

HS240128S1 图形点阵液晶模块使用说明书

感谢您关注和使用我们的液晶产品。如果您在使用中有任何疑问，请拨打我们的客户服务热线 **0755-86114312** 寻求技术支持和获取相关资料，我们竭诚为您服务。您可以登录我们的网站了解最新产品信息。或者您可以在我公司网站的留言簿栏目留下您宝贵的意见。

深圳汉昇实业有限公司

SHENZHEN HANSHENG INDUSTRIAL CO.,LTD

地 址：深圳市南山区西丽阳光社区米坑新锋工业园 2 栋 5 楼
邮 编：518055
公司主页：www.hsicm.com
联系电话：0755-86114312
传 真：0755-86114312-812

外形尺寸表:

项 目	标 准 尺 寸	单 位
模 块 体 积	144×104×13.3	mm
定 位 尺 寸	138×97	mm
视 域	114×64	mm
行 列 点 阵 数	240×128	dots
点 距 离	0.45×0.45	mm
点 大 小	0.40×0.40	mm

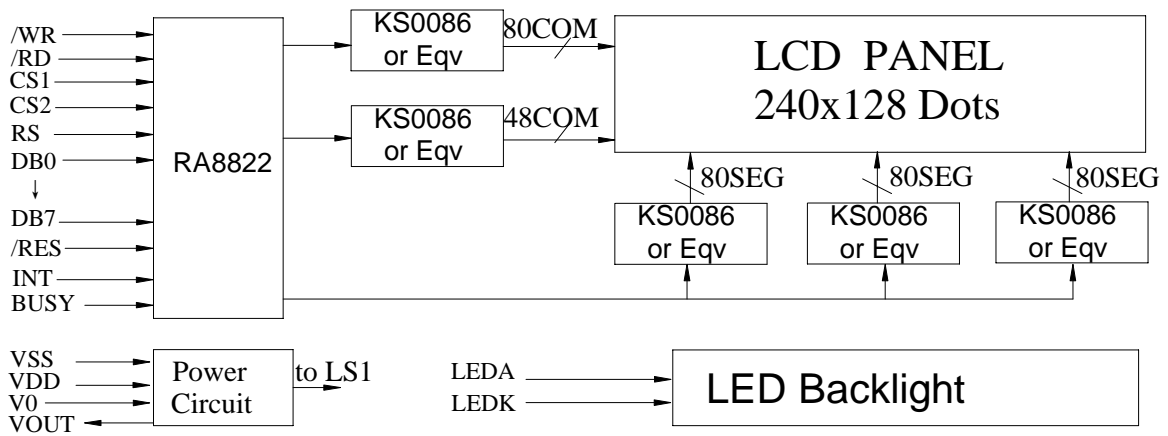
接口定义

引脚号	引脚名称	级 别	引 脚 功 能 描 述
1, 3	VSS	0V	电源地
2, 4	VDD	+5V	电源正
5	V0(NC)	负压	LCD 驱动电压调节 (对比度调节)
6	RS	H/L	指令数据选择 H:数据; L:指令
7	WR	H/L	写信号 低有效
8	RD	H/L	读信号 低有效
9	CS1	H/L	片选 1, 低有效
10	CS2	H/L	片选 2, 高有效
11	BUSY	H/L	忙信号
12	INT	H/L	中断信号
13	RST	H/L	复位信号 低有效, 复位完成后拉高
14-21	DB0-DB7	H/L	八位三态并行数据总线
22	VEE(NC)	负压	LCD 驱动电压
23	BLA	5.0V	背光正
24	BLK	0V	背光负

键盘接口定义

引脚号	引脚名称	级 别	引 脚 功 能 描 述
1-8	KC0-KC7	H/L	键盘输出
9-16	KR0-KR7	H/L	键盘输入

原理简图



最大工作范围

- 1、逻辑工作电压 (V_{cc}): 4.5~5.5V
- 2、电源地 (GND): 0V

电气特性 (测试条件 $T_a=25$, $V_{dd}=5.0\pm 0.25V$)

- 1、输入高电平 (V_{ih}): 3.5Vmin
- 2、输入低电平 (V_{il}): 0.55Vmax
- 3、输出高电平 (V_{oh}): 3.75Vmin
- 4、输出低电平 (V_{ol}): 1.0Vmax
- 5、模块工作电流:
- 6、侧黄绿光工作电流:

指令寄存器总表

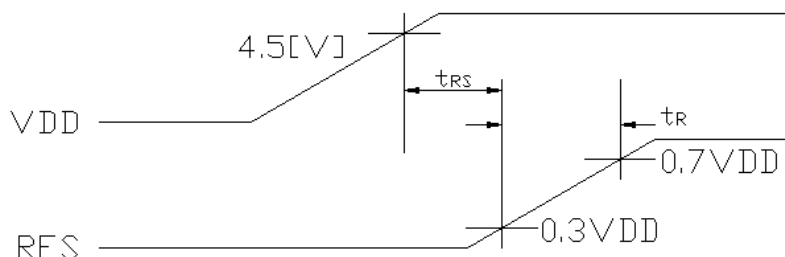
Reg. No	Reg. Name	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Default Data
00h	WLCR	R/W	PW1	PW0	SR		CG	DP	DK	DV	C9h
01h	MISC	R/W		CKN		PLR			CKB1	CKB0	F0h
02h	APSR	R/W			SP1	SP0	OAR		SRFS		10h
03h	ADSR	R/W					DADR	AUCM	AUSG	SGCM	80h
10h	WCCR	R/W	ARI	ALG	WDI	WBC	AWI	CP	CK	CSD	6Fh
11h	DWLR	R/W	CR3	CR2	CR1	CR0	DY3	DY2	DY1	DY0	22h
12h	MAMR	R/W	GIM	RM2	RM1	RM0	OP1	OP2	WM1	WM0	91h
20h	AWRR	R/W			X5	X4	X3	X2	X1	X0	27h
21h	DWRR	R/W			A5	A4	A3	A2	A1	A0	27h
30h	AWBR	R/W	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	EFh
31h	DWBR	R/W	B7	B6	B5	B4	B3	B2	B1	B0	EFh
40h	AWLR	R/W			SS5	SS4	SS3	SS2	SS1	SS0	00h
41h	DWLR	R/W			C5	C4	C3	C2	C1	C0	00h
50h	AWTR	R/W	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0	00h
51h	DWTR	R/W	D7	D6	D5	D4	D3	D2	D1	D0	00h
60h	CPXR	R/W			RS5	RS4	RS3	RS2	RS1	RS0	00h
61h	BGS	R/W			DS5	DS4	DS3	DS2	DS1	DS0	00h
70h	CPYR	R/W	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	00g
71h	BGCM	R/W	CB7	CB6	CB5	CB4	CB3	CB2	CB1	CB0	00h
72h	EDCM	R/W	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0	EFh
80h	BTMR	R/W	BT7	BT6	BT5	BT4	BT3	BT2	BT1	BT0	33h
81h	FRCA	R/W	FA7	FA6	FA5	FA4	FA3	FA2	FA1	FA0	00h
90h	SCCR	R/W	CK7	CK6	CK5	CK4	CK3	CK2	CK1	CK0	04h
91h,	FRCB	R/W	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0	00h
A0h	INTR	R/W	INK	INT	INX	INY	MSK	MST	MSX	MSY	00h
A1h	KSCR	R/W	KEN	KSZ	KDT1	KDT0		KF2	KF1	KF0	00h
A2h	KSDR	RO	KS7	KS6	KS5	KS4	KS3	KS2	KS1	KS0	00h
A3h	KSER	RO	KD7	KD6	KD5	KD4	KD3	KD2	KD1	KD0	00h
B0h	INTX	R/W			IX5	IX4	IX3	IX2	IX1	IX0	27h
B1h	INTY	R/W	IY7	IY6	IY5	IY4	IY3	IY2	IY1	IY0	EFh
C0h	TPCR	R/W	AZEN	AZOE		AUTO	AS3	AS2	AS1	AS0	00h
C1h	TPSR	R/W	ARDY	ADET	1		AF1	AF0			0Fh
C8h	TPXR	RO	TPX9	TPX8	TPX7	TPX6	TPX5	TPX4	TPX3	TPX2	00h
C9h	TPYR	RO	TPY9	TPY8	TPY7	TPY6	TPY5	TPY4	TPY3	TPY2	00h
CAh	TPZR	RO	TPX1	TPX0			TPY1	TPY0			00h
D0h	LCCR	R/W	DZEN			DAC4	DAC3	DAC2	DAC1	DAC0	8Fh
E0h	PNTR	R/W	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0	00h
F0h	FNCR	R/W	TNS	BNK	RM1	RM0	FDA	ASC	ABS1	ABS0	92h
F1h	FVHT	R/W	FH1	FH0	FV1	FV0	1	1	1	1	0Fh

