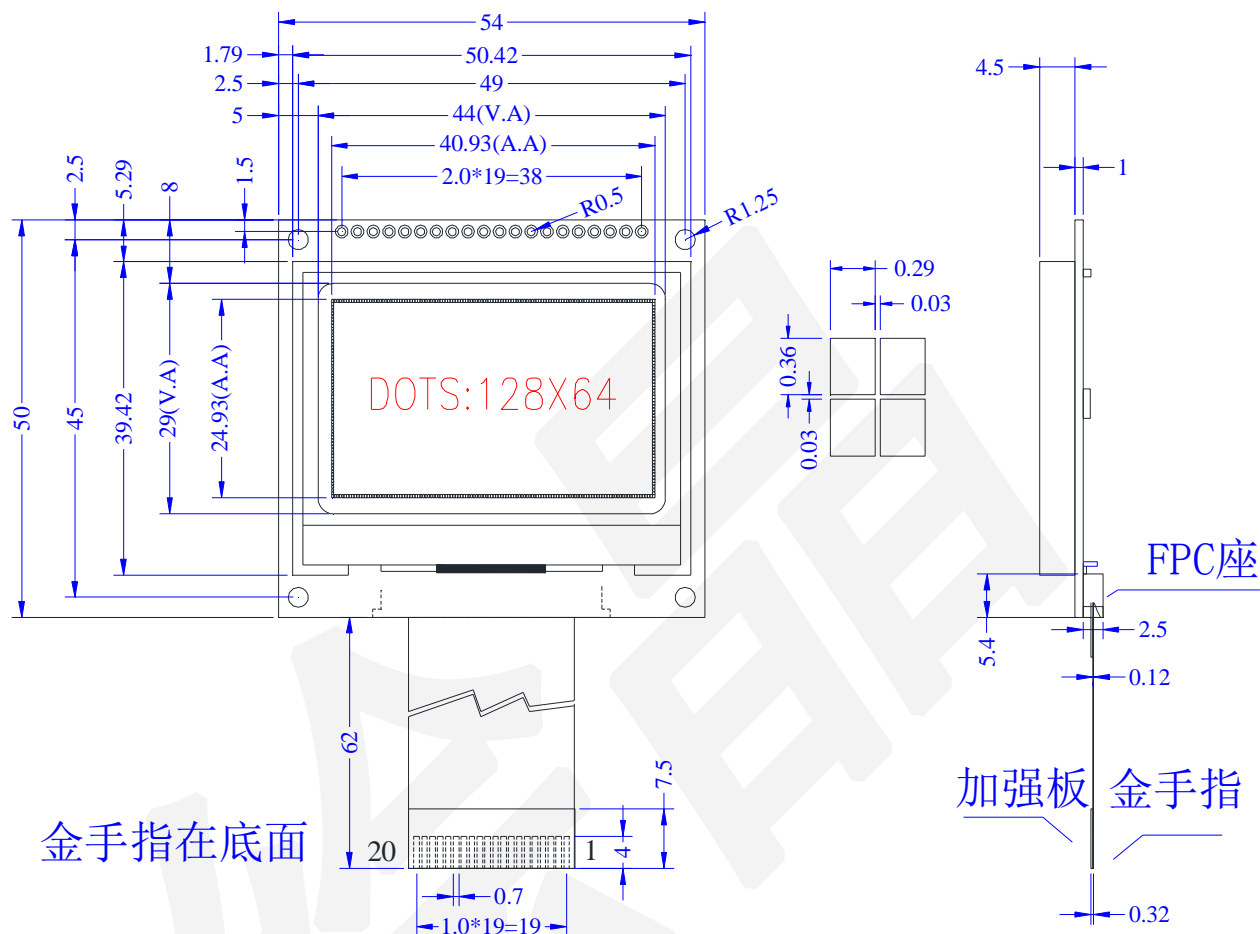


一、模块尺寸



项目	参考值
LCM 尺寸 (长×宽×厚)	54.0×50.0×8.0
可视区域 (长×宽)	44.0×29.0
点尺寸 (长×宽)	0.29×0.36
点间距 (长×宽)	0.32×0.39

二、功能介绍

- 2 工作电压 3.3V 或者 5.0V 供电, 默认为 5.0V
- 2 通讯方式: 8 位并行 (6800 和 8080 时序), 4 线 SPI 串行通讯, 默认为串口
- 2 128*64 点阵显示器内存
 - n 可自由编写图形或者汉字
 - n 竖置横排内存结构, 方便调入中文汉字, ASCII 字符
- 2 汉字和字符 (使用汉字说明用蓝字区分, 不需要字库的用户, 可以省看)
 - n 15X16 点 GB2312 标准点阵字库 6763+376 个
 - n 8X16 点国标扩展字符 126 个
- 2 ASCII 字符
 - n 5X7 点 ASCII 字符 96 个
 - n 7X8 点 ASCII 字符 96 个
 - n 8X16 点 ASCII 字符 96 个
 - n 8X16 点 ASCII 粗体字符 96 个
 - n 16 点阵不等宽 ASCII 方头 (Arial) 字符 96 个
 - n 16 点阵不等宽 ASCII 白正 (TimesNewRoman) 字符 96 个
- 2 功能指令

n 行列扫描顺序设置	起始行 页 列地址设置
n 显示开/关	反白显示
n 电源控制	待机模式
n 编压比设置	LCD 倍压设置
n 显示对比度设置	软件复位
- 2 低功率省电设计 (除背光 60MA)
 - n 正常模式 (450uA typ VDD=5.0V)
 - n 待机模式 (30uA max VDD=5.0V)
- 2 显示对比度, 内部寄存器调节 (软件调对比度)
- 2 占空比 1/64 偏压比 1/9
- 2 工作温度 -20 到 +70 度
- 2 视角 6 点钟, 显示颜色可以黄绿, 蓝白, 灰白, 可以订制。

三. 模块的接口定义

并口：PSB=1（模块电路板上 P 短接，S 空贴）

引脚	名称	方向	说明
1	RWR	I	6800 时序时：R/W=0 写操作， R/W=1 读操作
			8080 时序时：写数据端口，低电平有效
2	ERD	I	6800 时序时：使能端，高电平有效
			8080 时序时：读数据端口，低电平有效
3	VSS	—	电源负端 (0V)
4	VDD	—	电源正端 (+3.3V 或 +5.0V，出厂时设定 +5.0V)
5	LEDA	—	背光电源正端 (+3.3V 或 +5.0V，默认电压同模块电压)
6	CSB	I	使能信号，低电平有效。
7	A0 (RS)	I	数据/指令寄存器选择（高为数据，低为指令）
8	RSTB	I	低电平复位，复位完成后，回到高电平，液晶模块开始工作
9-16	DB7 ~ DB0	I/O	MPU 与模块之间并口的数据传送通道，
17	S0	O	空脚/需要带中文字库，ASCII 字符时：串行数据输出
18	SI	I	空脚/需要带中文字库，ASCII 字符时：串行数据输入
19	SCLK	I	空脚/需要带中文字库，ASCII 字符时：串行时钟输入
20	CS#	I	空脚/需要带中文字库，ASCII 字符时：片选输入

串口 PSB=0（模块电路板上 S 短接，P 空贴）

引脚	名称	方向	说明
1	RWR	—	空脚
2	ERD	—	空脚
3	VSS	—	电源负端 (0V)
4	VDD	—	电源正端 (+3.3V 或 +5.0V，出厂时设定 +5.0V)
5	LEDA	—	背光电源正端 (可选择模块电压，默认接模块 VDD)
6	CSB	I	使能信号，低电平有效。
7	A0 (RS)	I	数据/指令寄存器选择（高为数据，低为指令）
8	RSTB	I	低电平复位，复位完成后，回到高电平，液晶模块开始工作
8	SDA	I/O	显示模块串行数据
10	SCLK	I	显示模块串行时钟
11	S0	O	GB2312 串行数据输出（字符，汉字点阵数据输出）
12	SI	I	GB2312 串行数据输入（字符码，汉字码输入）
13	SCLK	I	GB2312 串行时钟输入
14	CS#	I	GB2312 片选输入
15-16	D1-D0	—	空脚
17-20	NC	—	空脚

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

www.huijinglcm.com

E-mail: huijinglcm@sina.com

四. 供电参数(直流)

