八疤巴拔跋靶
 把耙坝霸罢爸白柏百摆佰败拜裨斑班
 搬扳般颁板版扮拌伴瓣半办絆邦帮梆
 榜膀绑棒棒蚌傍傍苞胞包褒剥薄苞
 保堡宝抱报暴豹鲍爆杯碑悲卑北辈
 背贝狈倍狻惫惫惫奔笨笨崩绷甬

15X16 点 GB2312 汉字

啊阿埃挨哎唉哀皑癌蔼矮艾碍爱隘鞍
 氨安俺按暗岸胺案肮盎凹敖熬翱袄
 傲奥懊澳芭捌扒叭吧芭八疤巴拔跋靶
 把耙坝霸罢爸白柏百摆佰败拜裨斑班
 搬扳般颁板版扮拌伴瓣半办絆邦帮梆
 榜膀绑棒棒蚌傍傍苞胞包褒剥薄苞
 保堡宝抱报暴豹鲍爆杯碑悲卑北辈
 背贝狈倍狻惫惫惫奔笨笨崩绷甬

24X24 点 GB2312 汉字

啊阿埃挨哎唉哀皑
 癌蔼矮艾碍爱隘鞍
 氨安俺按暗岸胺案
 肮盎盎凹敖熬翱袄

32X32 点 GB2312 汉字

啊阿埃挨哎唉
 哀皑癌蔼矮艾
 碍爱隘鞍氨安
 碍爱隘鞍氨安

5x7 点 ASCII 字符

```
!"#$%&'()*+,-./0123456789:;  
=>?@ABCDEFGHIJKLMN O PQRSTU V  
WXY[\]^_`abcdefghijklmnopqrstuvwxyz
```

7x8 点 ASCII 字符

```
!"#$%&'()*+,-./01234  
56789:;<=>?@ABCDEFGHIJ  
KLMNOPQRSTUVWXYZ[\]^_`  
abcdefghijklmnopqrstuvwxyz  
6789:;<=>?@ABCDEFGHIJ
```

6x12 点 ASCII 字符

```
!"#$%&'()*+,-./0123456789:;  
=>?@ABCDEFGHIJKLMN O PQRSTU V W  
XYZ[\]^_`abcdefghijklmnopqrstuvwxyz  
{|}~áâãäéèëìíîïðóôù
```

8x16 点 ASCII 字符

```
!"#$%&'()*+,-./0123456789:;  
=>?@ABCDEFGHIJKLMN O PQRSTU V  
WXY[\]^_`abcdefghijklmnopqrstuvwxyz
```

12 点阵不等宽 ASCII 方头

```
!"#$%&'()*+,-./01234  
56789:;<=>?@ABCDEFGHIJ  
KLMNOPQRSTUVWXYZ[\]^_`  
abcdefghijklmnopqrstuvwxyz  
6789:;<=>?@ABCDEFGHIJ
```

16 点阵不等宽 ASCII 方头

```
!"#$%&'()*+,-./0123456789:;  
=>?@ABCDEFGHIJKLMN O PQRSTU V W  
XYZ[\]^_`abcdefghijklmnopqrstuvwxyz  
{|}~áâãäéèëìíîïðóôù
```

2. 外形尺寸及接口引脚功能

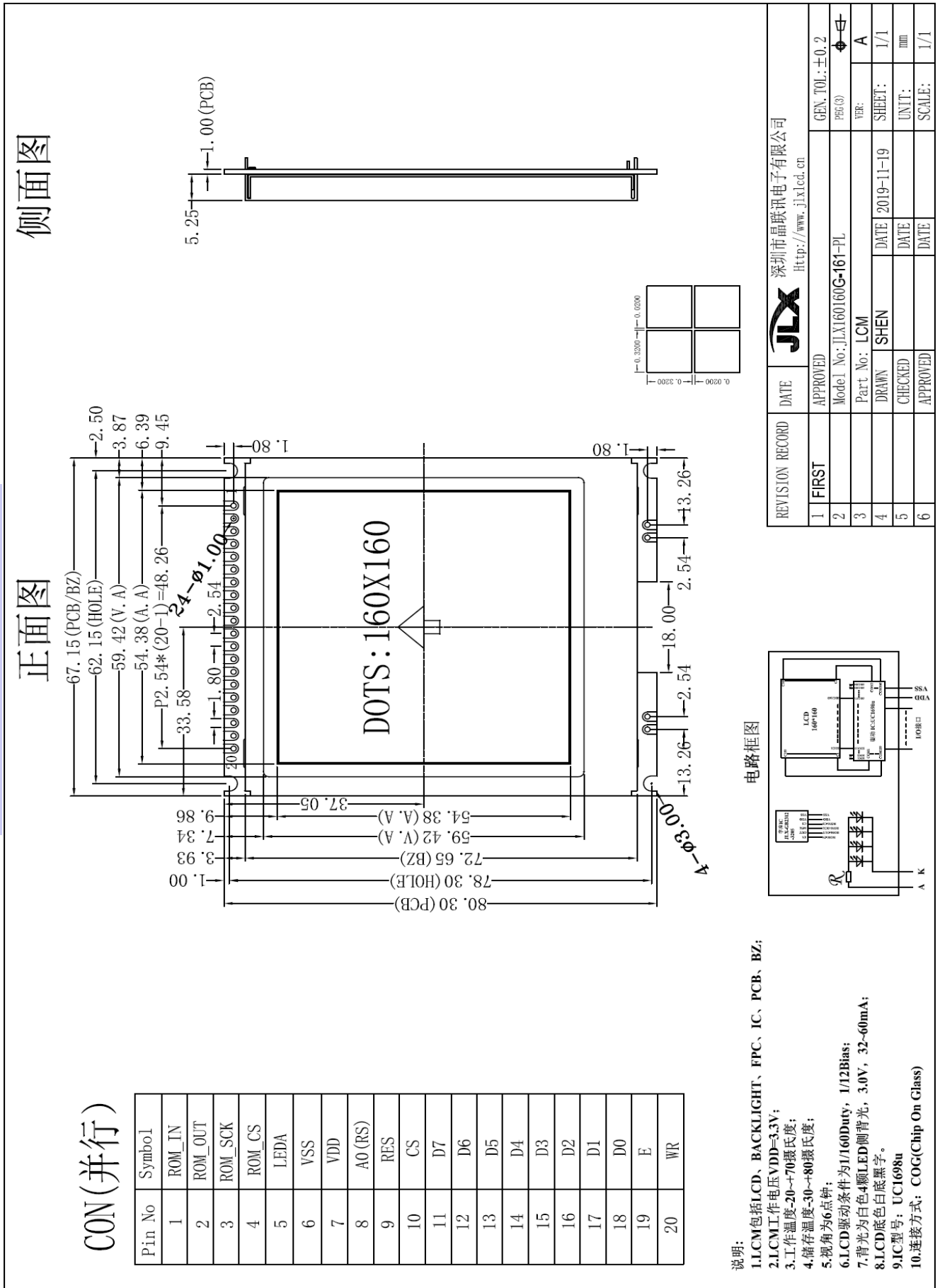


图 1. 外形尺寸

模块的接口引脚功能

引线号	符号	名称	功能
1	ROM_IN	字库 IC 接口 SI	串行数据输入
2	ROM_OUT	字库 IC 接口 SO	串行数据输出
3	ROM_SCK	字库 IC 接口 SCLK	串行时钟输入
4	ROM_CS	字库 IC 接口 CS#	片选输入
5	LEDA	背光电源正极	背光电源正极, 同 VDD 电压 (5V 或 3.3V)
6	VSS	供电电源负极	供电电源负极
7	VDD	供电电源正极	供电电源正极 (注意: 购买时须选择 3.3V 或者是 5V 供电)
8	A0 (RS)	寄存器选择信号	H: 数据寄存器 0: 指令寄存器 (IC 资料上所写为 "A0")
9	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
10	CS	片选	低电平片选
11-18	D7-D0	I/O	数据总线
19	E (RD)	使能信号, 或 "读"	并行接口时并且选择 6800 时序时: 使能信号, 高电平有效. 并行接口时并且选择 8080 时序时: 读信号, 低电平有效.
20	WR (R/W)	读/写, 或写	并行接口时并且选择 6800 时序时: H: 读信号 L: 写信号 并行接口时并且选择 8080 时序时: 写信号, 低电平有效.