时序

显示器在上电之后要先对整个模块进行一次复位，即在 RESET 脚加一个低电平一段时间（HS240128S1 复位时间需要的较长，建议 500 毫秒），然后拉高到高电平，再开始对模块进行初始化等操作。本手册后面的演示程序没有复位信号是因为外部采取了硬件电阻电容上电充电复位，用户也可以自己拉一个 IO 口进行软件复位。

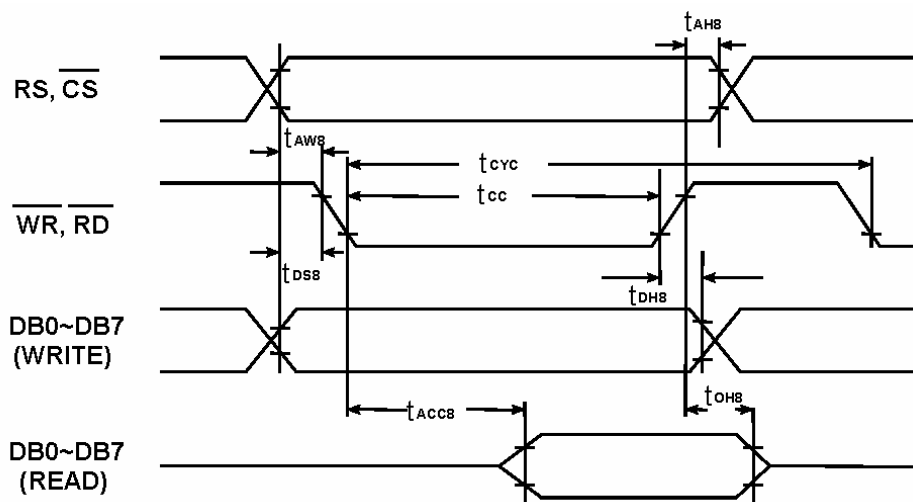
始化条件：

项目	名称	最小值	标准值	最大值	单位
Reset Time	t_{RS}	100.0	-	-	ms
Rise Time	t_R	-	-	200	ns



HS240128S1 支持 Intel8080 时序和 M6800 时序。RS 为“L”时是表示对缓存器下命令，也就是对 HS240128S1 的指令寄存器进行读写的动作 (Register Access Cycle)，而 RS 为“H”时是表示对 Display RAM 进行 Data 读写的动作 (Data Access Cycle)。不论是 8080 或 6800，“RS” Pin 通常接到 MCU 的 Address Pin “A0”，8080 系列 MCU 与 6800 最大的不同是 Read、Write 的控制信号是分开的，RD 为 Low 时是进行读取动作，WR 为 Low 时是进行写入动作，至于读写的目的地则由 RS 决定。

下图表示如果是对指令寄存器进行读取动作，MCU 必须透过数据总线先送出指令寄存器的地址，然后才能在数据总线上读取寄存器的数据，如果是对指令寄存器进行写入动作，MCU 必须透过数据总线先送出指令寄存器的地址，然后再送出要写入的数据。当 8080 MCU 对 HS240128S1 Display RAM 进行数据的读取动作，MCU 能直接在数据总线上读取 Display RAM 的数据，如果 8080 MCU 对 Display RAM 进行数据的写入动作，MCU 则直接在数据总线上送出要写入的数据。