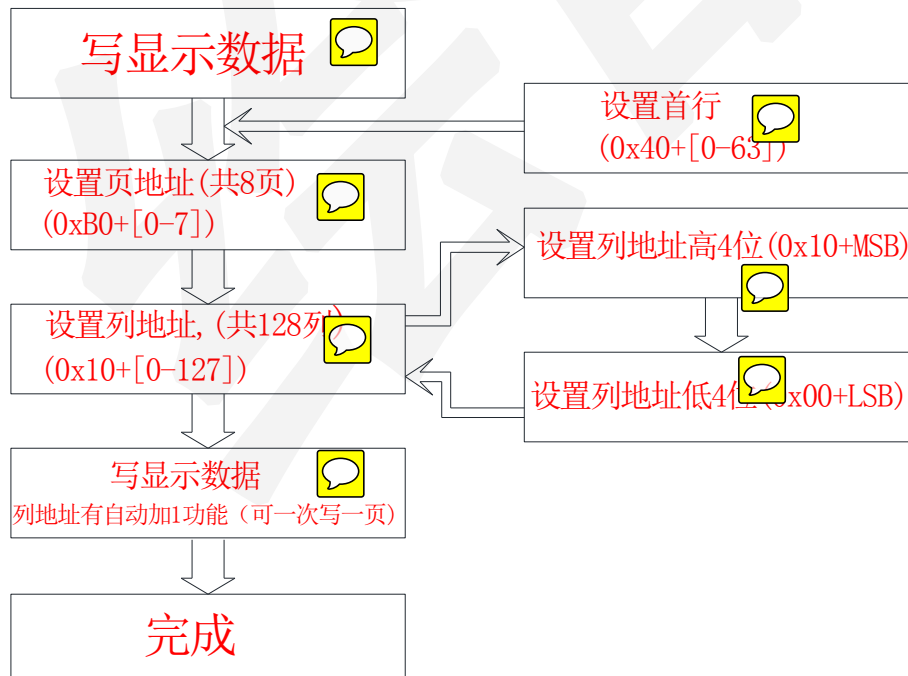
名称	符号	测试条件	参数范围			单位
			最小	标准	最大	
模块工作电压	VDD	—	4.8/2.4	5.0/3.3	5.2/3.5	V
背光工作电压	VLED	—	VDD	VDD	VDD	V
IO 输入高电平	VIH	—	0.7VDD	—	VDD	V
IO 输入低电平	VIL	—	0	—	0.6	V
LCM 输出高电平	VOH	—	0.7VDD	—	—	V
LCM 输出低电平	VOL	—	—	—	0.4	V
模块工作电流	IDD	=VDD	—	—	1.0	MA
模块待机电流	IDO	=VDD	—	—	10	uA
背光工作电流	ILED	=VLED	45	60	80	MA

五. 显示结构原理

这是一款 128*64 点阵的纯点阵屏，地址结构如图，列地址范围为（0-127），行地址并不是（0-63）行范围，而是用（0-7）页来定义的，一页占了 8 行（一页占（128*8=1024 点），所以 64 行分了 8 页，占 8 行也是为满足一个字节（上低下高），为是编程方便，可以配合字库 IC 使用，在使用取模软件的时候使用纵向取模方式，除了设置列地址和页地址外，还需要设置一个首行地址，行地址范围为（0-63），本产品显示屏的第一行设置在 COM0，首行地址设置 0 就行，假如你把首行地址设置 8，那么显示都会从第 8 行开始。

Y=	0-127列										行号
	0	1	...	62	63	64	65	...	126	127	
X=0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	0
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	7
↓	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	8
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	55
X=7	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	DB0	56
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	DB7	63

了解显示结构原理，再了解下面写程序的流程，你就能控制你想要显示的的内容，写入数据的流程图如下：



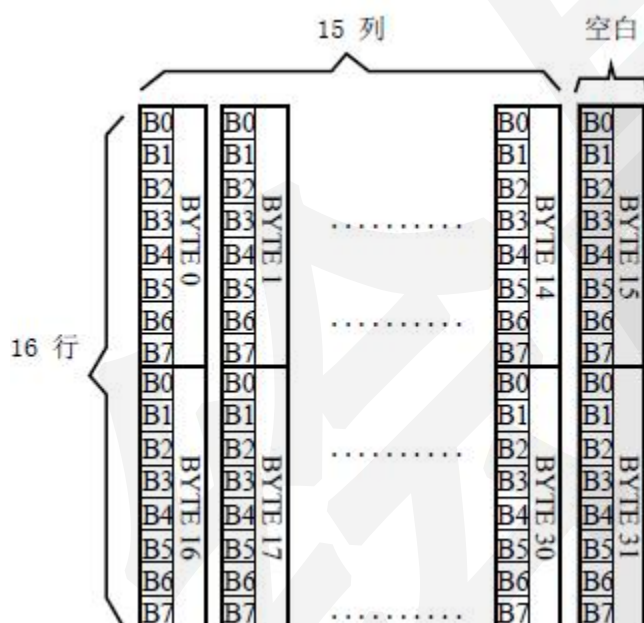
六. 字库、字符的使用方法

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。点阵排列格式为竖置横排：即一个字节的低位表示下面的点，高位表示上面的点（如果用户按16bit 总线宽度读取点阵数据，请注意高低字节的顺序），排满一行后再排下一行。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。

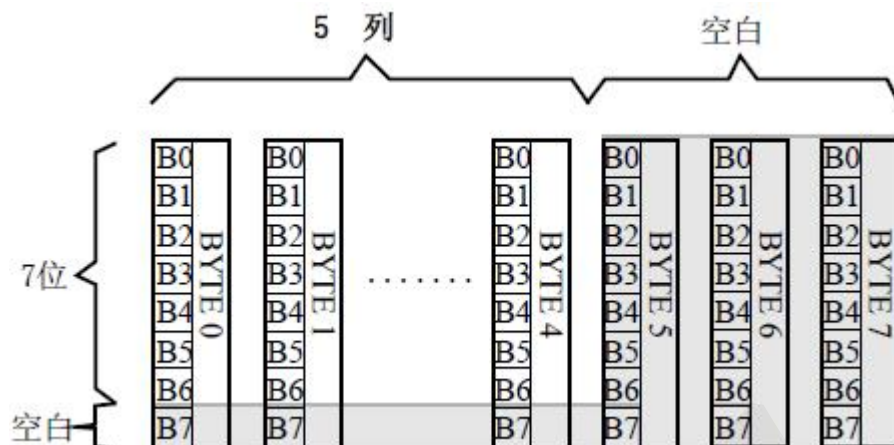
15X16 点汉字排列格式

15X16 点汉字的信息需要32 个字节（BYTE 0 - BYTE 31）来表示。该15X16 点汉字的点阵数据是竖置横排的，其具体排列结构如下图：



5X7 点 ASCII 字符排列格式

5X7 点ASCII 的信息需要8 个字节（BYTE 0 - BYTE7）来表示。该ASCII 点阵数据是竖置横排的，其具体排列结构如下图：



7X8 点 ASCII 字符排列格式

7X8 点 ASCII 的信息需要 8 个字节 (BYTE 0 - BYTE7) 来表示。该 ASCII 点阵数据是竖置横排的，其具体排列结构如下图：



8X16 点字符排列格式

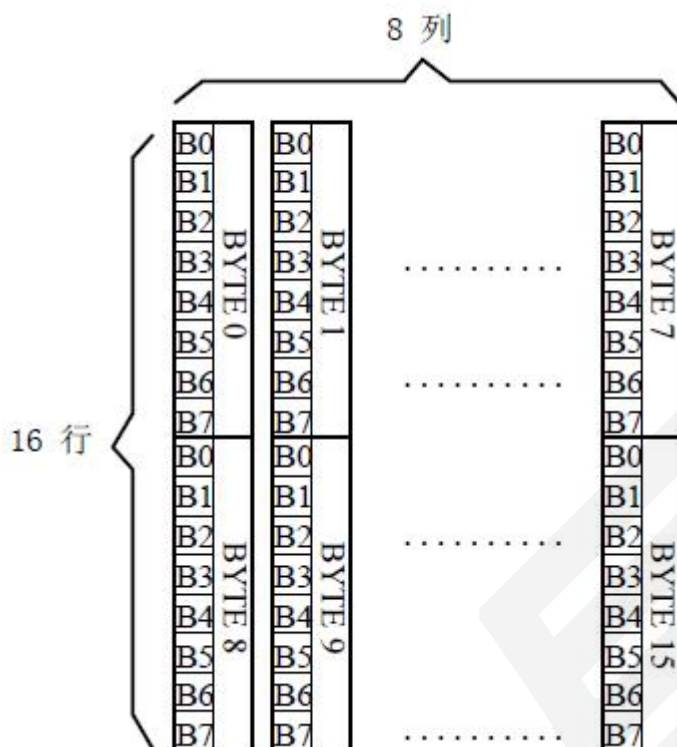
适用于此种排列格式的字体有：

8X16 点 ASCII 字符

8X16 点 ASCII 粗体字符

8X16 点 国标扩展字符

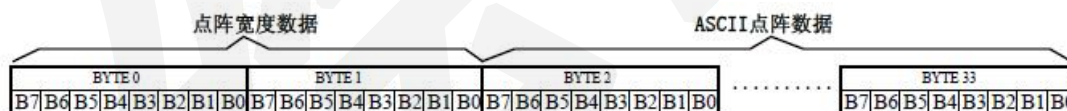
8X16 点字符信息需要 16 个字节 (BYTE 0 - BYTE15) 来表示。该点阵数据是竖置横排的，其具体排列结构如下图：