表 1: 模块的接口引脚功能

3. 基本原理

3.1 液晶屏 (LCD)

在 LCD 上排列着 160×160 点阵, 160 个列信号与驱动 IC 相连, 160 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)。

3.2 工作电路图:

图 1 是 JLX160160G-161 图像点阵型模块的电路框图, 它由驱动 IC UC1698u 及几个电阻电容组成。

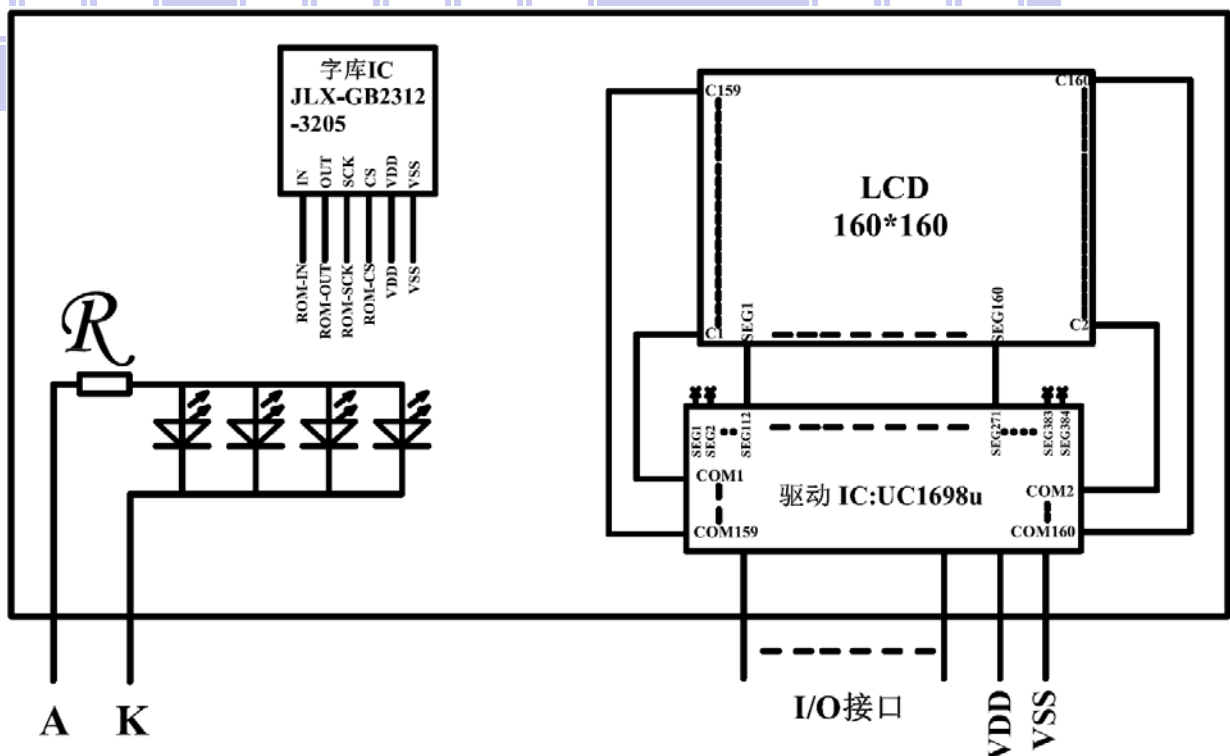


图 2: JLX160160G-161-PL 图像点阵型液晶模块的电路框图

3.3 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: $-30\sim+80^{\circ}\text{C}$;

背光灯选用白色;

正常工作电流为: $32\sim 80\text{mA}$;

工作电压: 3.0V (PCB 已加限流电阻, 供电同 VDD 电压)

4. 技术参数

4.1 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		3.45	V
LCD 驱动电压	VDD - V0	VDD - 13.0		VDD + 0.3	V
静电电压		-	-	100	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

4.2 直流 (DC) 参数

可以选择 3.3V 供电及 5.0V 供电两种方式:

名称	符号	测试条件	标准值			单位
			MIN	TYPE	MAX	
工作电压 5.0	VIN	5.0V 供电	4.7	5.0	5.2	V
工作电压 3.3	VIN	3.3V 供电	2.7	3.3	3.45	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 = 3.3V	-		0.3	mA
背光工作电流	ILED	VLED=3.0V	32	60	80	mA

表 3: 直流 (DC) 参数

4.3 LCD 驱动 IC 指令表详见“JLX160160G-161-PN”的中文说明书

5.1 字库 IC (JLX-GB2312-3205) 的操作指令及点阵数据的调用方法:

5.1.1 字库 IC 的操作指令只有两条, 两条只选一条进行使用, 操作指令表如下:

Instruction Set

Instruction	Description	Instruction Code(One-Byte)		Address Bytes	Dummy Bytes	Data Bytes
READ	Read Data Bytes	0000 0011	03 h	3	—	1 to ∞
FAST_READ	at Higher Speed	0000 1011	0B h	3	1	1 to ∞

Read Data Bytes

所有对本芯片 SPI 接口的操作只有 2 个，那就是 Read Data Bytes (READ “一般读取”)和 Read Data Bytes at Higher Speed (FAST_READ “快速读取点阵数据”)。

以下分别介绍一般读取和快速读取:

5.2.1.1 Read Data Bytes (一般读取)

Read Data Bytes 需要用指令码来执行每一次操作。READ 指令的时序如下(图):

首先把片选信号 (CS#) 变为低, 紧跟着的是 1 个字节的命令字 (03 h) 和 3 个字节的地址和通过串行数据输入引脚 (SI) 移位输入, 每一位在串行时钟 (SCLK) 上升沿被锁存。

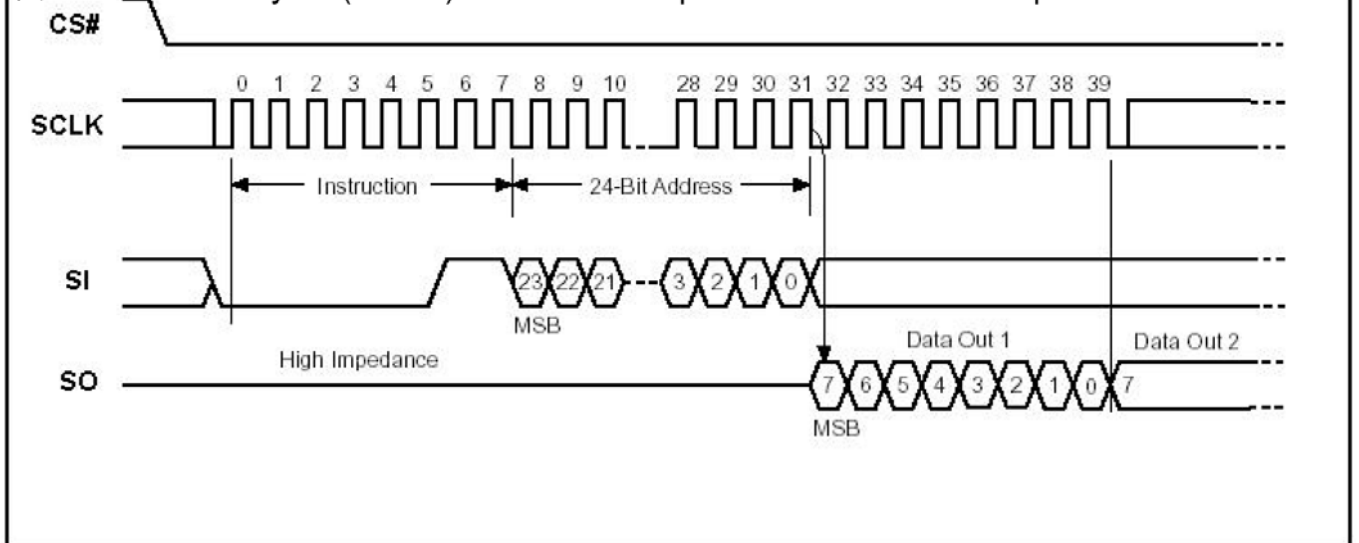
然后该地址的字节数据通过串行数据输出引脚 (SO) 移位输出, 每一位在串行时钟 (SCLK) 下降沿被移出。

读取字节数据后, 则把片选信号 (CS#) 变为高, 结束本次操作。

如果片选信号 (CS#) 继续保持为底, 则下一个地址的字节数据继续通过串行数据输出引脚 (SO) 移位输出。



图: Read Data Bytes (READ) Instruction Sequence and Data-out sequence



5.2.1.2 Read Data Bytes at Higher Speed (快速读取点阵数据)

Read Data Bytes at Higher Speed 需要用指令码来执行操作。**READ_FAST** 指令的时序如下(图):