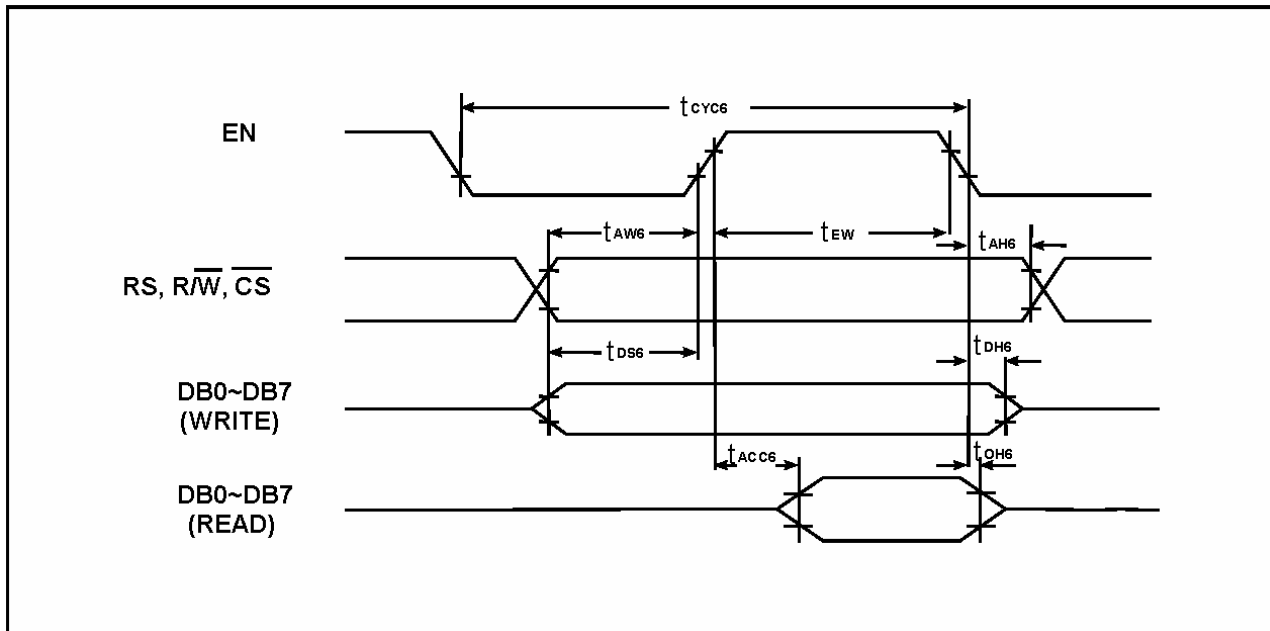


8080 MCU 对寄存器/Data 进行读取/写入动作

Signal	Symol	Parameter	Rating		Unit	Condition
			Min	Max		
RS,CS#	t _{AH8}	Address hold time	10	--	ns	System Clock: 8MHz Voltage: 3.3V
	t _{Aw8}	Address setup time	63	--	ns	
WR# , RD#	t _{CYC8}	System cycle time	800	--	ns	
	t _{CC8}	Strobe pulse width	400	--	ns	
DB0 to DB7	t _{DS8}	Data setup time	63	--	ns	
	t _{DH8}	Data hold time	10	--	ns	
	t _{ACC8}	RD access time	--	330	ns	
	t _{OH8}	Output disable time	10	--	ns	

HS240128S1 无法同时接受6800 及8080 的控制信号，因此在MCU 的接口上，某些脚位上会因为使用者选择不同的MCU 而有不同的定义，例如脚位RD#(EN)，当使用者选择的MCU 接口为8080 时是定义成RD#，而选择6800 MCU 时是定义为EN。而脚位WR#(R/W#)，当使用者选择的MCU 接口为8080 时是定义成WR#，而选择6800 MCU 时是定义为R/W#。

下图表示如果是6800 MCU 对HS240128S1的指令寄存器进行读取动作，MCU 必须透过数据总线先送出缓存器的地址，然后才能在数据总线上读取缓存器的数据，如果是对缓存器进行写入动作，MCU 必须透过数据总线先送出缓存器的地址，然后再送出要写入的数据。当6800MCU 对HS240128S1 Display RAM 进行数据的读取动作，MCU 能直接在数据总线上读取Display RAM 的数据，如果6800 MCU 对Display RAM 进行数据的写入动作，则MCU 直接在数据总线上送出要写入的数据。



6800 MCU 对寄存器/Data 进行读取/写入动作

Signal	Symol	Parameter	Rating		Unit	Condition
			Min	Max		
A0, RW# CS#	t _{AH6}	Address hold time	10	--	ns	System Clock: 8MHz
	t _{Aw6}	Address setup time	63	--	ns	

	t _{CYC6}	System cycle time	800	--	ns	Voltage: 3.3V
DB0 toDB7	t _{DS6}	Data setup time	400	--	ns	
	t _{DH6}	Data hold time	63	--	ns	
	t _{ACC6}	access time	10	--	ns	
	t _{OH6}	Output disable time	--	330	ns	
EN	t _{EW6}	Enable pulse width	10	--	ns	

MCU驱动程序说明

NO.	RS	6800	8080		DB0-DB7	Function
		R/W#	RD#	WR#		
①	1	1	0	1	xxh	Read Display Data
②	1	0	1	0	Hight Byte →Low Byte	Write Display Data (中文汉字) 步骤②必须作两次，第一次写入中文字内码的高字节位，第二此再写入低字节位
③	1	0	1	0	xxh	Write Display Data (英文,ASCII) 步骤③只须作一次，直接写入英文字型码或ASCII码
④	1	0	1	0	xxh	Write Display Data (图形模式)
⑤	0	0	1	0	Address	当要读取某指令寄存器状态(Read Status)必须完成两项步骤： 先步骤⑤→后步骤⑥，才可以读取状态
⑥	0	1	0	1	Status	
⑦	0	0	1	0	Address	当要写入控制命令到某指令寄存器必须要完成两项步骤： 先步骤⑦→后步骤⑧，才可以写入控制命令到指定的寄存器
⑧	0	0	1	0	Command	

下面列举一些简单的程序说明一下HS240128S1的指令或者数据读写方式，这些程序都是以C51编写的，浅显易懂，也非常容易转换成其他语言。

⑤ ⑦两步里面提到的Address即是指令寄存器总表里面的Reg.NO，要写入指令到指定的指令寄存器需要先写入该指令寄存器的地址，也就是编号，然后再写入指令字。读取相应的指令寄存器的的步骤也是一样。写指令寄存器的地址的时序（也就是子程序）和写入控制指令字的是一样的。

例1. REG [00h] = #CDH
WriteReg(0x00); //选择LCD Controller Register (WLCR)
WriteReg(0xCD); //写入" 0xCD" 到WLCR指令寄存器
//本手册后面附带的完整示例程序中，写指令是两个参数一起写

例2. REG [E0h] = #5AH
WriteReg(0xE0); //选择Pattern Data Register (PNTR)
WriteReg(0x5A); //写入" 0x5A" 到PNTR指令寄存器

例3. LCD显示“啊”字

```
WriteData(0xB0); //先写入“啊”的内码(B0A1)的高字节
```

```
WriteData(0xA1); //先写入“啊”的内码(B0A1)的低字节
```

各相关指令寄存器的功能说明

以下是各个相关指令寄存器的详细说明和解释，请参考本手册最后的示例程序。

REG [00h] Whole Chip LCD Controller Register (WLCR)

Bit	Description	Text/Graph	Default	Access
7-6	电源模式(Power Mode) 11: 正常模式(Normal Mode) RA8803/8822 的所有功能都可以使用(Available)。 00: 关闭模式(Off Mode) 除了唤醒(Wake-Up)电路工作外，其它功能都被禁止。当 Wake-Up 电路被触发，HS240128S1将回复至正常模式。	--	3h	R/W
5	软件重置 所有缓存器回到初始值，但是RAM 的内容不会被清除。 1: 重置所有缓存器 0: 正常模式，平常应保持为“0”	--	0h	R/W
4	保留	--	0h	R/W
3	选择显示工作模式 1: 文字模式，写入的数据会被视为是GB/BIG/ASCII 等字码。 0: 绘图模式，写入的数据会被视为是Bit-Map 的模式。	--	1h	R/W
2	设定屏幕显示为开启或关闭 此位用来控制连接到LCD 驱动器接口的“DISPOFF” 讯号 1: “DISPOFF” 讯号输出High(屏幕开启) 0: “DISPOFF” 讯号输出Low(屏幕关闭)	Text/Graph	0h	R/W
1	闪烁模式选择 1: 整个屏幕闪烁，闪烁时间可由缓存器BTMR 来设定 0: 正常显示，不闪烁	Text/Graph	0h	R/W
0	屏幕反白模式选择 1: 正常显示，不反白 0: 屏幕全反白，DDRAM里面内容全部反相	Text/Graph	1h	R/W

REG[00h] bit3 = 1

HS240128S1 的文字模式可以支持全角(中文或英文)及半角(英文)的显示，全角文字是以16x16 的点矩阵组成，半角文字是8x16 的点矩阵组成。HS240128S1还支持全角(中文)及半角(英文) 文字的混和显示。