16 点阵不等宽 ASCII 方头 (Arial)、白正 (TimesNewRoman) 字符排列格式

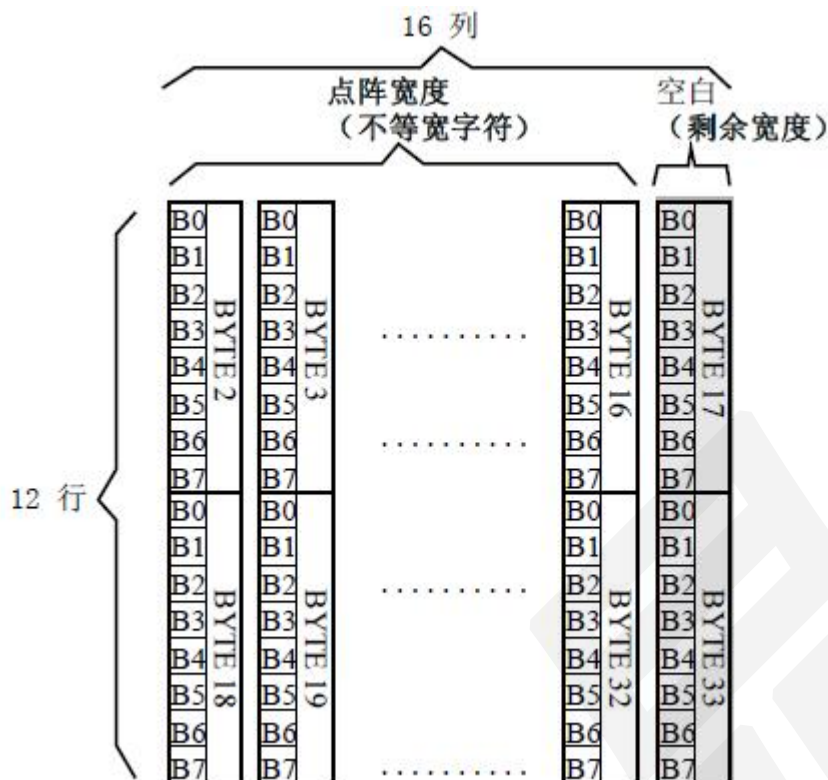
16 点阵不等宽字符的信息需要 34 个字节 (BYTE 0 - BYTE33) 来表示。

存储格式由于字符是不等宽的，因此在存储格式中 BYTE0~ BYTE1 存放点阵宽度数据，BYTE2-33 存放竖置横排点阵数据。具体格式见下图：



存储结构

不等宽字符的点阵存储宽度是以 BYTE 为单位取整的，根据不同字符宽度会出现相应的空白区。根据 BYTE0~ BYTE1 所存放点阵的实际宽度数据，可以对还原下一个字的显示或排版留作参考。



例如：ASCII 方头字符B

0-33BYTE 的点阵数据是：00 0C 00 F8 F8 18 18 18 18 18 F8 F0 00 00 00 00 00 00 00 7F 7F 63 63 63 63 63 67 3E 1C 00 00 00 00 00

其中：BYTE0~ BYTE1: 00 0C 为ASCII 方头字符B 的点阵宽度数据，即：12 位宽度。字符后面有4 位空白区，可以在排版下一个字时考虑到这一点，将下一个字的起始位置前移。

BYTE2-33: 00 F8 F8 18 18 18 18 18 F8 F0 00 00 00 00 00 00 00 7F 7F 63 63 63 63 63 67 3E 1C 00 00 00 00 00 为ASCII 方头字符B 的点阵数据。

汉字点阵字库地址表

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

www.huijinglcm.com

E-mail: huijinglcm@sina.com

	字库内容	编码体系	码位范围	字符数	起始地址	结束地址	参考算法
1	15X16 点 GB2312 标准点阵字库	GB2312	A1A1-F7FE	6763+376	00000	3B7BF	6.3.1.1
2	7X8 点 ASCII 字符	ASCII	20~7F	96	66C0	69BF	6.3.2.2
3	8X16 点国标扩展字符	GB2312	AAA1-AB0	126	3B7D0	3BFBF	6.3.1.2
4	8X16 点 ASCII 字符	ASCII	20~7F	96	3B7C0	3BFBF	6.3.2.3
5	5X7 点 ASCII 字符	ASCII	20~7F	96	3BFC0	3C2BF	6.3.2.1
6	16 点阵不等宽 ASCII 方头 (Arial) 字符	ASCII	20~7F	96	3C2C0	3CF7F	6.3.2.4
7	8X16 点 ASCII 粗体字符	ASCII	20~7F	96	3CF80	3D57F	6.3.2.5
8	16 点阵不等宽 ASCII 白正 (TimesNewRoman) 字符	ASCII	20~7F	96	3D580	3E23F	6.3.2.6

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。

汉字字符的地址计算

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=0;

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

Address = (282 + (LSB - 0xA1)) * 32 + BaseAdd;

else if (MSB >= 0xA1 && MSB <= 0xA3 && 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;

8X16 点国标扩展字符说明:

BaseAdd: 说明本套字库在字库芯片中的起始字节地址。

FontCode: 表示字符内码 (16bits)

ByteAddress: 表示字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x3b7d0

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

www.huijinglcm.com

E-mail: huijinglcm@sina.com

```

if (FontCode >= 0xAAA1) and (FontCode <= 0xAAFE ) then
ByteAddress = (FontCode - 0xAAA1 ) * 16 + BaseAdd
Else if (FontCode >= 0xABA1) and (FontCode <= 0xABC0 ) then
ByteAddress = (FontCode - 0xABA1 + 95) * 16 + BaseAdd

```

5X7 点ASCII 字符参数说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

```

BaseAdd=0x3bfc0
if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then
Address = (ASCIICode - 0x20 ) * 8 + BaseAdd

```

7X8 点ASCII 字符参数说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

```

BaseAdd=0x66c0
if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then
Address = (ASCIICode - 0x20 ) * 8 + BaseAdd

```

8X16 点 ASCII 字符说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

```

BaseAdd=0x3b7c0
if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then
Address = (ASCIICode - 0x20 ) * 16 + BaseAdd

```

16 点阵不等宽ASCII 方头 (Arial) 字符说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

```

BaseAdd=0x3c2c0
if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then
Address = (ASCIICode - 0x20 ) * 34 + BaseAdd

```

8X16 点ASCII 粗体字符

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

www.huijinglcm.com

E-mail: huijinglcm@sina.com

说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

BaseAdd=0x3cf80

if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then

Address = (ASCIICode - 0x20) * 16 + BaseAdd

16 点阵不等宽ASCII 白正 (TimesNewRoman) 字符说明:

ASCIICode: 表示 ASCII 码 (8bits)

BaseAdd: 说明该套字库在芯片中的起始地址。

Address: ASCII 字符点阵在芯片中的字节地址。

计算方法:

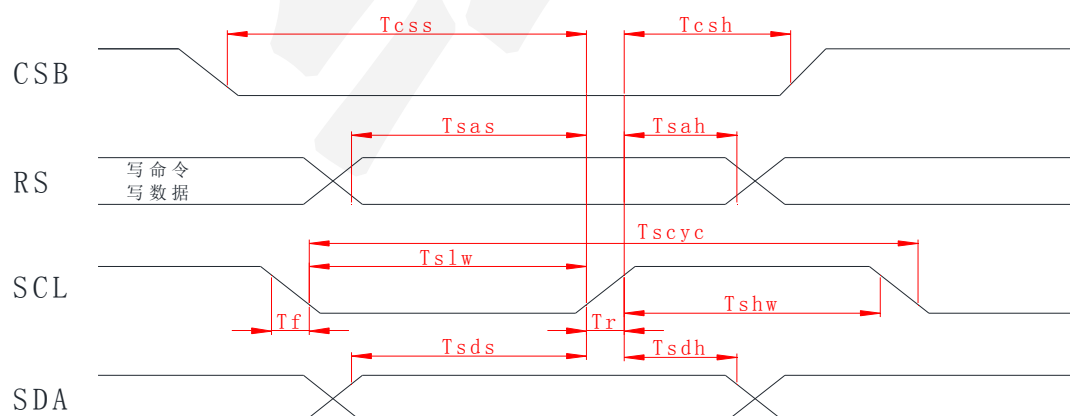
BaseAdd=0x3d580

if (ASCIICode >= 0x20) and (ASCIICode <= 0x7E) then

Address = (ASCIICode - 0x20) * 34 + BaseAdd

七. 时序图 (读写出入口时序)

模块串行接口时序图 (4-line SPi)