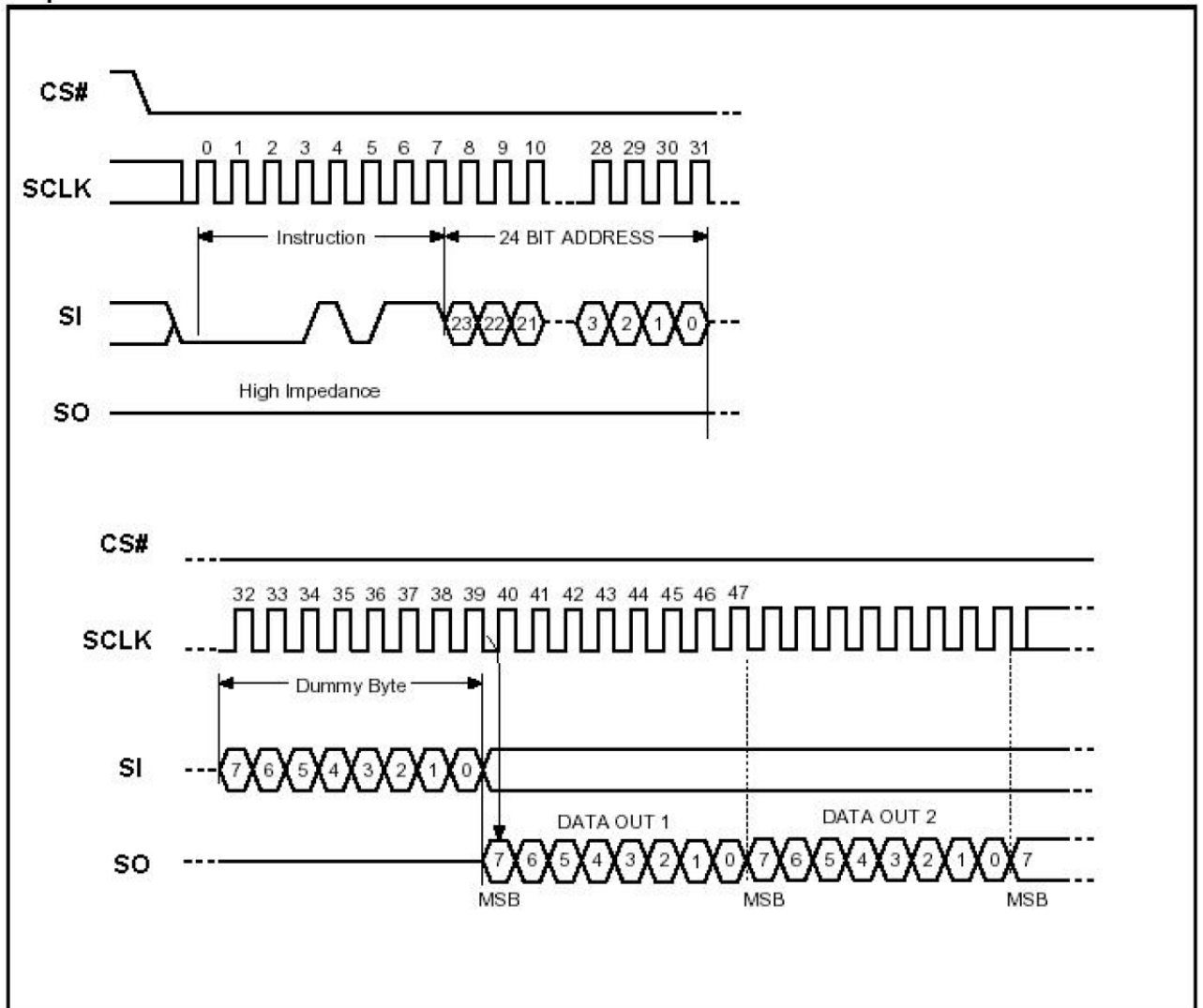
首先把片选信号 (**CS#**) 变为低, 紧跟着的是 1 个字节的命令字 (**0B h**) 和 3 个字节的地址以及一个字节 **Dummy Byte** 通过串行数据输入引脚 (**SI**) 移位输入, 每一位在串行时钟 (**SCLK**) 上升沿被锁存。

然后该地址的字节数据通过串行数据输出引脚 (**SO**) 移位输出, 每一位在串行时钟 (**SCLK**) 下降沿被移出。

如果片选信号 (**CS#**) 继续保持为底, 则下一个地址的字节数据继续通过串行数据输出引脚 (**SO**) 移位输出。例: 读取一个 **15x16** 点阵汉字需要 **32Byte**, 则连续 **32** 个字节读取后结束一个汉字的点阵数据读取操作。

如果不需要继续读取数据, 则把片选信号 (**CS#**) 变为高, 结束本次操作。

图: Read Data Bytes at Higher Speed (READ_FAST) Instruction Sequence and Data-out sequence



5.2.1 字库调用方法:

5.2.2.1 汉字点阵排列格式

每个汉字在芯片中是以汉字点阵字模的形式存储的，每个点用一个二进制位表示，存 1 的点，当显示时可以在屏幕上显示亮点，存 0 的点，则在屏幕上不显示。点阵排列格式为横置横排：即一个字节的低位表示左面的点，高位表示右面的点（如果用户按 word mode 读取点阵数据，请注意高低字节的顺序），排满一行的点后再排下一行。这样把点阵信息用来直接在显示器上按上述规则显示，则将出现对应的汉字。

5.2.2.2 11X12 点、15X16点、24X24点、32X32点汉字及5X7 点、7X8 点、6X12点、12X24 点字符、12 点阵不等宽字符、16点阵不等宽字符的排列格式：详见字库IC资料（JLX-GB2312-32S4W）的第19-26页。

5.2.2.3 汉字点阵字库地址表如下：

	字库内容	编码体系	码位范围	字符数	起始地址	参考算法
1	11X12 点 GB2312 标准点阵字库	GB2312	A1A1-F7FE	6763+846	0000	6.3.1.1
2	15X16 点 GB2312 标准点阵字库	GB2312	A1A1-F7FE	6763+846	2C9D0	6.3.1.2
3	24X24 点 GB2312 标准点阵字库	GB2312	A1A1-F7FE	6763+846	68190	6.3.1.3
4	32X32 点 GB2312 标准点阵字库	GB2312	A1A1-F7FE	6763+846	EDF00	6.3.1.4
5	6X12 点国标扩展字符	GB2312	A1A1-ABC0	126	1DBE0C	6.3.1.5
6	6X12 点 ASCII 字符	ASCII	20~7F 96		1DBE00	6.3.2.3
7	12 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	20~7F 96		1DC400	6.3.2.7
8	12 点阵不等宽 ASCII 白正字符	ASCII	20~7F	96	1DCDC0	6.3.2.8
9	8X16 点国标扩展字符	GB2312	A1A1-ABC0	126	1DD790	6.3.1.6
10	8X16 点 ASCII 字符	ASCII	20~7F 96		1DD780	6.3.2.4
11	5X7 点 ASCII 字符	ASCII	20~7F 96		1DDF80	6.3.2.1
12	7X8 点 ASCII 字符	ASCII	20~7F 96		1DE280	6.3.2.2
13	16 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	20~7F 96		1DE580	6.3.2.9
14	16 点阵不等宽 ASCII 白正字符	ASCII	20~7F	96	1DF240	6.3.2.10
15	12X24 点国标扩展字符	GB2312	A1A1-ABC0	126	1DFF30	6.3.1.8
16	12X24 点 ASCII 字符	ASCII	20~7F 96		1DFF00	6.3.2.5
17	24 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	20~7F 96		1E22D0	6.3.2.11
18	24 点阵不等宽 ASCII 白正字符	ASCII	20~7F	96	1E3E90	6.3.2.12
19	16X32 点国标扩展字符	GB2312	A1A1-ABC0	126	1E5A90	6.3.1.9
20	16X32 点 ASCII 字符	ASCII	20~7F 96		1E5A50	6.3.2.6
21	32 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	20~7F 96		1E99D0	6.3.2.13
22	32 点阵不等宽 ASCII 白正字符	ASCII	20~7F	96	1ECA90	6.3.2.14
23	保留区				1EFB50	
29	输入法码表	GB2312			1F36F0	
32	保留区				1F7CC8	

5.2.2.4 字符在芯片中的地址计算方法：

用户只要知道字符的内码，就可以计算出该字符点阵在芯片中的地址，然后就可从该地址连续读出点阵信息用于显示。

举例说明:15X16 点 GB2312 标准点阵字库：

参数说明：

GBCode表示汉字内码。

MSB 表示汉字内码GBCode的高8bits。

LSB 表示汉字内码GBCode 的低8bits。

Address 表示汉字或ASCII字符点阵在芯片中的字节地址。

BaseAdd：说明点阵数据在字库芯片中的起始地址。

计算方法：

BaseAdd=0x2C9D0;

if(MSB >=0xA1 && MSB <= 0xA9 && LSB >=0xA1)

Address = ((MSB - 0xA1) * 94 + (LSB - 0xA1))*32+ BaseAdd;