HS240128S1 的中文显示方式与传统的LCD Controller 不同，传统的LCD Controller 是在绘图模式下，以 Bit-Map 的方式去绘出中文，HS240128S1 的中文显示方式则是在文字模式，直接输入中文字码(GB 或BIG5 码)，就可以在光标所在位置显示中文。因为中文字码占两个Byte，所以如果MCU 接口是8-Bit，则MCU 必须分两次将中文字内码(High Byte & Low Byte) 写入HS240128S1，而英文或数字码只占一个Byte，因此只要将内码一次写入HS240128S1 即可。HS240128S1 支持之最大显示像素范围为240x128 点，若以显示文字为例，可显示 8行x15个全角字型，8行x30个半角字型。

下面例题程序就是说明如何显示汉字和西文字符：

显示字型	字型内码
电	B5E7
子	D7D3
2	32H
4	34H
0	30H
1	31H
2	32H
8	38H

程序示例：

```
WriteData(0xB5); //先写入“电”的内码(B5E7)的高字节
WriteData(0xE7); //先写入“电”的内码(B5E7)的低字节
WriteData(0xD7); //先写入“子”的内码(D7D3)的高字节
WriteData(0xD3); //先写入“子”的内码(D7D3)的低字节
```

其实真正写程序时候不需要这么麻烦，不需要去查找每一个汉字的汉字内码，因为一般的编译器都支持直接编译汉字，编译的时候就把汉字当作内码处理了，比如KeilC编译的时候，显示图5-1只需要如下一条语句：

```
PutStr("汉昇液晶HS240128S1"); //PutStr子函数请参口本手册后面的示例程序
```

REG[00h] bit3 = 0

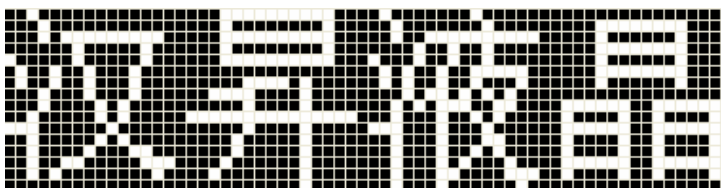
HS240128S1则进入绘图模式，按照bitmap的映射显示图形，这个和传统的LCD显示器一样，这里就不多做说明，可以参考本手册后面的示例程序；

REG[00h] bit0

此项设置是让全屏幕反相显示，如图所示：



正常显示



反白显示

注：后面为了手册制作方便，插入的图片有部分是直接调用的繁体字的插图，原理是相同的。

REG[00h] bit1

此功能就是上面反白功能的扩展利用，即利用全屏反白显示实现屏幕的闪烁。闪烁频率相当于反白显示和不反白显示之际的时间间隔。

REG [01h] Misc. Register (MISC)

Bit	Description	Default	Access
7	保留	1h	R/W
6	CLK_OUT 致能控制 1: 致能 0: 禁能 (此功能与使用无关, 可禁止掉)	1h	R/W
5	保留	1h	R/W
4	设定中断 (INT)/ BUSY 的触发准位 1: 设定高电位触发动作 0: 设定低电位触发动作	1h	R/W
3-2	保留	0h	R/W
1-0	系统时钟选择 0 0: 3MHz 0 1: 4MHz 1 0: 8MHz 1 1: 12MHz	0h	R/W

系统时钟一般选择4MHz或8 MHz, CLK_OUT对用户没有作用。

REG [02h] Advance Power Setup Register (APSR)

Bit	Description	Default	Access
7-6	保留	0h	R/W
5-4	设定ROM / RAM 的读取速度 0 0: Speed0 (30ns@Vdd=3.3V) 0 1: Speed1 (60ns@Vdd=3.3V) 1 0: Speed2 (90ns@Vdd=3.3V) 1 1: Speed3 (120ns@Vdd=3.3V)	1h	R/W
3	字型ROM 的直接读取 1: 致能 0: 禁能	0h	R/W
2	保留	0h	R/W
1	Scrolling Reset for Start 1: 致能 0: 禁能	0h	R/W
0	保留		

字型ROM的直接读取允许将HS240128S1里面的汉字字库的点阵读取出来用于其他用途，比如打印等功能。

读取的步骤为：1. 设定REG[02h]Bit3 = 1

2. 写入所需要读取的中文字的内码

3. 连续32次读数据即可读出此汉字的点阵数据

ROM/RAM的读取速度一般取默认值Speed1 60ns

REG [03h] Advance Display Setup Register (ADSR)

Bit	Description	Default	Access
7-4	保留	8h	R/W
3	设定 Display Data 的顺序, 以Byte 为单位作用 1: 反转整个Byte 内容 0: 正常状态, 不反转内容	0h	R/W
2	设定 Common 的自动滚动 1: 致能 0: 禁能	0h	R/W
1	设定 Segment 的自动平移 1: 致能 0: 禁能	0h	R/W
0	设定选择 Common 的滚动或是 Segment 的平移模式 1: Segment 的平移 0: Common 的滚动 扩展模式下(REG[12h] 的bit[6:4] = "110" 或"111"), 此位必须设为1。	0h	R/W

REG [10h] Whole Chip Cursor Control Register (WCCR)

Bit	Description	Text/Graph	Default	Access
7	设定当数据读出 DDRAM 时, 光标是否自动移位。 1: 致能(自动移位) 0: 禁能(不自动移位)	Text/Graph	0h	R/W
6	中/英文字对齐 1: 致能(对齐) 0: 禁能(不对齐) 此功能仅在文字模式时有效, 可以将全角与半角混合显示时作对齐调整。	Text	1h	R/W
5	储存 MCU 进来数据(正相/反相)于 DDRAM 1: 直接储存数据于 DDRAM 中 0: 存入相反的数据于 DDRAM 中	Text/Graph	1h	R/W
4	设定粗体字型(仅文字模式适用) 1: 粗体字型 0: 正常字型	Text	0h	R/W
3	此位用来设定当数据写入 DDRAM 时, 光标是否自动移位 1: 致能(自动移位) 0: 禁能(不自动移位)	Text/Graph	1h	R/W
2	光标显示 On/Off 设定 1: 设定光标 On 0: 设定光标Off	Text/Graph	1h	R/W
1	光标闪烁控制 1: 光标闪烁, 闪烁时间由缓存器 BTMR 来决定 0: 光标不闪烁	Text/Graph	1h	R/W
0	设定光标宽度 Mode 1: 会随着输入的数据而变动光标宽度, 当数据为半型时, 光标为一个字节宽度(8 个Pixel), 当数据为全型时, 光标为二个	Text	1h	R/W