广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

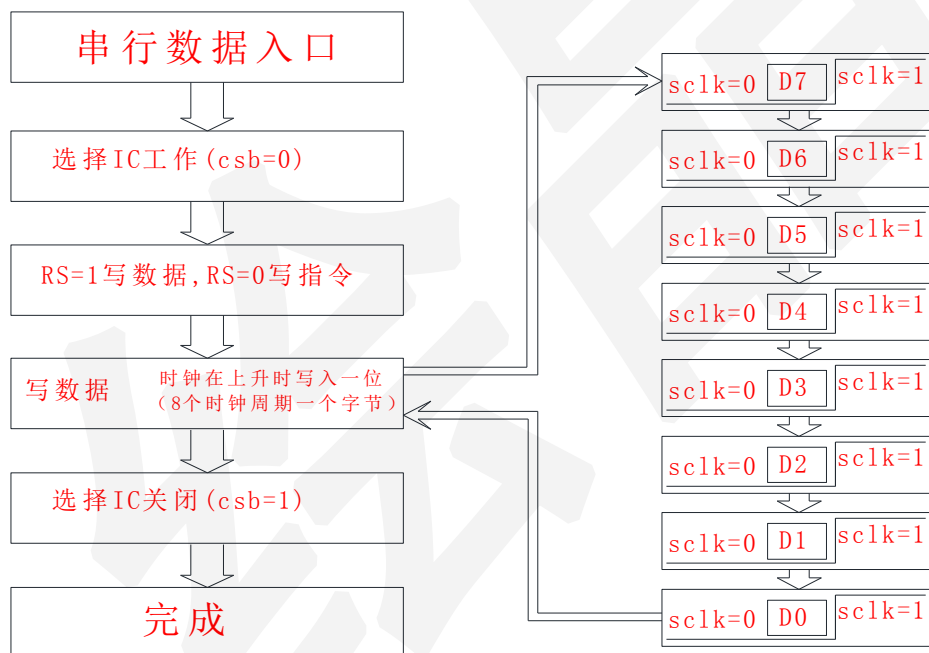
TEL: 0755-23146001 FAX: 0755-23146002

www.huijinglcm.com

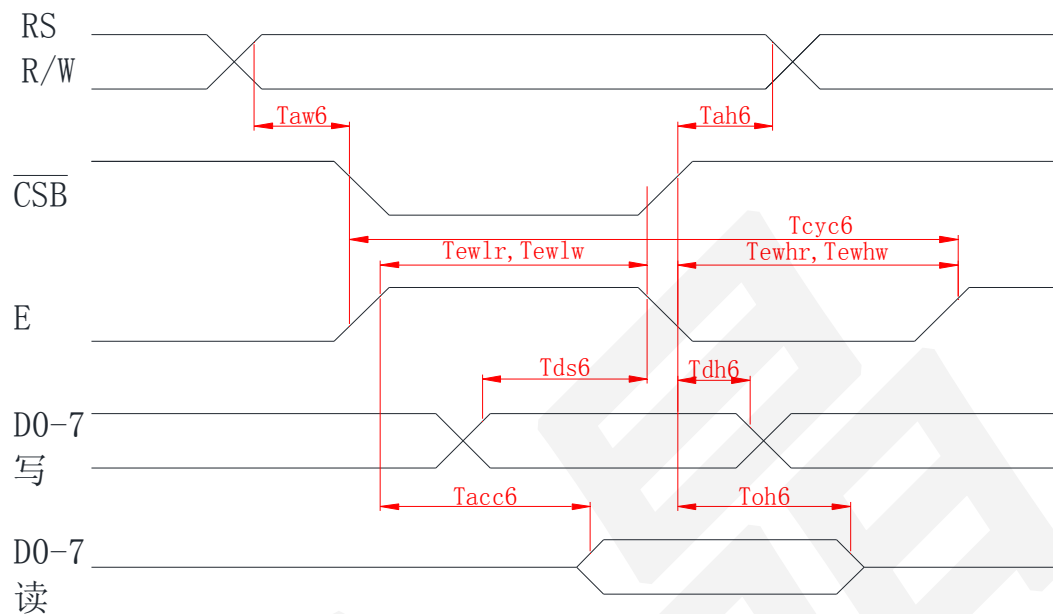
E-mail: huijinglcm@sina.com

项目	信号	标识	条件	最小	最大	单位
串行时钟	SCL	Tscyc		50	—	纳秒
时钟高电平宽度		Tshw		25	—	
时钟高电平宽度		Tslw		25	—	
指令建立时间	RS	Tsas		20	—	
指令宽度		Tsah		10	—	
数据建立时间	SDA	Tsds		20	—	
数据宽度		Tsdh		10	—	
片选建立时间	CSB	Tcss		20	—	
片选保持时间		Tcsh		40	—	

写串口入口程序流程图

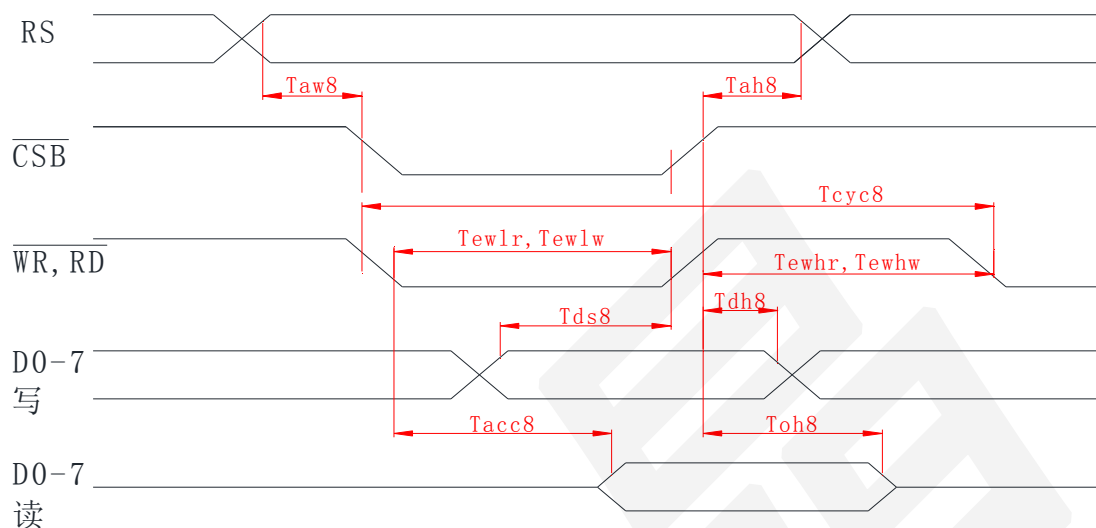


模块并行接口时序图(6800)



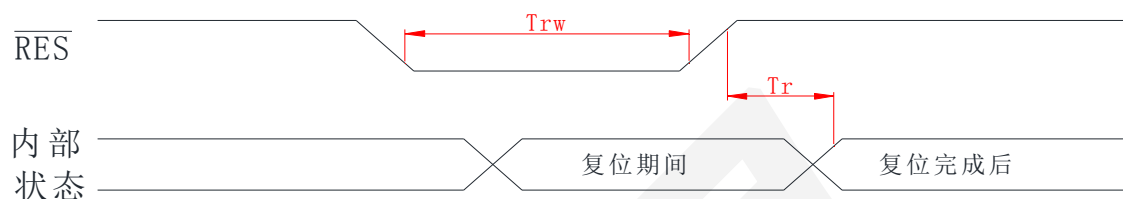
项目	信号	标识	条件	最小	最大	单位
指令建立时间	RS	Taw6		0	-	纳秒
指令宽度		Tah6		0	-	
系统循环时间	E	Tcyc6		400	-	
写使能低电平宽度		Tewlw		220	-	
写使能高电平宽度		Tewhw		180	-	
读使能低电平宽度		Tewlr		220	-	
读使能高电平宽度		Tewhr		180	-	
写数据建立时间	D[7:0]	Tds6		40	-	
写数据数据宽度		Tdh6		0	-	
读数据时间		Tacc6	CL=100pF	-	140	
读数据禁用时间		Toh6	CL=100pF	10	100	

模块并行接口时序图(8080)



项目	信号	标识	条件	最小	最大	单位
指令建立时间	RS	Taw8		0	-	纳秒
指令宽度		Tah8		0	-	
系统循环时间	/WR	Tcyc8		400	-	
写使能低电平宽度		Tewlw		220	-	
写使能高电平宽度		Tewhw		180	-	
读使能低电平宽度	RD	Tewlr		220	-	
读使能高电平宽度		Tewhr		180	-	
写数据建立时间	D[7:0]	Tds8		40	-	
写数据数据宽度		Tdh8		0	-	
读数据时间		Tacc8	CL=100pF	-	140	
读数据禁用时间		Toh8	CL=100pF	10	100	

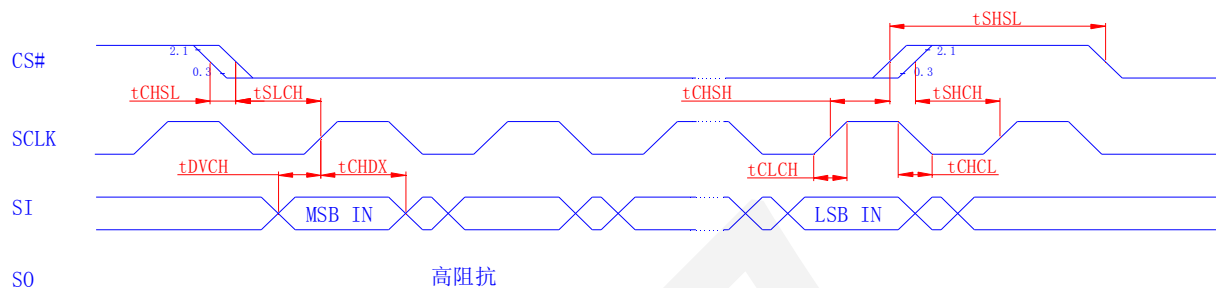
模块复位时序图



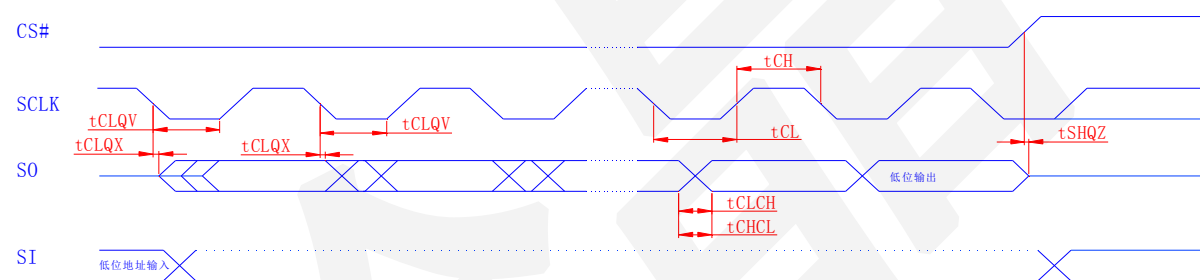
项目	符号	测试条件	范围			单位
			最小	最佳	最大	
复位保持低电平时间	T_{rw}	引脚: RES	3.0	—	—	us
复位到内部状态延时	T_r	引脚和 IC 内部状态	6.0	—	—	ms