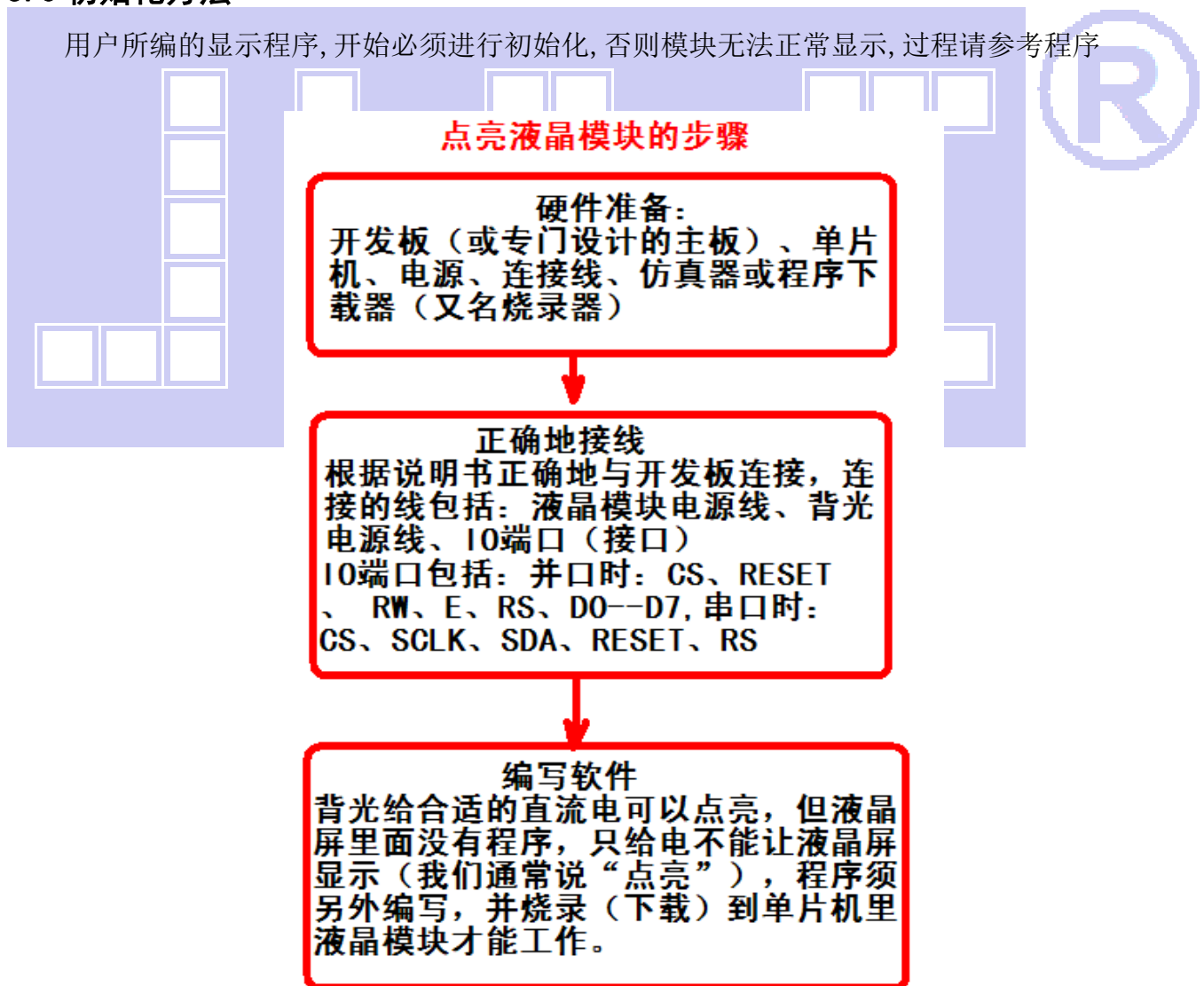
else if(MSB >=0xB0 && MSB <= 0xF7 && LSB >=0xA1)

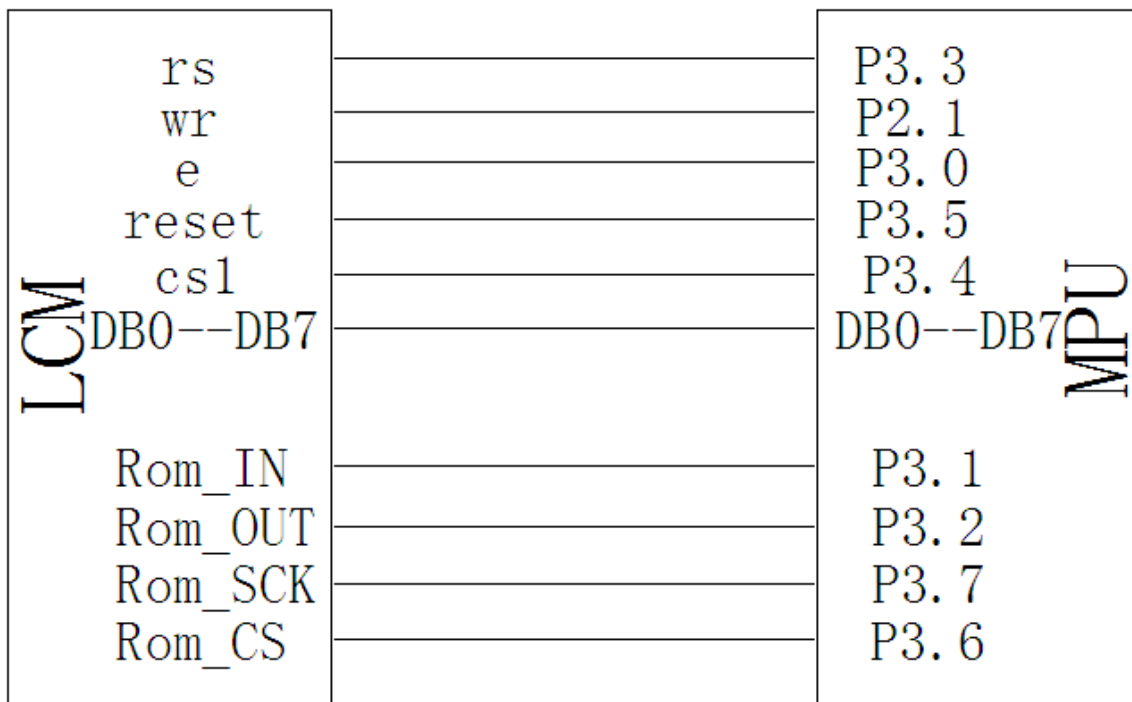
Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 846)*32+ BaseAdd;

详见字库IC资料（JLX-GB2312-3205） 的第15-31页。

5.3 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





5.4 程序举例:

/* Test program for JLX160160G-161-PL, 并行接口, 6800 时序

字库 IC 是: JLXGB2312-3205

驱动 IC 是: UC1698(or compatible)

晶联讯电子: 网址 <http://www.jlxlcd.cn>;

*/

```
#include <STC15F2K60S2.H>
```

```
sbit rs=P3^3;    /*接口定义: rs 就是 LCD 的 RS*/
sbit rd=P3^0;    /*接口定义:rd (e) 就是 LCD 的 E*/
sbit wr=P2^1;    /*接口定义:wr 就是 LCD 的 WR*/
sbit reset=P3^5; /*接口定义: reset 就是 LCD 的 RST*/
sbit cs1=P3^4;   /*接口定义: cs1 就是 LCD 的 CS*/
sbit key=P2^0;   //我司主板按键接口, 可用延时代替
```

```
sbit Rom_IN    = P3^1; //字库 IC 接口定义:Rom_IN 就是字库 IC 的 SI
sbit Rom_OUT   = P3^2; //字库 IC 接口定义:Rom_OUT 就是字库 IC 的 SO
sbit Rom_SCK   = P3^7; //字库 IC 接口定义:Rom_SCK 就是字库 IC 的 SCK
sbit Rom_CS    = P3^6; //字库 IC 接口定义 Rom_CS 就是字库 IC 的 CS#
```

```
#define uchar unsigned char
```

```
#define uint unsigned int
```

```
#define ulong unsigned long
```

```
//=====delay time=====
```

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

//-----wait a switch,jump out if P2.0 get a signal"0"-----
```

```
void waitkey()
{
    repeat:
        if (key==1) goto repeat;
        else delay(3000);
}
```

```
//=====transfer command to LCM=====
```

```
void transfer_command(int data1)
```

```
{
    cs1=0;
    rs=0;
    rd=0;
    wr=0;
    P1=data1;
    rd=1;
    cs1=1;
    rd=0;
    P1=0; //此条非必要
}
```



```
//-----transfer data to LCM-----
```

```
void transfer_data(int data1)
```

```
{
    cs1=0;
    rs=1;
    rd=0;
    wr=0;
    P1=data1;
    rd=1;
    cs1=1;
    rd=0;
    P1=0; //此条非必要
}
```

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

```
void initial_lcd()
```

```
{
```

```

reset=0;                //低电平复位
delay(500);
reset=1;                //复位完毕
delay(500);
transfer_command(0xe2); //软件复位
delay(500);            //延时
transfer_command(0x25); //设置温度补偿系数: TC1TC0 01=-0.05%
transfer_command(0x2b); //internal pump:0x2b
transfer_command(0xc4); //设置 LCD 翻转方法
transfer_command(0xa3); //set line rate
transfer_command(0xd1); //set color pattem
transfer_command(0xd6); //设置为 64 彩色*/
transfer_command(0xeb); //设置偏压比: 1/10bias:0xea 1/12bias:0xeb
transfer_command(0x81); //set Vbias potentiometer
transfer_command(0x8c); //对比度设置
transfer_command(0xf1); //设置行数 0xf1
transfer_command(0x9f); //设置行数的数值为 80 行:4f,160 行: 9f
transfer_command(0xad); //打开显示
    