	字节宽度(16 个Pixel) 0: 光标固定为一个字节的宽度(8 个Pixel)			
--	--	--	--	--

REG[10h] bit7 bit3

设置你每次读写数据的时候地址指针是否自动移位，一般都设置为1，这样每次连续读写数据的时候（包括写入汉字和图形）不需要每次都设置地址。

REG[10h] bit6

此设置仅仅在文本模式下，中英文混排的时候才需要用到，用户可以根据自己的需要设置，建议使用默认值。

中英文对齐：设定 REG[10h] bit6 = 1，写入两次“中文文字/图形LCD 控制器”，显示如图：

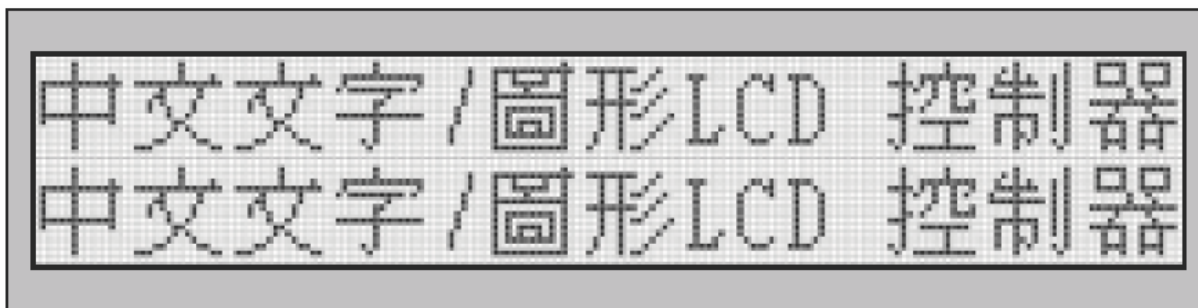


图5-3：中英文对齐

中英文不对齐：设定REG[10h] bit6 = 1，写入一次“中文文字/图形LCD控制器”，再设定REG[10h] bit6 = 0，写入一次“中文文字/图形LCD控制器”，显示如图5-4：

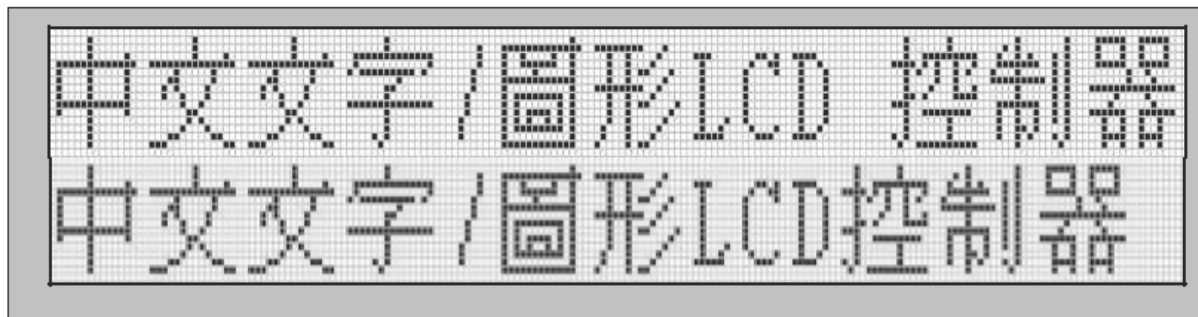


图5-4：中英文不对齐

REG[10h] bit5此设置和WLCR的bit0不一样，这个是在写入数据到DDRAM两面时候设置的，可以设置LCD部分内容反白显示，也可以使汉字反白显示。见图5-5：

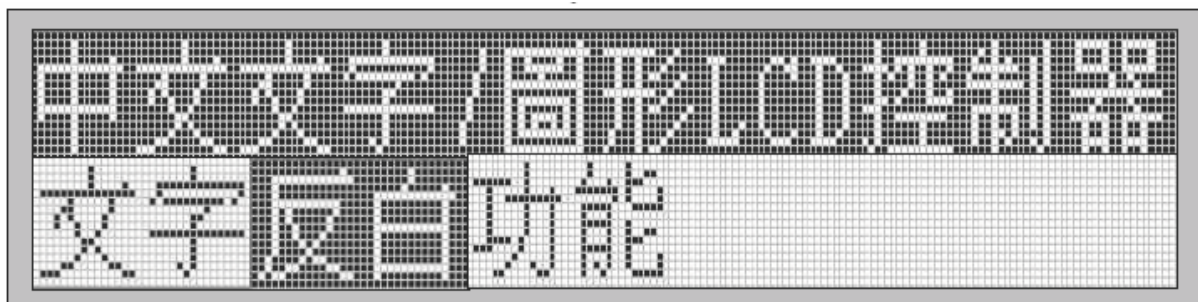


图5-5：汉字反白显示

1. 设定寄存器[10h] bit5=1

2. 写入"文字"的内码, LCD 就可显示出"文字"
3. 设定寄存器[10h] bit5=0
4. 写入"反白"的内码, LCD 就可显示出"反白"字样
5. 设定寄存器[10h] bit5=1
6. 写入"功能"的内码, LCD 就可显示出 "功能"

REG[10h] bit2

此设置允许设置写入的文字是否为粗体字, 粗体字的效果如图5-6:



图5-6: 粗体字显示效果

REG [11h] Distance of Words or Lines Register (DWLR)

Bit	Description	Default	Access
7-4	设定光标高度	2h	R/W
3-0	设定行与行的距离	2h	R/W

REG[11h]Bit7-4设置的屏幕上显示光标的高度, 与WCCR的bit0配套使用, 可以设置屏幕上光标的形状。如图5-7是宽度为16的时候高度为不同值时候的光标形状, 到16x16就是一个方块形状的光标。

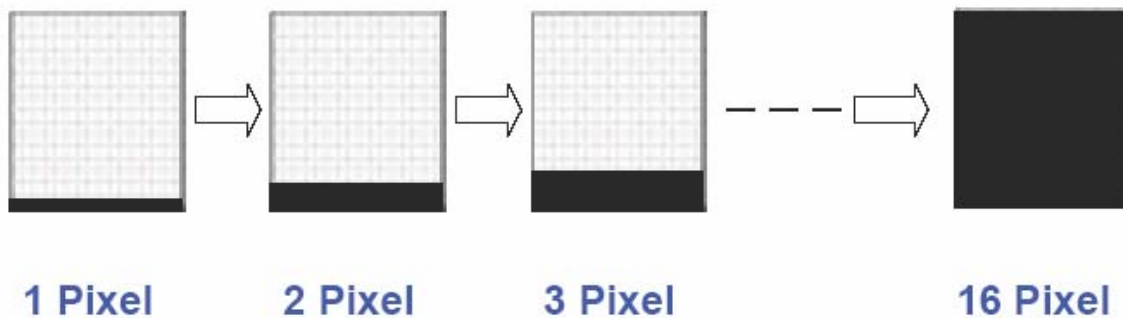


图5-7: 光标的高度

REG[11h]Bit3-0设置文本显示的时候行与行之间的距离。在中文显示的时候, 每一行如果有适当的间隔, LCD 的显示画面看起来会比较舒适。HS240128S1 行与行相隔的间距设定范围为1~16 Pixel 的高度, 使用者可依需求来决定行与行间距的大小, 一旦设定后, 当每填满一行的中文字, 跳到下一行时, 其行距会依照先前所设定的间距来显示。

REG [12h] Memory Access Mode Register (MAMR)