字库IC时序图

串行输入时间



输出时间



标识	ALT	说明	最小	最大	单位
Fc	F	时钟频率	D.C	30	MHZ
Tch	T	高时钟时间	15		NS
Tcl	Tcll	低时钟时间	15		NS
TCLCH		时钟上升时间	0.1		V/NS
TCHCL		时钟下降时间	0.1		V/NS
TSLCH	TCSS	片选启动设置时间 (相对SCLK)	5		NS
TCHSL		片选不启动保持时间 (相对SCLK)	5		NS
TDVCH	TDSU	数据输入设置时间	2		
TCHDX	TDH	数据保持时间	5		NS
TCHSH		片选动作保持时间 (相对SCLK)	5		NS
TSHCH		片选不动作保持时间 (相对 (SCLK)	5		NS
TSHSL	TCSH	片选选择时间	100		NS
TSHQZ	TDIS	输出禁止时间		9	NS
TCLQV	TV	低时钟输出有效		9	NS
TCLQX	TH0	输出保持时间			ns

八. 用户指令表

1、模块指令表:

NO	指令	指令码										HEX	说明
		A0	RWR	D7	D6	D5	D4	D3	D2	D1	D0		
1	显示开/关	0	0	1	0	1	0	1	1	1	D	AF	D=1, 显示开 D=0, 显示关
2	起始行设置 (COM0)	0	0	0	1	S5	S4	S3	S2	S1	S0	40	设置显示屏第一行的位置
3	页地址设置	0	0	1	0	1	1	Y3	Y2	Y1	Y0	B0	设置页地址
4	列地址设置	0	0	0	0	0	1	X7	X6	X5	X4	10	列地址高4位 (MSB)
		0	0	0	0	0	0	X3	X2	X1	X0	00	列地址低4位 (LSB)
5	读状态	0	1	BUS Y	MX	D	RST	0	0	0	0		读出模块的IC内部工作状态
6	写数据	1	0	D7	D6	D5	D4	D3	D2	D1	D0	00	向模块内写显示数据
7	读数据	1	1	D7	D6	D5	D4	D3	D2	D1	D0	00	读出模块IC的数据
8	列方向设置 (SEG)	0	0	1	0	1	0	0	0	0	MX	A0	设置列的扫描方向 MX=1, 反方向 (右到左) MX=0, 正方向 (左到右)
9	显示方向	0	0	1	0	1	0	0	1	1	INV	A6	INV=1, 反向显示 INV=0, 正常显示
10	全部点阵打开	0	0	1	0	1	0	0	1	0	AP	A4	AP=1, 点阵全部打开 AP=0, 正常
11	偏压选择	0	0	1	0	1	0	0	0	1	BS	A2	偏压比设置 0=1/9; 1=1/7 (1/64duty)
12	Read-modify-Write	0	0	1	1	1	0	0	0	0	0	E0	Columnaddress increment: Read:+0, Write:+1
13	END	0	0	1	1	1	0	1	1	1	0	EE	Exit Read-modify-Write mode
14	复位	0	0	1	1	1	0	0	0	1	0	E2	软件复位
15	行方向 (COM)	0	0	1	1	0	0	MY	-	-	-	C0	设置行方向 MY=1, 反方向 (下到上) MY=0, 正方向 (上到下)
16	LCD电源控制	0	0	0	0	1	0	1	VB	VR	VF	28	供应LCD电压的电路控制 =1, 打开; =0, 关闭
17	Regulation Ratio	0	0	0	0	1	0	0	RR2	RR1	RR0	20	选择调节电阻比
18	LCD电压设置 (对比度)	0	0	1	0	0	0	0	0	0	1	81	双指令模式, 6位精度调节LCD 电压 (显示双比度)
		0	0	0	0	EV5	EV4	EV3	EV2	EV1	EV0	00	
19	睡眠模式设置	0	0	1	0	1	0	1	1	0	MD	AC	MD=0, 睡眠模式
		0	0	0	0	0	0	0	0	0	0	00	MD=1, 正常模式
20	省电	0	0	复合指令									显示关+全部点阵开
21	设置升压倍数 (LCD 电压)	0	0	1	1	1	1	1	0	0	0	F8	双指令模式 设置升压倍数 BL (0 0) 2倍, 3倍, 4倍 BL (0 1) 5倍 BL (1 0) 6倍
		0	0	0	0	0	0	0	0	BL1	BL0	00	
22	空指令	0	0	1	1	1	0	0	0	1	1	E3	空操作
23	测试	0	0	1	1	1	1	-	-	-	-	F0	不要使用, 保留测试

字库IC指令表

	作用	指令码								HEX	地址字节	虚拟字节	数据字节
1	一般读取	0	0	0	0	0	0	1	1	03H	3	-	1 to ∞
2	快速读取点阵数据	0	0	0	0	1	0	1	1	0BH	3	1	1 to ∞

2、模块具体指令介绍:

1) 显示开关

D标志选择显示模式。

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	1	0	1	0	1	1	1	D	AE

D=1; 正常显示模式

D=0; 显示关闭, 所以SEGS/COMS输出低电平

2) 设置起始行

显示内存的第一行对应显示屏顶部的第一行地址设置

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	0	1	S5	S4	S3	S2	S1	S0	40

3) 页地址设置

128*8点阵为一页, $12864 = (1286 * 8) * 8$ 页, 与起始行决定行在显示屏上的位置

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	1	0	1	1	Y3	Y2	Y1	Y0	B0

4) 设置列地址

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	0	0	0	1	X7	X6	X5	X4	10

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	0	0	0	0	X3	X2	X1	X0	00

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

www.huijinglcm.comE-mail: huijinglcm@sina.com

第1列为 0x10(高4位) 0x00 (低4位) 列地址需要写两次
 第2列为 0x10(高4位) 0x01 (低4位)
 第128列为 0x17(高4位) 0x0F (低4位)

5) 读状态

读HJ12864-COG-7内部状态。串行接口模式读取功能不可用。

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	BUSY	MX	D	RST	0	0	0	0	

标志	描述
BUSY	BUSY= 0: 命令可以接受 BUSY= 1: 命令或复位程序执行
MX	MX=0; 反向 (SEG131--SEG0) MX=1; 正常 (SEG0--SEG131)
D	D=0; 显示开 D=1; 显示关
RST	RST=1在复位 (硬件或软件复位) RES=0; 正常运行

- 6) 写数据 (看指令表)
- 7) 读数据 (看指令表)
- 8) 列方向 (看指令表)
- 9) 显示方向 (看指令表)
- 10) 全部点打开 (看指令表)
- 11) 偏压选择 (看指令表)
- 12) Read-modify-Wrie (少用)
- 13) END (少用)
- 14) 复位 (看指令表)
- 15) 水平方向设置 (看指令表)

16) 电源控制

一般3个标志位同时打开

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	0	0	1	0	1	VB	VR	VF	28

标志	描述
VB	VB=0; 内置升压关闭

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

www.huijinglcm.com

E-mail: huijinglcm@sina.com

	VB=1; 内置升压打开
VR	VR=0; 内置稳压关闭 VR=1; 内置稳压打开
VF	VF=0; 内置降压关闭 VF=1; 内置降压打开

17) 调节比

LCD对比度档位选择

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	0	0	1	0	0	RR2	RR1	RR0	20
	标志		调节比								
			0x20		3.0						
			0x21		3.5						
			0x22		4.0						
			0x23		4.5						
			0x24		5.0						
			0x25		5.5						
			0x26		6.0						
			0x27		6.5						

工作电压 (V0) 的计算公式如下: (RR 为调节比, EV 为 LCD 电压设置)

$$V0 = RR \times [1 - (63 - EV) / 162] \times 2.1, \text{ 或 } V0 = RR \times [(99 + EV) / 162] \times 2.1$$