```

```

}

/*写液晶显示的地址: y 是行, x 是列*/
void set_lcd_address(int y,int x)
{
    int row_msb,row_lsb,column,column_msb,column_lsb;
    row_lsb = y&0x0f;
    row_msb = ((y>>4)&0x0f);
    transfer_command(0x60+row_lsb); //行地址低 4 位
    transfer_command(0x70+row_msb); //行地址高 4 位
    column=x+37;
    column_lsb =column&0x0f;
    column_msb =((column>>4)&0x0f);
    transfer_command(0x00+column_lsb); //列地址低 4 位
    transfer_command(0x10+column_msb); //列地址高 4 位
}
    
```

```

void clear_screen()
{
    int i,j;
    set_lcd_address(0,0);
    for(i=0;i<160;i++)
    {
        for(j=0;j<128;j++)
        {
            transfer_data(0x00);
            transfer_data(0x00);
        }
    }
}
    
```



```

    }
}

//=====
/*因为这款 IC 原是由于彩色 STN 液晶屏上的，每个像素点分为 R\G\B(红绿蓝) 三个点阵，
现在用的是单色屏，所以每 3 个点阵还是要依照：
R4R3R2R1R0（第 1 个点阵） G5G4G3G2G1G0（第二个点阵） B4B3B2B1B0（第三个点阵），
共 16 位数据来表达*/
void display_3pixels(unsigned long int mono_data)
{
    int color_data;
    if(mono_data&0x80000000) //bit 15, 12, 9, 6, 3
        {color_data=0xf800;} //0xf800 表示红色
    else
        {color_data=0x0000;}
    if(mono_data&0x40000000) //bit 14, 11, 8, 5, 2
        {color_data=color_data+0x07e0;} //0x07e0 表示绿色
    else;
    if(mono_data&0x20000000) //bit 13, 10, 7, 4, 1
        {color_data=color_data+0x001f;} //0x001f 表示蓝色
    else;
    transfer_data(color_data>>8); //writting 3dots
    transfer_data(color_data);
}

//显示 32*32 的字体
/*y 是行地址，x 是列地址，dp 是数据指针*/
void display_char32x32(int y,int x,uchar *dp)
{
    int i,j,k;
    int row;
    unsigned long int mono_data;
    for(i=0;i<32;i++)
    {
        row=y;
        row=row+i;
        set_lcd_address(row,x);
        mono_data=0x00000000;
        for(k=0;k<4;k++) //4*8=32
        {
            mono_data<<=8;
            mono_data+=*dp;
            dp++;
        }

        for(j=0;j<11;j++) //11 次*3 个点阵=33 个点阵
    }
}

```



```

        {
            display_3pixels(mono_data);
            mono_data=mono_data<<3;
        }
    }
}

```

/*写一个 16x32 点阵的汉字或图片*/

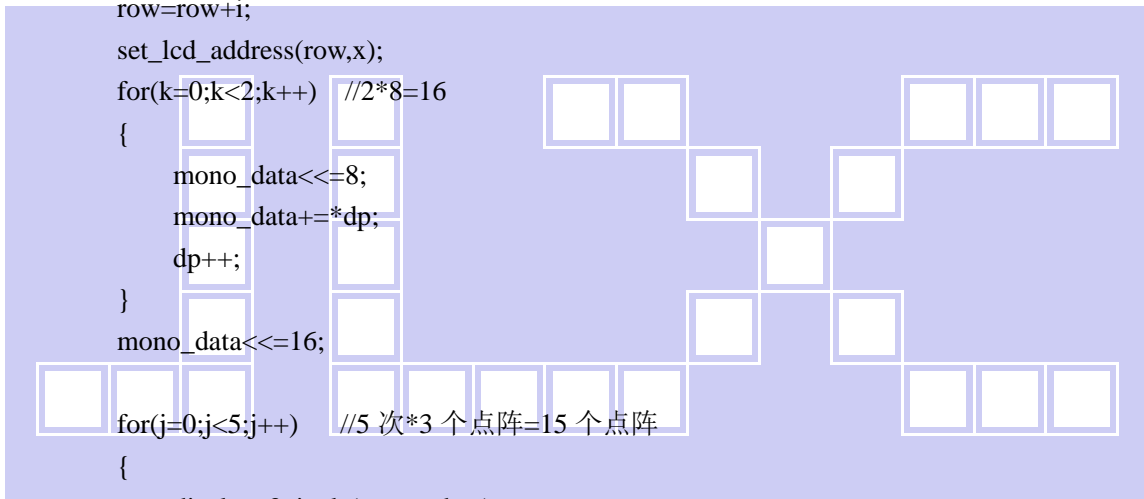
/*y 是行地址, x 是列地址, dp 是数据指针*/

void display_char16x32(int y,int x,uchar *dp)

```

{
    int i,j,k;
    int row;
    unsigned long int mono_data;
    for(i=0;i<32;i++)
    {
        row=y;
        row=row+i;
        set_lcd_address(row,x);
        for(k=0;k<2;k++) //2*8=16
        {
            mono_data<<=8;
            mono_data+=*dp;
            dp++;
        }
        mono_data<<=16;
        for(j=0;j<5;j++) //5次*3个点阵=15个点阵
        {
            display_3pixels(mono_data);
            mono_data=mono_data<<3;
        }
    }
}

```



//显示 24*24 的字体

/*y 是行地址, x 是列地址, dp 是数据指针*/

void display_char24x24(int y,int x,uchar *dp)

```

{
    int i,j,k;
    int row;
    unsigned long int mono_data;
    for(i=0;i<24;i++)
    {
        row=y;
        row=row+i;

```

```

set_lcd_address(row,x);
mono_data=0x00000000;
for(k=0;k<3;k++) //3*8=24
{
    mono_data<<=8;
    mono_data+=*dp;
    dp++;
}
mono_data<<=8;
for(j=0;j<8;j++) //8次*3个点阵=24个点阵
{
    display_3pixels(mono_data);
    mono_data=mono_data<<3;
}
}
}

```

/*写一个 12x24 点阵的汉字或图片*/

/*y 是行地址, x 是列地址, dp 是数据指针*/

```
void display_char12x24(int y,int x,uchar *dp)
```

```

{
    int i,j,k;
    int row;
    unsigned long int mono_data;
    for(i=0;i<24;i++)
    {
        row=y;
        row=row+i;
        set_lcd_address(row,x);
        mono_data=0x00000000;
        for(k=0;k<2;k++) //2*8=16
        {
            mono_data<<=8;
            mono_data+=*dp;
            dp++;
        }
        mono_data<<=16;

        for(j=0;j<4;j++) //4次*3个点阵=12个点阵
        {
            display_3pixels(mono_data);
            mono_data=mono_data<<3;
        }
    }
}
}

```



/*写一个 15x16 点阵的汉字或图片，因为此液晶驱动 IC 是每 3 列是一个地址，所以显示格式是 15 列（3X5=15）*/

/*y 是行地址，x 是列地址，dp 是数据指针*/

```
void display_char16x16(int y,int x,uchar *dp)
```

```
{
    int i,j,k;
    int row;
    unsigned long int mono_data;
    for(i=0;i<16;i++)
    {
        row=y;
        row=row+i;
        set_lcd_address(row,x);
        mono_data=0x00000000;
        for(k=0;k<2;k++)
        {
            mono_data<<=8;
            mono_data+=*dp;
            dp++;
        }
        mono_data<<=16;
        for(j=0;j<5;j++) //5 次*3 个点阵=15 个点阵
        {
            display_3pixels(mono_data);
            mono_data=mono_data<<3;
        }
    }
}
```



/*写一个 8x16 点阵的汉字或图片*/

/*y 是行地址，x 是列地址，dp 是数据指针*/

```
void display_char8x16(int y,int x,uchar *dp)
```

```
{
    int i,j;
    int row;
    unsigned long int mono_data;
    for(i=0;i<16;i++)
    {
        row=y;
        row=row+i;
        set_lcd_address(row,x);
        mono_data=0x00000000;

        mono_data<<=8;
        mono_data+=*dp;
```

```

dp++;
mono_data<<=24;

for(j=0;j<3;j++) //5次*3个点阵=15个点阵
{
    display_3pixels(mono_data);
    mono_data=mono_data<<3;
}
}
}

```

//显示 12*12 的字体

/*y 是行地址, x 是列地址, dp 是数据指针*/

void display_char12x12(int y,int x,uchar *dp)

```

{
    int i,j,k;
    int row;
    unsigned long int mono_data;
    for(i=0;i<12;i++)
    {
        row=y;
        row=row+i;
        set_lcd_address(row,x);
        mono_data=0x00000000;
        for(k=0;k<2;k++)
        {
            mono_data<<=8;
            mono_data+=*dp;
            dp++;
        }
        mono_data<<=16;
    }

```

```

for(j=0;j<4;j++) //4次*3个点阵=12个点阵
{
    display_3pixels(mono_data);
    mono_data=mono_data<<3;
}
}
}

```

//显示 6*12 的字体

/*y 是行地址, x 是列地址, dp 是数据指针*/

void display_char6x12(int y,int x,uchar *dp)