Bit	Description	Default	Access																		
7	图形模式时，光标自动移位的方向选择 1: 先水平移动再垂直移动 0: 先垂直移动再水平移动	1h	R/W																		
6-4	设定选择 Display Data RAM 的图层显示模式 0 0 1: 只有显示Page1 的图层 (单一上层显示模式) 0 1 0: 只有显示Page2 的图层 (单一下层显示模式) 0 1 1: 同时显示Page1 和Page2 的图层 (双层模式) 0 0 0: 灰阶显示 (Gray Mode), 此模式下每一个点的灰度决定于DDRAM Page1 与Page2 相对映的值。 <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Page1</th> <th style="text-align: center;">Page2</th> <th style="text-align: center;">灰度</th> </tr> </thead> <tbody> <tr> <td colspan="3" style="text-align: center;">-----</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">Level1</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">Level2</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">Level3</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">Level4</td> </tr> </tbody> </table>	Page1	Page2	灰度	-----			0	0	Level1	1	0	Level2	0	1	Level3	1	1	Level4	1h	R/W
Page1	Page2	灰度																			

0	0	Level1																			
1	0	Level2																			
0	1	Level3																			
1	1	Level4																			
3-2	在双层模式下图层逻辑关系 0 0: Page1 RAM “OR” Page2 RAM 0 1: Page1 RAM “XOR” Page2 RAM 1 0: Page1 RAM “NOR” Page2 RAM 1 1: Page1 RAM “AND” Page2 RAM	0h	R/W																		
1-0	设定Read/ Write 要在哪一个图层运行 0 0: 存取Page0 (512B SRAM)的Display Data RAM 0 1: 存取Page1 (9.6KB SRAM)的Display Data RAM 1 0: 存取Page2 (9.6KB SRAM)的Display Data RAM 1 1: 同时存取Page1 和Page2 的Display Data RAM	1h	R/W																		

HS240128S1 提供了双图层的功能，可经由缓存器REG[12h]来做设定，并提供4种(OR, NOR, XOR 和 AND)图层显示模式，供使用者设定选用。实际的显示效果，请参考图5-8。

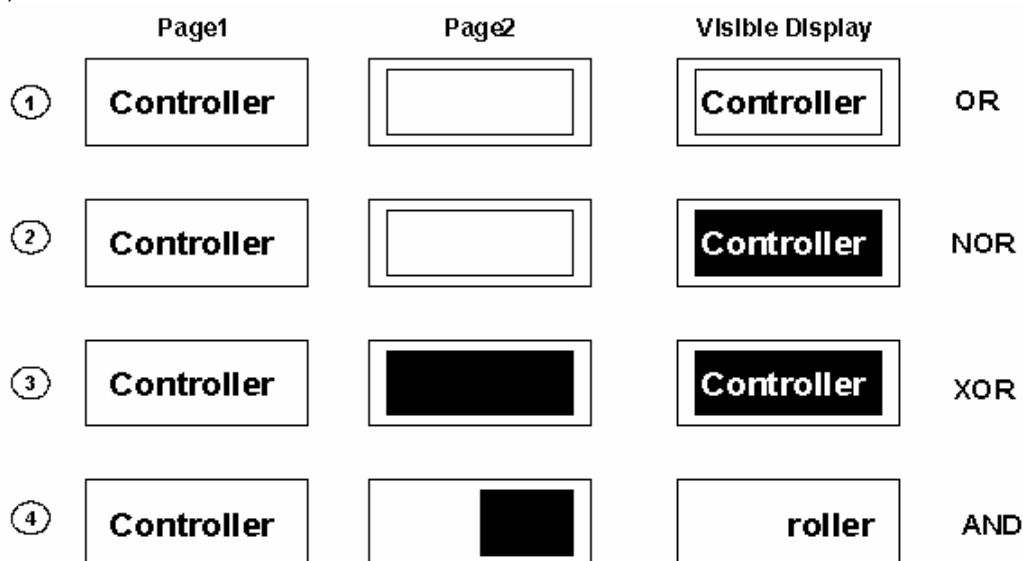


图5-8: 图层显示关系与效果

REG [20h] Active Window Right Register (AWRR)

Bit	Description	Default	Access
7-6	保留	0h	R
5-0	设定工作窗口(Active window)右边位置 Segment-Right	27h	R/W

REG [30h] Active Window Bottom Register (AWBR)

Bit	Description	Default	Access
7-0	设定工作窗口(Active window) 底边位置 Common-Bottom	EFh	R/W

REG [40h] Active Window Left Register (AWLR)

Bit	Description	Default	Access
7-6	保留	0h	R
5-0	设定工作窗口(Active window)左边位置 Segment-Left	0h	R/W

REG [50h] Active Window Top Register (AWTR)

Bit	Description	Default	Access
5-0	设定工作窗口(Active window)顶边位置 Common-Top	0h	R/W

REG [20h, 30h, 40h, 50h] 四个指令寄存器可作为换行/换页的功能，可让使用者利用这4个Register自行设定一个区块为工作窗口。当数据超过窗口的右边界REG [20h, 30h, 40h, 50h] 所设定的值，光标会自动换行(也就是光标会回到工作窗口的左边界REG[40h]所设定的值)，继续将数据写入。当数据写入到工作窗口的右下角时（REG[20h, 30h]所设定的值），会自动把光标移到工作窗口的左上角(REG[40h, 50h] 所设定的值)，继续的将数据填入窗口。

在HS240128S1中，这四个指令寄存器所设定的窗口不能超过240x128，在使用过程中，这四个指令寄存器划分的区域就是工作区域窗口，如果要全屏操作则设置为240x128。

REG [21h] Display Window Right Register (DWRR)

Bit	Description	Default	Access
7-6	保留	0h	R/W
5-0	设定显示窗口(Display Window)右边位置 Segment-Right Segment_Right = (Segment Number / 8) - 1 HS240128S1此参数设置为 $(240 / 8) - 1 = 29 = 1Dh$	27h	R/W

REG [31] Display Window Bottom Register (DWBR)

Bit	Description	Default	Access
7-0	设定显示窗口(Display Window) 底边位置 Common_Bottom Common_Bottom = LCD Common Number - 1 HS240128S1此参数设置为 $160 - 1 = 159 = 9Fh$ (不是127)	EFh	R/W

REG [41] Display Window Left Register (DWLR)

Bit	Description	Default	Access
7-0	设定显示窗口(Display Window) 左边位置 Segment-Left 通常将此缓存器的值设定为“0h”。	0h	R/W

REG [51] Display Window Top Register (DWTR)

Bit	Description	Default	Access
7-0	设定显示窗口(Display Window) 顶边位置Common-Top 通常将此缓存器的值设定为“0h”。	0h	R/W

REG[21h, 31h, 41h, 51h]是用来设定显示窗口。

注意! HS240128S1是根据这四个指令寄存器的值来计算和设定驱动波形的，而控制器是专门为240x160点阵设计的，为了不影响显示效果，**所以这里这四个设置必须按照240x160来初始化**，然后在前面提到的Active Window里面设置为240x128来使用。

Note: 寄存器的设定，请遵照以下的规范：

1. DWRR ≥ AWRR ≥ CPXR ≥ AWBR ≥ DWBR
2. DWLR ≥ AWLR ≥ CPYR ≥ AWTR ≥ DWBR

REG [60h] Cursor Position X Register (CPXR)