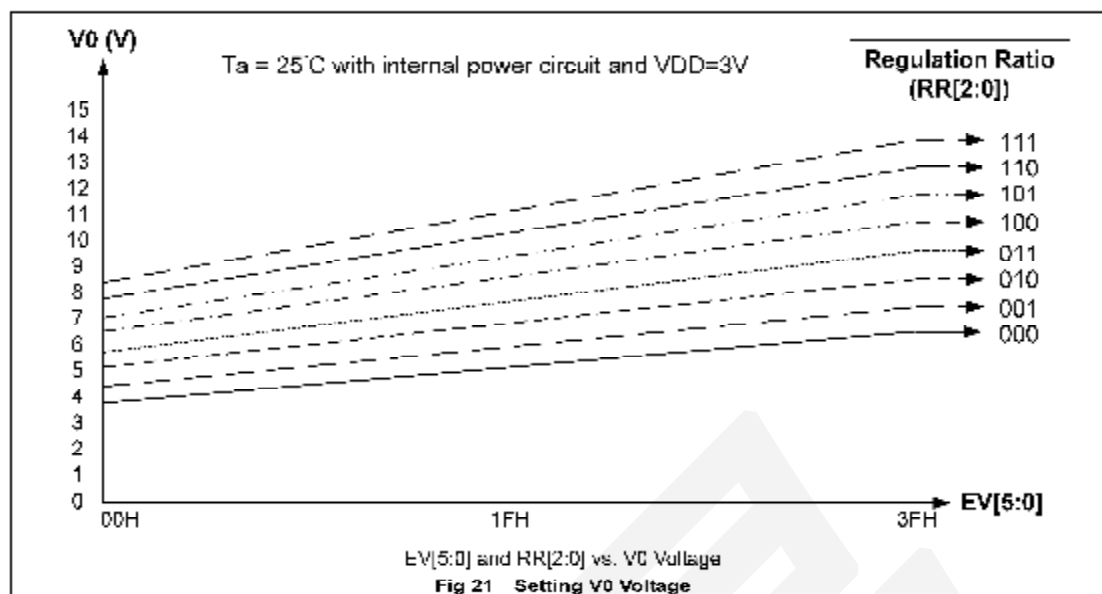
标志	寄存器	数值
RR	RR[2: 0]	3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5
EV	EV[5: 0]	0-63

18) LCD电压设置 (对比度调节)

双字节指令, 两条指令要一起使用可以调节LCD的深浅显示

	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	HEX
代码	0	0	1	0	0	0	0	0	0	1	81
	0	0	0	0	EV5	EV4	EV3	EV2	EV1	EV0	0

LCD电压与EV和RR之前的关系图



- 19) 睡眠模式设置 (看指令表)
- 20) 省电设置 (看指令表)
- 21) 升压倍数设置 (看指令表)
- 22) 空操作 (少用)
- 23) 测试 (不用)

字库IC指令说明

一般读取

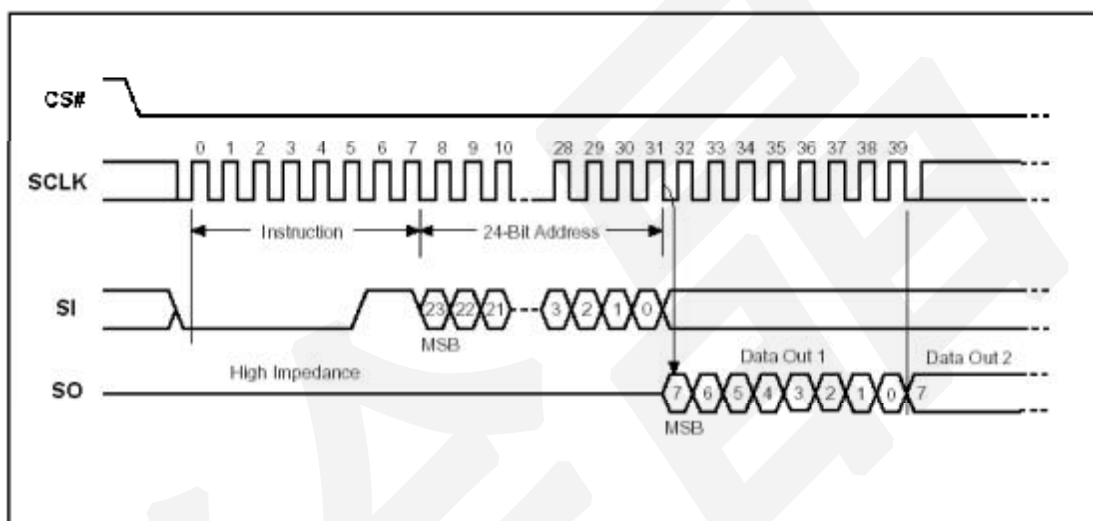
需要用指令码来执行每一次操作。READ 指令的时序如下(图)：

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。



快速读取点阵数据

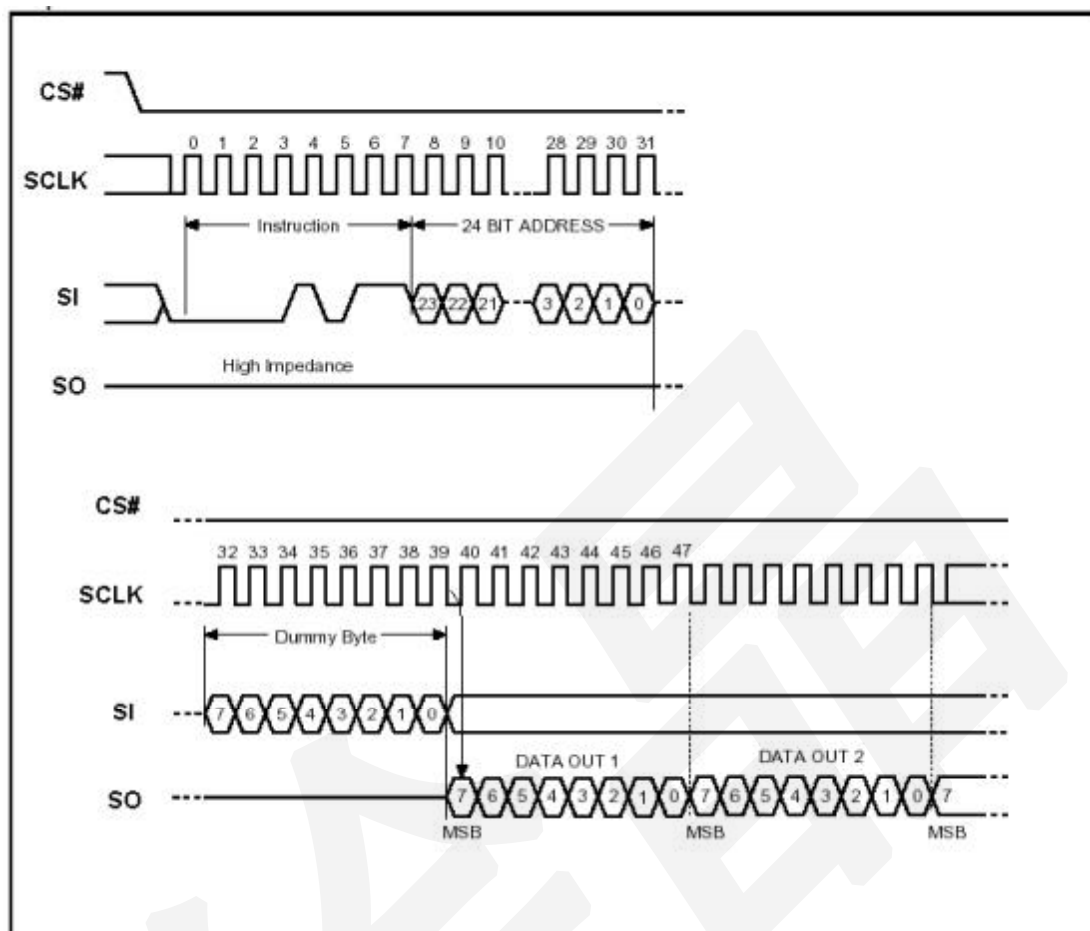
需要用指令码来执行操作。READ_FAST 指令的时序如下(图)：

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。

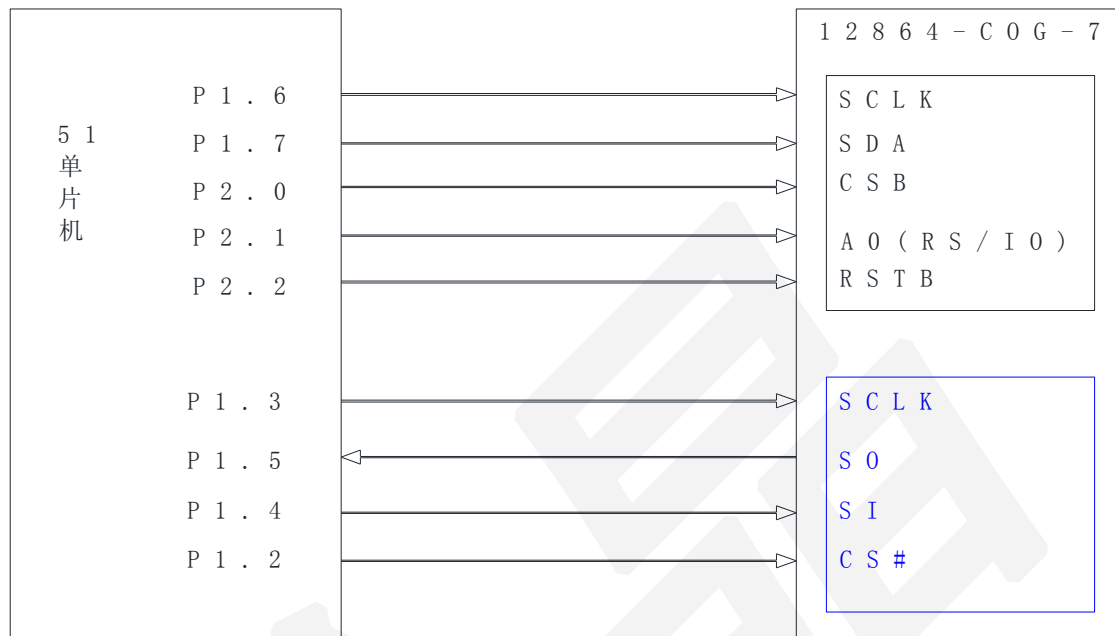
如果片选信号 (CS#) 继续保持为底, 则下一个地址的字节数据继续通过串行数据输出引脚 (SO) 移位输出。例: 读取一个15x16 点阵汉字需要32Byte, 则连续32 个字节读取后结束一个汉字的点阵数据读取操作。