```

{
    int i,j;
    int row;

```



```

unsigned long int mono_data;
for(i=0;i<12;i++)
{
    row=y;
    row=row+i;
    set_lcd_address(row,x);
    mono_data=0x00000000;

    mono_data<<=8;
    mono_data+=*dp;
    dp++;
    mono_data<<=24;

    for(j=0;j<2;j++) //2次*3个点阵=6个点阵
    {
        display_3pixels(mono_data);
        mono_data=mono_data<<3;
    }
}

```

//显示 5*7 的字体

/*y 是行地址, x 是列地址, dp 是数据指针*/

void display_char5x7(int y,int x,uchar *dp)

```

{
    int i,j;
    int row;
    unsigned long int mono_data;
    for(i=0;i<7;i++)
    {
        row=y;
        row=row+i;
        set_lcd_address(row,x);
        mono_data=0x00000000;

        mono_data<<=8;
        mono_data+=*dp;
        dp++;
        mono_data<<=24;

        for(j=0;j<2;j++) //2次*3个点阵=6个点阵
        {
            display_3pixels(mono_data);
            mono_data=mono_data<<3;
        }
    }
}

```



```

}

/****送指令到晶联讯字库 IC****/
void send_command_to_ROM( uchar datu )
{
    uchar i;
    for(i=0;i<8;i++ )
    {
        if(datu&0x80)
            Rom_IN = 1;
        else
            Rom_IN = 0;
        datu = datu<<1;
        Rom_SCK=0;
        Rom_SCK=1;
    }
}

```

****从晶联讯字库 IC 中取汉字或字符数据 (1 个字节) ****/

```

static uchar get_data_from_ROM()
{
    uchar i;
    uchar ret_data=0;
    Rom_SCK=1;
    for(i=0;i<8;i++)
    {
        Rom_OUT=1;
        Rom_SCK=0;
        ret_data=ret_data<<1;
        if( Rom_OUT )
            ret_data=ret_data+1;
        else
            ret_data=ret_data+0;
        Rom_SCK=1;
    }
    return(ret_data);
}

```

/*从相关地址 (addrHigh: 地址高字节,addrMid: 地址中字节,addrLow: 地址低字节) 中连续读出 DataLen 个字节的数据到 pBuff 的地址*/

/*连续读取*/

```

void get_n_bytes_data_from_ROM(uchar addrHigh,uchar addrMid,uchar addrLow,uchar *pBuff,uchar DataLen )
{
    uchar i;
    Rom_CS = 0;
    cs1=1;

```



```

Rom_SCK=0;
send_command_to_ROM(0x0b);
send_command_to_ROM(addrHigh);
send_command_to_ROM(addrMid);
send_command_to_ROM(addrLow);
send_command_to_ROM(0x00);
for(i = 0; i < DataLen; i++ )
    *(pBuff+i) =get_data_from_ROM();
Rom_CS = 1;
}

/*****
ulong fontaddr=0;
void display_GB2312_12x12_string(uchar y,uchar x,uchar *text)
{
    uchar i= 0;
    uchar addrHigh,addrMid,addrLow ;
    uchar fontbuf[24];
    while((text[i]>0x00))
    {
        if(((text[i]>=0xb0) &&(text[i]<=0xf7))&&(text[i+1]>=0xa1))
        {
            /*国标简体 (GB2312) 汉字在晶联讯字库 IC 中的地址由以下公式来计算: */
            /*Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 846)*32+ BaseAdd;BaseAdd=0*/
            /*由于担心 8 位单片机有乘法溢出问题, 所以分三部取地址*/
            fontaddr = (text[i]- 0xb0)*94;
            fontaddr += (text[i+1]-0xa1)+846;
            fontaddr = (ulong)(fontaddr*24);
            fontaddr = (ulong)(fontaddr+0x00);

            addrHigh = (fontaddr&0xff0000)>>16; /*地址的高 8 位,共 24 位*/
            addrMid = (fontaddr&0xff00)>>8; /*地址的中 8 位,共 24 位*/
            addrLow = fontaddr&0xff; /*地址的低 8 位,共 24 位*/
            get_n_bytes_data_from_ROM(addrHigh,addrMid,addrLow,fontbuf,24 );/*取 32 个字节的数据, 存到
            "fontbuf[32]"*/
            display_char12x12(y,x,fontbuf);/*显示汉字到 LCD 上, y 为页地址, x 为列地址, fontbuf[]为数据*/
            i+=2;
            x+=4;
        }
        else if(((text[i]>=0xa1) &&(text[i]<=0xa3))&&(text[i+1]>=0xa1))
        {
            /*国标简体 (GB2312) 15x16 点的全角标点符号等等在晶联讯字库 IC 中的地址由以下公式来计算:
            */
            /*Address = ((MSB - 0xa1) * 94 + (LSB - 0xA1))*32+ BaseAdd;BaseAdd=0*/
            /*由于担心 8 位单片机有乘法溢出问题, 所以分三部取地址*/
            fontaddr = (text[i]- 0xa1)*94;

```



```

fontaddr += (text[i+1]-0xa1);
fontaddr = (ulong)(fontaddr*24);
fontaddr = (ulong)(fontaddr+0x00);

addrHigh = (fontaddr&0xff0000)>>16; /*地址的高 8 位,共 24 位*/
addrMid = (fontaddr&0xff00)>>8; /*地址的中 8 位,共 24 位*/
addrLow = fontaddr&0xff; /*地址的低 8 位,共 24 位*/
get_n_bytes_data_from_ROM(addrHigh,addrMid,addrLow,fontbuf,24 );/*取 32 个字节的数
据, 存到
"fontbuf[32]"*/
display_char12x12(y,x,fontbuf);/*显示汉字到 LCD 上, y 为页地址, x 为列地址,
fontbuf[]为数据*/
i+=2;
x+=4;
}
else if((text[i]>=0x20) &&(text[i]<=0x7e))
{
    unsigned char fontbuf[12];
    fontaddr = (text[i]- 0x20);
    fontaddr = (unsigned long)(fontaddr*12);
    fontaddr = (unsigned long)(fontaddr+0x1dbe00);

    addrHigh = (fontaddr&0xff0000)>>16;
    addrMid = (fontaddr&0xff00)>>8;
    addrLow = fontaddr&0xff;

    get_n_bytes_data_from_ROM(addrHigh,addrMid,addrLow,fontbuf,12 );/*取 16 个
字节的数据, 存到
"fontbuf[32]"*/
    display_char6x12(y,x,fontbuf);/*显示 8x16 的 ASCII 字到 LCD 上, y 为页地
址, x 为列地址, fontbuf[]
为数据*/
    i+=1;
    x+=2;
}
else
    i++;
}
}

```

```

void display_GB2312_16x16_string(uchar y,uchar x,uchar *text)
{
    uchar i= 0;
    uchar addrHigh,addrMid,addrLow ;
    uchar fontbuf[32];
    while((text[i]>0x00))
    {
        if(((text[i]>=0xb0) &&(text[i]<=0xf7))&&(text[i+1]>=0xa1))
        {

```

```

/*国标简体 (GB2312) 汉字在晶联讯字库 IC 中的地址由以下公式来计算: */
/*Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1) + 846) * 32 + BaseAdd; BaseAdd=0*/
/*由于担心 8 位单片机有乘法溢出问题, 所以分三部取地址*/
fontaddr = (text[i] - 0xb0) * 94;
fontaddr += (text[i+1] - 0xa1) + 846;
fontaddr = (ulong)(fontaddr * 32);
fontaddr = (ulong)(fontaddr + 0x2c9d0);

addrHigh = (fontaddr & 0xff0000) >> 16; /*地址的高 8 位, 共 24 位*/
addrMid = (fontaddr & 0xff00) >> 8; /*地址的中 8 位, 共 24 位*/
addrLow = fontaddr & 0xff; /*地址的低 8 位, 共 24 位*/
get_n_bytes_data_from_ROM(addrHigh, addrMid, addrLow, fontbuf, 32); /*取 32 个字节的数
据, 存到
"fontbuf[32]"*/
display_char16x16(y, x, fontbuf); /*显示汉字到 LCD 上, y 为页地址, x 为列地址,
fontbuf[] 为数据*/
i += 2;
x += 5;
}
else if (((text[i] >= 0xa1) && (text[i] <= 0xa3)) && (text[i+1] >= 0xa1))
{
/*国标简体 (GB2312) 15x16 点的全角标点符号等等在晶联讯字库 IC 中的地址由以下公式来计算:
*/
/*Address = ((MSB - 0xa1) * 94 + (LSB - 0xA1)) * 32 + BaseAdd; BaseAdd=0*/
/*由于担心 8 位单片机有乘法溢出问题, 所以分三部取地址*/
fontaddr = (text[i] - 0xa1) * 94;
fontaddr += (text[i+1] - 0xa1);
fontaddr = (ulong)(fontaddr * 32);
fontaddr = (ulong)(fontaddr + 0x2c9d0);

addrHigh = (fontaddr & 0xff0000) >> 16; /*地址的高 8 位, 共 24 位*/
addrMid = (fontaddr & 0xff00) >> 8; /*地址的中 8 位, 共 24 位*/
addrLow = fontaddr & 0xff; /*地址的低 8 位, 共 24 位*/
get_n_bytes_data_from_ROM(addrHigh, addrMid, addrLow, fontbuf, 32); /*取 32 个字节的数
据, 存到
"fontbuf[32]"*/
display_char16x16(y, x, fontbuf); /*显示汉字到 LCD 上, y 为页地址, x 为列地址,
fontbuf[] 为数据*/
i += 2;
x += 5;
}
else if ((text[i] >= 0x20) && (text[i] <= 0x7e))
{
unsigned char fontbuf[16];
fontaddr = (text[i] - 0x20);
fontaddr = (unsigned long)(fontaddr * 16);
fontaddr = (ulong)(fontaddr + 0x1dd780);
addrHigh = (fontaddr & 0xff0000) >> 16;
addrMid = (fontaddr & 0xff00) >> 8;
addrLow = fontaddr & 0xff;

```