Bit	Description	Default	Access
7-6	保留	0h	R
5-0	设定光标Segment 地址	0h	R/W

REG [70h] Cursor Position Y Register (CPYR)

Bit	Description	Default	Access
7-0	设定光标Common地址	0h	R/W

寄存器[60h]CPXR 的 Bit[5..0]用来设定光标的 Segment 地址，光标的 Segment 地址是以每 8-Bit 为单位，例如，想在 LCD Panel 的左上角秀出“控”，则必须设定光标寄存器 CPXR = 00h，CPYR = 00h，又例如想在 Panel 的左上角第三个全角位置秀出“制”，则必须设定光标寄存器 CPXR = 04h，CPYR = 00h，同理，想在 Panel 的左上角第二行第一个全角位置秀出“器”，则必须设定光标寄存器 CPXR = 00h，CPYR = 10h，请参考图 5-10。

不论文字或是绘图模式，都是使用寄存器[60h]CPXR 与[70h]CPYR 来设定光标的地址。

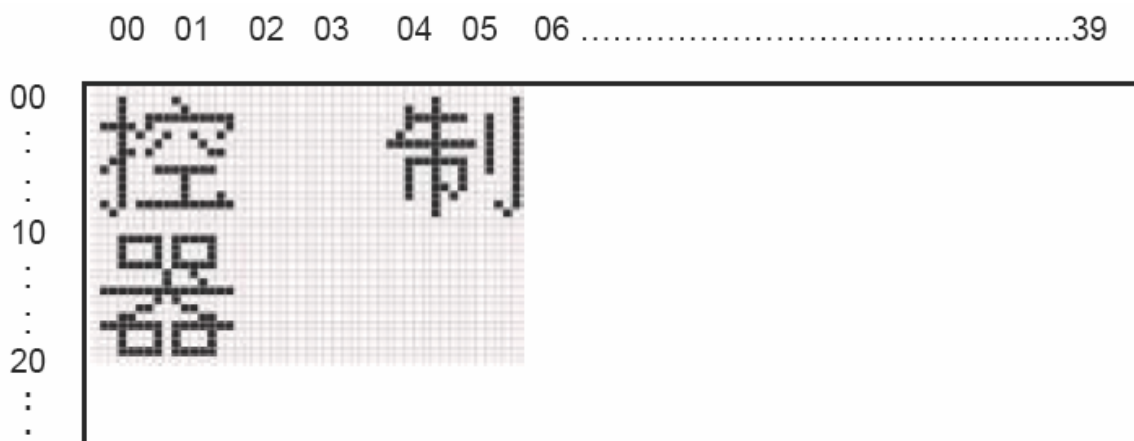


图5-10: 光标位置设置示例

REG [61h] Begin Segment Position Register (BGSF)

Bit	Description	Default	Access
7-6	保留	0h	R/W
5-0	显示 Segment 开始的位置	0h	R/W

REG[61h]设置Segment起始边界的值，一般使用默认值0

REG [71h] Shift action range, Begin Common Register (BGCM)

Bit	Description	Default	Access
7-0	在水平移动模式下，设定区块移动的起始Common位置	0h	R/W

REG [72h] Shift action range END Common Register (EDCM)

Bit	Description	Default	Access
7-0	在水平移动模式下，设定区块移动的结束Common位置	0h	R/W

在屏幕所显示的文字可以作水平移动，须由寄存器[03h]来做设定。该项功能可达到左右的水平移动，每次移动的刻度为 1 个 Byte。另外，还可透过寄存器[71h, 72h]来设定屏幕的区块文字水平移动。



```
WriteCommand(0x80,0x05); //设定水平或垂直卷动速度
WriteCommand(0x71,0x00); //设定 REG[71]区块 Y1 坐标
WriteCommand(0x72,0x00); //设定 REG[72]区块 Y2 坐标
WriteCommand(0x03,0x83); //设定 REG[03]:bit[1,0]="11"
//此时屏幕将以 Y1/Y2 设定的区块做水平卷动
```

在屏幕所显示的文字可以作垂直卷动，须由寄存器[03h]来做设定。该项功能可达到上下的垂直卷动，每次移动的刻度为1 个像素(Pixel)。如图5-12 所示，可作卷动的效果



图5-12: 垂直转动效果

REG [80h] Blink Time Register (BTMR)

Bit	Description	Default	Access
7-0	光标闪烁时间设定 闪烁时间 = [80h]Bit[7..0] x (1/Frame_Rate) Frame Rate 的设定是通常依照LCD 面板所提供的最佳值。	33h	R/W

REG [81h] Frame Rate Polarity Change at Common_A Register (FRCA)

Bit	Description	Default	Access
7-0	在 N_line inversion 模式下, 可设定 FRM 要变换极性的起始 Common 位置	0h	R/W

REG [91h] Frame Rate Polarity Change at Common_B Register (FRCB)

Bit	Description	Default	Access
7-0	在 N_line inversion 模式下, 可设定 FRM 要变换极性的结束 Common 位置	0h	R/W

REG [90h] Shift Clock Control Register (SCCR)

Bit	Description	Default	Access
7-0	设定 XCK 讯号周期 $SCCR = (SCLK \times DW) / (Seg \times Com \times FRM)$ SCLK: HS240128S1 系统频率(System Clock) (单位: Hz) DW: LCD 驱动器的Data Bus 宽度(单位: Bit) Seg: LCD 面板的Segment 大小(单位: Pixel) Com: LCD 面板的Common 大小 (单位: Pixel) FRM: LCD 面板的Frame Rate(单位: Hz) 限制条件: LCD 的Data Bus 为4it, $SCCR \geq 4$	4h	R/W

HS240128S1建议使用System Clock 8Mhz, XCK信号周期设置为0x0C

1. 系统频率(SCLK) = 8MHz
 2. LCD 驱动器 的Data Bus 宽度(DBW) = 4Bit
 3. 使用240 x 128的LCD 面板, Column = 240, Row = 160 (按照240x160来初始化)
 4. LCD 面板的Frame Rate 为70Hz
- 则 $SCCR = (8MHz \times 4) / (240 \times 160 \times 70) = 11.9$
 所以建议设定 $SCCR = 12 = 0Ch$

REG [A0h] Interrupt Setup & Status Register (INTR)

Bit	Description	Default	Access
7	保留	0h	
6	保留	0h	
5	光标行(Column)状态 1: 光标的Column 等于缓存器INTX 0: 光标的Column 不等于缓存器INTX	0h	R (Read Clear)
4	光标列(Row)状态 1: 光标行等于缓存器INTY 0: 光标行不等于缓存器INTY	0h	R (Read Clear)
3	Key Scan 中断屏蔽控制 1: 致能Key Scan 中断, 或致能BUSY(忙碌讯号)。 0: 禁能Key Scan 中断	0h	R/W
2	触控屏幕中断屏蔽 1: 如果触控屏幕被侦测到, 则产生中断输出。或致能BUSY。 0: 如果触控屏幕被侦测到, 亦不产生中断输出。	0h	R/W
1	设定缓存器[B0h]INTX 是否发生中断 1: 致能INTX 中断 0: 禁能INTX 中断	0h	R/W
0	设定缓存器[B1h]INTY 是否发生中断 1: 致能INTY 中断 0: 禁能INTY 中断	0h	R/W