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。

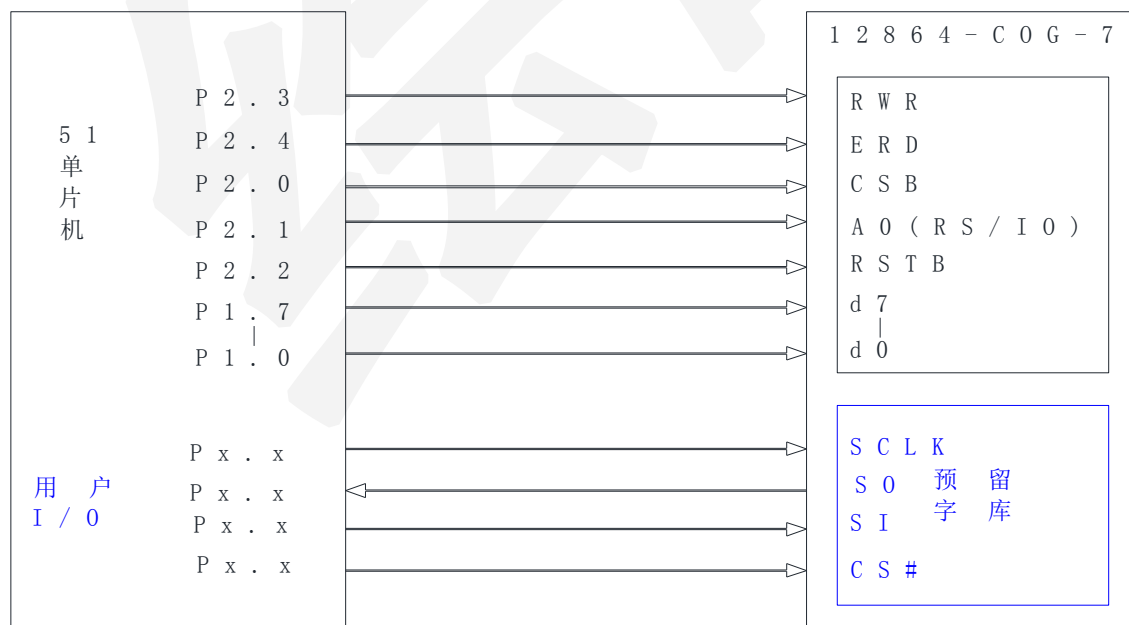


九. 连接图

串口+中文字库连接方法



并口连接方法



十. 应用举例

//中文字库：带中文字库IC:GB2312, 竖置横排

//单片机：51 系列(STC完全兼容)

//接口：串行时序

```
#include <reg52.H>
#include <intrins.h>
sbit lcd_sclk=P1^6;
sbit lcd_sid=P1^7;
sbit lcd_rs=P2^1;
sbit lcd_reset=P2^2;
sbit lcd_cs1=P2^0;
sbit Rom_IN=P1^4;
sbit Rom_OUT=P1^5;
sbit Rom_SCK=P1^3;
sbit Rom_CS=P1^2;

#define uchar unsigned char
#define uint unsigned int
#define ulong unsigned long

uchar code hui1[]={/*-- 文字： 绘 --*/
/*-- 黑体12； 此字体下对应的点阵为： 宽x高=16x16 --*/
0x40, 0x70, 0x7C, 0xFE, 0xFC, 0xF0, 0xE0, 0xF0, 0x7C, 0x7E, 0x4E, 0x7E, 0x78, 0x60, 0xE0,
0x40,
0x60, 0x66, 0x27, 0x27, 0x35, 0x34, 0x76, 0x72, 0x7A, 0x5E, 0x4E, 0x5A, 0xFA, 0xF2, 0xC2,
0x00,
};

uchar code jing1[]={/*-- 文字： 晶 --*/
/*-- 黑体12； 此字体下对应的点阵为： 宽x高=16x16 --*/
0x00, 0x00, 0x00, 0x7C, 0x7C, 0x54, 0x54, 0x54, 0x54, 0x54, 0x7C, 0x7C, 0x7C, 0x00, 0x00,
0x00,
0x00, 0x7F, 0x7F, 0x29, 0x29, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x7F, 0x29, 0x29, 0x7F, 0x7F,
0x00, };

//延时
void delay(int n_ms)
{
```

```
int j,k;
for(j=0;j<n_ms;j++)
for(k=0;k<110;k++);
}
//短延时
void delay_us(int n_us)
{
int j,k;
for(j=0;j<n_us;j++)
for(k=0;k<1;k++);
}

//写指令到LCD 模块
void transfer_command_lcd(int data1)
{
char i;
lcd_cs1=0;
lcd_rs=0;
for(i=0;i<8;i++)
{
lcd_sclk=0;
//delay_us(10); //加少量延时
if(data1&0x80) lcd_sid=1;
else lcd_sid=0;
lcd_sclk=1;
//delay_us(10); //加少量延时
data1=data1<<=1;
}
lcd_cs1=1;
}
//写数据到LCD 模块
void transfer_data_lcd(int data1)
{
char i;

lcd_cs1=0;
lcd_rs=1;
for(i=0;i<8;i++)
{
lcd_sclk=0;
if(data1&0x80) lcd_sid=1;
```

```
else lcd_sid=0;
lcd_sclk=1;
data1=data1<<=1;
}
lcd_csl=1;
}
//LCD 模块初始化
void initial_lcd()
{
    lcd_reset=0; //低电平复位
    delay(100);
    lcd_reset=1; //复位完毕
    delay(100);
    transfer_command_lcd(0xe2); //软复位
    delay(5);
    transfer_command_lcd(0x2c); //升压
    delay(50);
    transfer_command_lcd(0x2e); //稳压
    delay(50);
    transfer_command_lcd(0x2f); //降压
    delay(5);
    transfer_command_lcd(0x23); //粗调对比度,
    transfer_command_lcd(0x81); //微调对比度
    transfer_command_lcd(0x28); //微调对比度的值,
    transfer_command_lcd(0xa2); //1/9 偏压比 (bias)
    transfer_command_lcd(0xc8); //行扫描顺序: 从上到下
    transfer_command_lcd(0xa0); //列扫描顺序: 从左到右
    transfer_command_lcd(0x40); //起始行: 第一行开始
    transfer_command_lcd(0xaf); //开显示
}
void lcd_address(uint page,uint column)
{
    column=column-0x01;
    transfer_command_lcd(0xb0+page-1); //设置页地址,
    transfer_command_lcd(0x10+(column>>4&0x0f)); //设置列地址的高4 位
    transfer_command_lcd(column&0x0f); //设置列地址的低4 位
}
//全屏清屏
void clear_screen()
{
    unsigned char i,j;
    for(i=0;i<9;i++)
```

```
{
transfer_command_lcd(0xb0+i);
transfer_command_lcd(0x10);
transfer_command_lcd(0x00);
for(j=0;j<132;j++)
{
transfer_data_lcd(0x00);
}
}
}
```

```
void display_graph1(uchar da1,uchar da2)
{
unsigned char i,j;
for(i=0;i<9;i++)
{
transfer_command_lcd(0xb0+i);
transfer_command_lcd(0x10);
transfer_command_lcd(0x00);
for(j=0;j<66;j++)
{
transfer_data_lcd(da1);
transfer_data_lcd(da2);
}
}
}
```