```

        get_n_bytes_data_from_ROM(addrHigh,addrMid,addrLow,fontbuf,16 );/*取 16 个字节的数据，存到
"fontbuf[32]"*/

        display_char8x16(y,x,fontbuf);/*显示 8x16 的 ASCII 字到 LCD 上，y 为页地址，x 为列地址，fontbuf[]
为数据*/

        i+=1;
        x+=3;
    }
    else
        i++;
    }
}

void display_GB2312_24x24_string(uchar y,uchar x,uchar *text)
{
    uchar i= 0;
    uchar addrHigh,addrMid,addrLow ;
    uchar fontbuf[72];
    while((text[i]>0x00))
    {
        if(((text[i]>=0xb0) &&(text[i]<=0xf7))&&(text[i+1]>=0xa1))
        {
            /*国标简体（GB2312）汉字在晶联讯字库 IC 中的地址由以下公式来计算：*/
            /*Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1)+ 846)*32+ BaseAdd;BaseAdd=0*/
            /*由于担心 8 位单片机有乘法溢出问题，所以分三部取地址*/
            fontaddr = (text[i]- 0xb0)*94;
            fontaddr += (text[i+1]-0xa1)+846;
            fontaddr = (ulong)(fontaddr*72);
            fontaddr = (ulong)(fontaddr+0X068190);

            addrHigh = (fontaddr&0xff0000)>>16; /*地址的高 8 位,共 24 位*/
            addrMid = (fontaddr&0xff00)>>8;      /*地址的中 8 位,共 24 位*/
            addrLow = fontaddr&0xff;          /*地址的低 8 位,共 24 位*/
            get_n_bytes_data_from_ROM(addrHigh,addrMid,addrLow,fontbuf,72);/*取 32 个字节的数据，存到
"fontbuf[32]"*/

            display_char24x24(y,x,fontbuf);/*显示汉字到 LCD 上，y 为页地址，x 为列地址，fontbuf[]为数据*/
            i+=2;
            x+=8;
        }
        else if(((text[i]>=0xa1) &&(text[i]<=0xa3))&&(text[i+1]>=0xa1))
        {
            /*国标简体（GB2312）15x16 点的全角标点符号等等在晶联讯字库 IC 中的地址由以下公式来计算：
*/

            /*Address = ((MSB - 0xa1) * 94 + (LSB - 0xA1))*32+ BaseAdd;BaseAdd=0*/
            /*由于担心 8 位单片机有乘法溢出问题，所以分三部取地址*/

```




```

fontaddr = (text[i]- 0xa1)*94;
fontaddr += (text[i+1]-0xa1);
fontaddr = (ulong)(fontaddr*72);
fontaddr = (ulong)(fontaddr+0X068190);

addrHigh = (fontaddr&0xff0000)>>16; /*地址的高 8 位,共 24 位*/
addrMid = (fontaddr&0xff00)>>8; /*地址的中 8 位,共 24 位*/
addrLow = fontaddr&0xff; /*地址的低 8 位,共 24 位*/
get_n_bytes_data_from_ROM(addrHigh,addrMid,addrLow,fontbuf,72 );/*取 32 个字节的数
据, 存到
"fontbuf[32]"*/
display_char24x24(y,x,fontbuf);/*显示汉字到 LCD 上, y 为页地址, x 为列地址,
fontbuf[]为数据*/
i+=2;
x+=8;
}
else if((text[i]>=0x20) &&(text[i]<=0x7e))
{
    unsigned char fontbuf[48];
    fontaddr = (text[i]- 0x20);
    fontaddr = (unsigned long)(fontaddr*48);
    fontaddr = (ulong)(fontaddr+0x1dff00);
    addrHigh = (fontaddr&0xff0000)>>16;
    addrMid = (fontaddr&0xff00)>>8;
    addrLow = fontaddr&0xff;

    get_n_bytes_data_from_ROM(addrHigh,addrMid,addrLow,fontbuf,48 );/*取 16 个
字节的数据, 存到
"fontbuf[32]"*/
    display_char12x24(y,x,fontbuf);/*显示 8x16 的 ASCII 字到 LCD 上, y 为页
地址, x 为列地址, fontbuf[]
为数据*/
    i+=1;
    x+=4;
}
else
    i++;
}
}

```

```

void display_GB2312_32x32_string(uchar y,uchar x,uchar *text)
{
    uchar i= 0;
    uchar addrHigh,addrMid,addrLow ;
    uchar fontbuf[128];
    while((text[i]>0x00))
    {
        if(((text[i]>=0xb0) &&(text[i]<=0xf7))&&(text[i+1]>=0xa1))
        {

```

```

/*国标简体 (GB2312) 汉字在晶联讯字库 IC 中的地址由以下公式来计算: */
/*Address = ((MSB - 0xB0) * 94 + (LSB - 0xA1) + 846) * 32 + BaseAdd; BaseAdd=0*/
/*由于担心 8 位单片机有乘法溢出问题, 所以分三部取地址*/
fontaddr = (text[i] - 0xb0) * 94;
fontaddr += (text[i+1] - 0xa1) + 846;
fontaddr = (ulong)(fontaddr * 128);
fontaddr = (ulong)(fontaddr + 0Xedf00);

addrHigh = (fontaddr & 0xff0000) >> 16; /*地址的高 8 位, 共 24 位*/
addrMid = (fontaddr & 0xff00) >> 8; /*地址的中 8 位, 共 24 位*/
addrLow = fontaddr & 0xff; /*地址的低 8 位, 共 24 位*/
get_n_bytes_data_from_ROM(addrHigh, addrMid, addrLow, fontbuf, 128); /*取 32 个字节的数据, 存到
"fontbuf[32]"*/
display_char32x32(y, x, fontbuf); /*显示汉字到 LCD 上, y 为页地址, x 为列地址, fontbuf[] 为数据*/
i += 2;
x += 11;
}
else if(((text[i] >= 0xa1) && (text[i] <= 0xa3)) && (text[i+1] >= 0xa1))
{
/*国标简体 (GB2312) 15x16 点的全角标点符号等等在晶联讯字库 IC 中的地址由以下公式来计算:
*/
/*Address = ((MSB - 0xa1) * 94 + (LSB - 0xA1)) * 32 + BaseAdd; BaseAdd=0*/
/*由于担心 8 位单片机有乘法溢出问题, 所以分三部取地址*/
fontaddr = (text[i] - 0xa1) * 94;
fontaddr += (text[i+1] - 0xa1);
fontaddr = (ulong)(fontaddr * 128);
fontaddr = (ulong)(fontaddr + 0Xedf00);

addrHigh = (fontaddr & 0xff0000) >> 16; /*地址的高 8 位, 共 24 位*/
addrMid = (fontaddr & 0xff00) >> 8; /*地址的中 8 位, 共 24 位*/
addrLow = fontaddr & 0xff; /*地址的低 8 位, 共 24 位*/
get_n_bytes_data_from_ROM(addrHigh, addrMid, addrLow, fontbuf, 128); /*取 32 个字节的数据, 存到
"fontbuf[32]"*/
display_char32x32(y, x, fontbuf); /*显示汉字到 LCD 上, y 为页地址, x 为列地址, fontbuf[] 为数据*/
i += 2;
x += 11;
}
else if((text[i] >= 0x20) && (text[i] <= 0x7e))
{
unsigned char fontbuf[64];
fontaddr = (text[i] - 0x20);
fontaddr = (unsigned long)(fontaddr * 64);
fontaddr = (ulong)(fontaddr + 0x1e5a50);
addrHigh = (fontaddr & 0xff0000) >> 16;
addrMid = (fontaddr & 0xff00) >> 8;
addrLow = fontaddr & 0xff;

```

```

        get_n_bytes_data_from_ROM(addrHigh,addrMid,addrLow,fontbuf,64);/*取 16 个字节的数据，存到
fontbuf[32]*/

```