//显示128x64 点阵边框

```
void display_bk()
{
uint i,j;

//左框
for(j=0;j<8;j++)
{
lcd_address(j+1,1);
transfer_data_lcd(0xff);
}
}
```

```
//右框
for(j=0;j<8;j++)
```



```
{
lcd_address(j+1, 128);
transfer_data_lcd(0xff);
}

//上框
lcd_address(1, 2);
for (i=0;i<126;i++)
{
transfer_data_lcd(0x01);
}

//下框
lcd_address(8, 2);
for (i=0;i<126;i++)
{
transfer_data_lcd(0x80);
}

}

//显示16x16 点阵图像、汉字、生僻字或16x16 点阵的其他图标
void display_graphic_16x16(uchar page, uchar column, uchar *dp)
{
uint i, j;
for(j=0;j<2;j++)
{
lcd_address(page+j, column);
for (i=0;i<16;i++)
{
transfer_data_lcd(*dp);
dp++;
}
}
}

//送指令到字库IC
void send_command_to_ROM( uchar datu )
{
uchar i;
```

```

for(i=0;i<8;i++ )
{
Rom_SCK=0;
delay_us(10);
if(datu&0x80)Rom_IN = 1;
else Rom_IN = 0;
datu = datu<<1;
Rom_SCK=1;
delay_us(10);
}
}
//从字库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;
for(i=0;i<8;i++)
{
Rom_OUT=1;
Rom_SCK=0;
//delay_us(1);
ret_data=ret_data<<1;
if( Rom_OUT )
ret_data=ret_data+1;
else
ret_data=ret_data+0;
Rom_SCK=1;
//delay_us(1);
}
return(ret_data);
}
//从指定地址读出数据写到液晶屏指定（page,column）座标中
void get_and_write_16x16(ulong fontaddr,uchar page,uchar column)
{
uchar i,j,disp_data;
Rom_CS = 0;
send_command_to_ROM(0x03);
send_command_to_ROM((fontaddr&0xff0000)>>16); //地址的高8 位,共24 位
send_command_to_ROM((fontaddr&0xff00)>>8); //地址的中8 位,共24 位
send_command_to_ROM(fontaddr&0xff); //地址的低8 位,共24 位
for(j=0;j<2;j++)
{

```

```

lcd_address(page+j, column);
for(i=0; i<16; i++ )
{
disp_data=get_data_from_ROM();
transfer_data_lcd(disp_data); //写数据到LCD, 每写完1 字节的数据后列地址自动加
1
}
}
Rom_CS=1;
}
//从指定地址读出数据写到液晶屏指定 (page, column)座标中
void get_and_write_8x16(ulong fontaddr, uchar page, uchar column)
{
uchar i, j, disp_data;
Rom_CS = 0;
send_command_to_ROM(0x03);
send_command_to_ROM((fontaddr&0xff0000)>>16); //地址的高8 位, 共24 位
send_command_to_ROM((fontaddr&0xff00)>>8); //地址的中8 位, 共24 位
send_command_to_ROM(fontaddr&0xff); //地址的低8 位, 共24 位
for(j=0; j<2; j++)

{
lcd_address(page+j, column);
for(i=0; i<8; i++ )
{
disp_data=get_data_from_ROM();
transfer_data_lcd(disp_data); //写数据到LCD, 每写完1 字节的数据后列地址自动加
1
}
}
Rom_CS=1;
}
//从指定地址读出数据写到液晶屏指定 (page, column)座标中
void get_and_write_5x8(ulong fontaddr, uchar page, uchar column)
{
uchar i, disp_data;
Rom_CS = 0;
send_command_to_ROM(0x03);
send_command_to_ROM((fontaddr&0xff0000)>>16); //地址的高8 位, 共24 位
send_command_to_ROM((fontaddr&0xff00)>>8); //地址的中8 位, 共24 位
send_command_to_ROM(fontaddr&0xff); //地址的低8 位, 共24 位
lcd_address(page, column);

```

```

for(i=0; i<5; i++ )
{
disp_data=get_data_from_ROM();
transfer_data_lcd(disp_data); //写数据到LCD, 每写完1 字节的数据后列地址自动加
1
}
Rom_CS=1;
}
//*****
ulong fontaddr=0;
void display_GB2312_string(uchar page,uchar column,uchar *text)
{
uchar i= 0;
while((text[i]>0x00))
{
if(((text[i]>=0xb0) &&(text[i]<=0xf7))&&(text[i+1]>=0xa1))
{
//国标简体 (GB2312) 汉字在字库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);
get_and_write_16x16(fontaddr, page, column);
i+=2;
column+=16;
}
else if(((text[i]>=0xa1) &&(text[i]<=0xa3))&&(text[i+1]>=0xa1))
{
//国标简体 (GB2312) 15x16 点的字符在字库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);
get_and_write_16x16(fontaddr, page, column);
i+=2;
column+=16;
}
else if((text[i]>=0x20) &&(text[i]<=0x7e))
{

```

```

fontaddr = (text[i]- 0x20);
fontaddr = (unsigned long)(fontaddr*16);
fontaddr = (unsigned long)(fontaddr+0x3cf80);
get_and_write_8x16(fontaddr, page, column);
i+=1;
column+=8;
}
else
i++;
}
}

```

```

void display_string_5x8(uchar page, uchar column, uchar *text)
{
unsigned char i= 0;
while((text[i]>0x00))
{
if((text[i]>=0x20) &&(text[i]<=0x7e))
{
fontaddr = (text[i]- 0x20);
fontaddr = (unsigned long)(fontaddr*8);
fontaddr = (unsigned long)(fontaddr+0x3bfc0);
get_and_write_5x8(fontaddr, page, column);
i+=1;
column+=6;
}
else
i++;
}
}

```

```

void ini_int1(void)
{
EA=1;
EX0=1;//允许外部INT0的中断
IT0=1;// 允许中断
}

```

```
int scankey1() interrupt 0 using 1 //使用外部中断1, 寄存器组3
```

广东省深圳市宝安区石岩镇麻布第二工业区 4 栋 2 楼

TEL: 0755-23146001 FAX: 0755-23146002

www.huijinglcm.com

E-mail: huijinglcm@sina.com

```
{
while(P3^2==0){for(;;);}
    IE1=0;//中断标志清零

}

//=====main program=====
void main(void)
{
    ini_int1();//开中断
    while(1)
    {
        initial_lcd();
        clear_screen(); //clear all dots

        display_graph1(0x55,0x55);//隔横显示
        delay(1000);
        clear_screen();

        display_graph1(0xff,0x00);//隔竖显示
        delay(1000);
        clear_screen();

        display_bk();
        display_GB2312_string(3,33,"绘晶科技");//图形加汉字 字库显示
        display_GB2312_string(5,40,"128x64");
        delay(1000);
        clear_screen();

        transfer_command_lcd(0xa7);//反显
        display_GB2312_string(3,33,"绘晶科技");//反显演示
        display_GB2312_string(5,40,"128x64");
        delay(1000);
        clear_screen();
        transfer_command_lcd(0xa6);

        display_GB2312_string(1,1,"12864,带中文字库");
        display_GB2312_string(3,1,"16X16 简体汉字库,");
        display_GB2312_string(5,1,"或8X16 点阵ASCII,");
        display_GB2312_string(7,1,"或5x8 点阵ASCII 码");
```

```

delay(1000);
clear_screen();
display_GB2312_string(1,1," 绘晶成立于二零");
display_GB2312_string(3,1,"一零年零一月一日");
display_GB2312_string(5,1,"主要生产液晶模块");
display_GB2312_string(7,1,"汇聚焦点精品至上");
delay(1000);
display_GB2312_string(1,1,"GB2312简体字库及");
display_GB2312_string(3,1,"有图型功能,可自");
display_GB2312_string(5,1,"编大字或图像或生");
display_GB2312_string(7,1,"生字,例如: ");
display_graphic_16x16(7,97,huil);
display_graphic_16x16(7,113,jing1);
delay(1000);

clear_screen();
display_GB2312_string(1,1,"<!@#$$%^&*()_+]/");
display_string_5x8(3,1,"<!@#$$%^&*()_+]/;.,?["");
display_string_5x8(4,1,"H J electronics Co., ");
display_string_5x8(5,1,"Ltd. established at ");
display_string_5x8(6,1,"year 2010.Focus LCM. ");
display_string_5x8(7,1,"TEL:0755-23146001 ");
display_string_5x8(8,1,"FAX:0755-61901158 ");
delay(1000);
display_GB2312_string(1,1,"啊阿埃挨哎唉哀皑");
display_GB2312_string(3,1,"癌蔼矮艾碍爱隘鞍");
display_GB2312_string(5,1,"氨安俺按暗岸胺案");
display_GB2312_string(7,1,"肮昂盎凹敖熬翱袄");
delay(1000);
display_GB2312_string(1,1,"鬟鬣麽麽麽麋麋塵");
display_GB2312_string(3,1,"麋麒麇麝麟黛黝黝");
display_GB2312_string(5,1,"黠黠黠黠黠黠黠");
display_GB2312_string(7,1,"飨飨飨飨飨飨飨");
delay(1000);

}
}

```

//~~~~~完~~~~~

并口时序

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

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

```
{  
cs1=0;  
rs=0;  
//rd=0;  
wr=0;  
P1=data1;  
//rd=1;  
wr=1;  
rs=1;  
cs1=1;  
//rd=0;  
}
```

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

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

```
{  
cs1=0;  
rs=1;  
//rd=0;  
wr=0;  
P1=data1;  
//rd=1;  
wr=1;  
rs=0;  
cs1=1;  
//rd=0;  
}
```

```
*/
```

```
//~~~~~完~~~~~
```