```

        display_char16x32(y,x,fontbuf);/*显示 8x16 的 ASCII 字到 LCD 上，y 为页地址，x 为列地址，fontbuf[]
为数据*/

```

```

        i+=1;
        x+=5;

```

```

    }
    else

```

```

        i++;
    }
}

```

```

void display_string_5x7(uchar y,uchar x,uchar *text)

```

```

{

```

```

    unsigned char i= 0;

```

```

    unsigned char addrHigh,addrMid,addrLow ;

```

```

    while((text[i]>0x00))

```

```

    {

```

```

        if((text[i]>=0x20) &&(text[i]<=0x7e))

```

```

        {

```

```

            unsigned char fontbuf[8];

```

```

            fontaddr = (text[i]- 0x20);

```

```

            fontaddr = (unsigned long)(fontaddr*8);

```

```

            fontaddr = (ulong)(fontaddr+0x1ddf80);

```

```

            addrHigh = (fontaddr&0xff0000)>>16;

```

```

            addrMid = (fontaddr&0xff00)>>8;

```

```

            addrLow = fontaddr&0xff;

```

```

        get_n_bytes_data_from_ROM(addrHigh,addrMid,addrLow,fontbuf,8);/*取 8 个字节的数据，存到

```

```

fontbuf[32]*/

```

```

        display_char5x7(y,x,fontbuf);/*显示 5x7 的 ASCII 字到 LCD 上，y 为页地址，x 为列地址，fontbuf[]
为数据*/

```

```

        i+=1;

```

```

        x+=2;

```

```

    }

```

```

    else

```

```

        i++;
    }
}

```

```

void display_box()

```

```

{

```

```

int j;
set_lcd_address(0,0);
for(j=0;j<54;j++)
{
    transfer_data(0xff);
    transfer_data(0xff);
}
for(j=1;j<160;j++)
{
    set_lcd_address(j,0);
    transfer_data(0xf8);
    transfer_data(0x00);
    set_lcd_address(j,53);
    transfer_data(0xf8);
    transfer_data(0x00);
}
set_lcd_address(159,0);
for(j=0;j<54;j++)
{
    transfer_data(0xff);
    transfer_data(0xff);
}
display_GB2312_16x16_string(1,6,"深圳市晶联讯电子");
display_GB2312_16x16_string(18,5,"JLX160160G-161 ");
display_GB2312_16x16_string(33,6," 160x160");
display_GB2312_16x16_string(31,30,"点阵");
display_GB2312_16x16_string(47,3,"可视区: ");
display_GB2312_16x16_string(49,23,"60x60mm");
display_GB2312_16x16_string(63,3,"驱动 : ");
display_GB2312_16x16_string(65,13,"IC");
display_GB2312_16x16_string(65,23,"UC1698u");
display_GB2312_16x16_string(79,1,"晶联讯的服务宗旨: ");
display_GB2312_16x16_string(95,8,"制造高品质产品");
display_GB2312_16x16_string(111,8,"做最好的液晶屏");
display_GB2312_16x16_string(127,8,"做客户信得过的企业");
display_GB2312_12x12_string(145,1,"价值源于创造, 利己须先利人");
}

//=====main program=====
void main(void)
{
    P1M1=0x00;
    P1M0=0x00;
    P2M1=0x00;
    P2M0=0x00;
    P3M1=0x00;

```



```

P3M0=0x00;
initial_lcd();
while(1)
{
    clear_screen();
    display_GB2312_32x32_string(0,0,"晶联讯电子");
    display_GB2312_32x32_string(32,0,"160160 点阵");
    display_GB2312_24x24_string(32*2,3,"晶联讯电子厂");
    display_GB2312_24x24_string(32*2+24,3,"160 160 161");
    display_GB2312_24x24_string(32*2+26,15,"X");
    display_GB2312_24x24_string(32*2+26,31,"G-");
    display_GB2312_16x16_string(32*2+24*2,2,"晶联讯专注液晶屏厂家");
    display_GB2312_12x12_string(32*2+24*2+18,4,"深圳晶联讯电子有限公司");
    display_GB2312_16x16_string(32*2+24*2+32,2,"JLX160160G-161-PL");
    waitkey();
    clear_screen();
    display_box();
    waitkey();
    clear_screen();
    display_GB2312_16x16_string(0,1,"晶联讯专注液晶屏厂家");
    display_GB2312_16x16_string(18,0,"JLX160160G-161-PL");
    display_GB2312_12x12_string(32,4,"深圳晶联讯电子有限公司");
    display_GB2312_12x12_string(44,9,"JLX160160G-161-PL");
    display_GB2312_12x12_string(56,0,"晶联讯成立于二零零四年十一");
    display_GB2312_12x12_string(68,0,"月七日, 主要生产液晶模块, ");
    display_GB2312_12x12_string(80,0,"品质至上真诚服务。");
    display_GB2312_12x12_string(92,0,"配合 JLXGB2312-3205 字库可显");
    display_GB2312_12x12_string(104,0,"示 32*32、24*24、16*16、12*");
    display_GB2312_12x12_string(116,0,"12 的中文汉字或显示 16*32、12");
    display_GB2312_12x12_string(128,0,"*24、8*16、6*12、5*7 的 ASCII");
    display_GB2312_12x12_string(142,1,"深圳市晶联讯电子专注液晶屏");
    waitkey();
    clear_screen();
    display_GB2312_32x32_string(0,1,"JLX160x160");
    display_GB2312_24x24_string(27,3,"160 160 161");
    display_GB2312_24x24_string(29,15,"X");
    display_GB2312_24x24_string(29,31,"G-");
    display_GB2312_16x16_string(50,1,"JLX160160G-161-PL");
    display_GB2312_12x12_string(63,1,"ABCDEFGHIJKLMNOPQRSTUVWXYZ");
    display_GB2312_12x12_string(73,1,"abcdefghijklmnopqrstuvwxyz");
    display_string_5x7(84,0,"ABCDEFGHIJKLMNOPQRSTUVWXYZ");
    display_string_5x7(92,0,"abcdefghijklmnopqrstuvwxyz");
    display_string_5x7(100,0,"01234567890123456789012345");
    display_string_5x7(108,0,"01234567890123456789012345");
    display_string_5x7(116,0,"01234567890123456789012345");
    display_string_5x7(124,0,"01234567890123456789012345");
}
    
```



```

display_string_5x7(132,0,"01234567890123456789012345");
display_string_5x7(140,0,"01234567890123456789012345");
display_string_5x7(148,0,"01234567890123456789012345");
waitkey();
clear_screen();
display_GB2312_16x16_string(0,0,"啊阿埃挨哎唉哀皑癌蔼");
display_GB2312_16x16_string(16*1,0,"矮艾碍爰隘鞍氨安俺按");
display_GB2312_16x16_string(16*2,0,"暗岸胺案肮昂盎凹敖熬");
display_GB2312_16x16_string(16*3,0,"翱袄鬣鬣麽麽糜麋糜");
display_GB2312_16x16_string(16*4,0,"黠黠黠黠黠黠黠黠黠");
display_GB2312_16x16_string(16*5,0,"黠黠黠黠黠黠黠黠黠");
display_GB2312_16x16_string(16*6,0,"颞颞颞颞颞颞颞颞颞");
display_GB2312_16x16_string(16*7,0,"臾臾春舄臬舄舄舄舄舄");
display_GB2312_16x16_string(16*8,0,"舄舄舄舄舄舄舄舄舄");
display_GB2312_16x16_string(16*9,0,"舄舄舄舄舄舄舄舄舄");
waitkey();
clear_screen();
display_GB2312_12x12_string(0,0,"啊阿埃挨哎唉哀皑癌蔼矮艾碍");
display_GB2312_12x12_string(12*1,0,"爰隘鞍氨安俺按暗岸胺案肮昂");
display_GB2312_12x12_string(12*2,0,"盎凹敖熬翱袄鬣鬣麽麽糜糜糜");
display_GB2312_12x12_string(12*3,0,"糜黠黠黠黠黠黠黠黠黠");
display_GB2312_12x12_string(12*4,0,"黠黠黠黠黠黠黠黠黠颞颞颞颞");
display_GB2312_12x12_string(12*5,0,"颞颞颞颞颞颞颞颞颞臾臾春舄");
display_GB2312_12x12_string(12*6,0,"舄舄舄舄舄舄舄舄舄舄舄舄");
display_GB2312_12x12_string(12*7,0,"舄舄舄舄舄舄舄舄舄舄舄舄");
display_GB2312_12x12_string(12*8,0,"羯羴羲糝糝糝糝糝糝糝糝");
display_GB2312_12x12_string(12*9,0,"糝糝糝糝糝糝糝糝糝");
display_GB2312_12x12_string(12*10,0,"翥翥翥翥翥翥翥翥翥");
display_GB2312_12x12_string(12*11,0,"翥翥翥翥翥翥翥翥翥");
display_GB2312_12x12_string(12*12,0,"酖酖酖酖酖酖酖酖酖");
waitkey();
}
}
    
```



-END-