REG [A1h] Key Scan Controller Register (KSCR)

Bit	Description	Default	Access
7	Key Scan 的致能控制位 1: 致能	0h	R/W

	0: 禁能		
6	Key San 的数组选择 1: Key Scan 为 8x8 数组 0: Key Scan 为 4x8 数组	0h	R/W
5-4	KeyScan 的扫描周期 0 0: 2 倍的Key Scan 扫描周期 0 1: 4 倍的Key Scan 扫描周期 1 0: 8 倍的Key Scan 扫描周期 1 1: 16 倍的Key Scan 扫描周期	0h	R/W
3	保留	0h	R/W
2-0	Key Scan 的扫描周期选择 0 0 0: 2 倍(LP peak to peak period) 0 0 1: 4 倍(LP peak to peak period) 0 1 0: 8 倍(LP peak to peak period) 0 1 1: 16 倍(LP peak to peak period) 1 0 0: 32 倍(LP peak to peak period) 1 0 1: 64 倍(LP peak to peak period) 1 1 0: 128 倍(LP peak to peak period) 1 1 1: 256 倍(LP peak to peak period)	0h	R/W

REG [A2h] Key Scan Data Register (KSDR)

Bit	Description	Default	Access
7-0	Key Scan KC[7~0] 的输出值	0h	RO

REG [A3h] Key Scan Data Register (KSDR)Key Scan Data Expand Register (KSER)

Bit	Description	Default	Access
7-0	Key Scan KR[7~0] 的输入值	0h	RO

REG [B0h] Interrupt Column Setup Register (INTX)

Bit	Description	Default	Access
7-6	保留	0h	R
5-0	设定行 (Column) 地址中断 假如光标位置 X 缓存器(CPXR)=INTX, 中断发生。	27h	R/W

REG [B1h] Interrupt Row Setup Register (INTY)

Bit	Description	Default	Access
7-0	设定列 (Row) 地址中断 假如光标位置Y 缓存器(CPYR)=INTY, 中断发生。	EFh	R/W

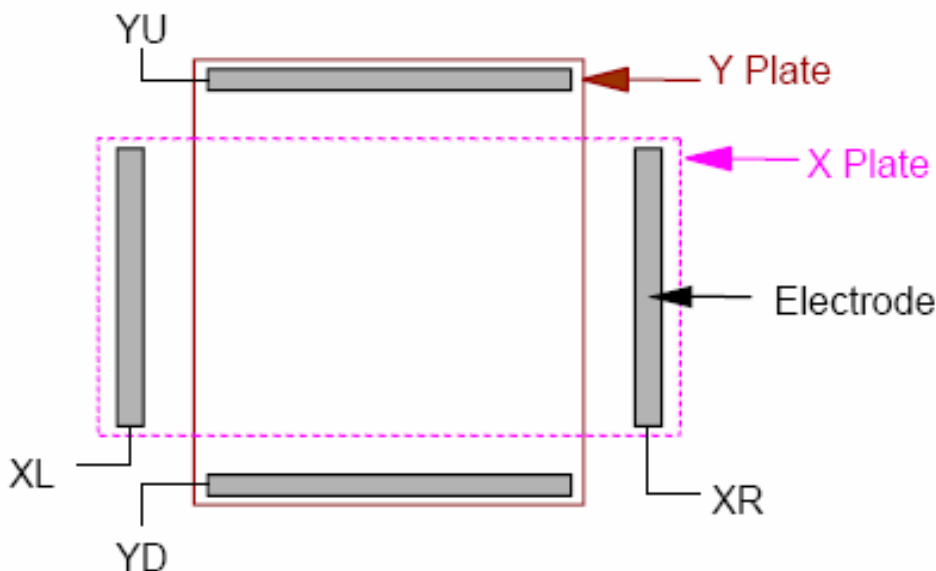
REG [C0h] Touch Panel Control Register (TPCR)

Bit	Description	Default	Access
7	触摸屏功能启动 1: 致能 0: 禁能	1h	R/W

6	触摸屏数据输出 1: 致能触摸屏数据输出 0: 禁能触摸屏数据输出	1h	R/W
5	保留	0h	R/W
4	触摸屏扫瞄 1: 禁能 0: 致能	1h	R/W
3-0	触摸屏控制位 Bit3: 控制SW3 ON/OFF (1/0), Bit2: 控制SW2 ON/OFF (1/0) Bit1: 控制SW1 ON/OFF (1/0), Bit0: 控制SW0 ON/OFF (1/0)	见图五-3-12	R/W

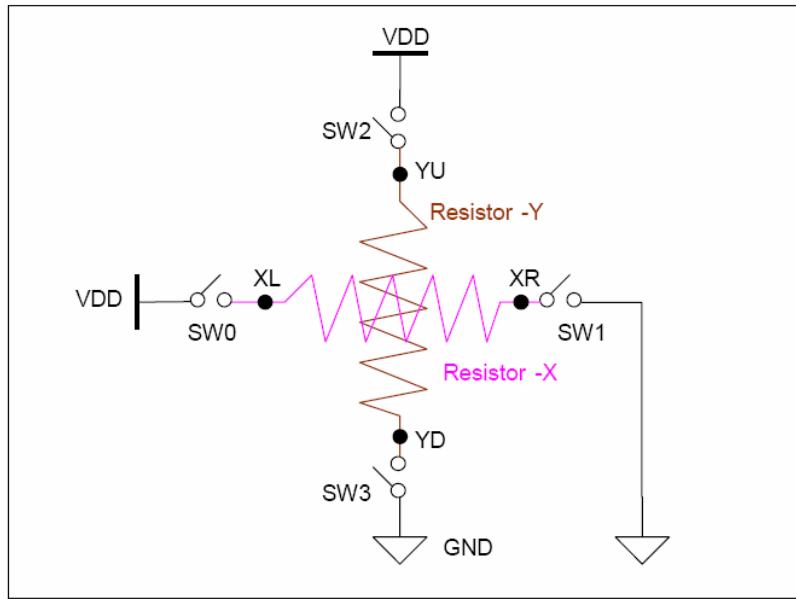
RA8803/RA8822内建了一个10-bit 模拟-数字转换器(Analog to Digital Converter, ADC)及数个模拟开关(Analog Switch), 使用者可以将四线电阻式触摸式面板的XL, XR, YU, YD 接到RA8803/8822, 然后利用模拟开关切换让ADC读取电阻上的电压值, 再由MPU 读取ADC 的转换值, 而得到触摸面板Touch 的相对位置。

电阻式触摸面板是由两层极薄的电阻面板组成, 如图五-3-11, 两层面板之间有一个很小的间距, 当有外力在面板上的某一点压下去时, 会在施力点造成两层电阻接触, 也就是短路(Short), 而两层电阻面板的端点都各有电极, 如图五-3-12: 触摸屏所示YU, YD, XL, XR, 因此配合一些开关就可检测出面板上哪一相对置被Touch。

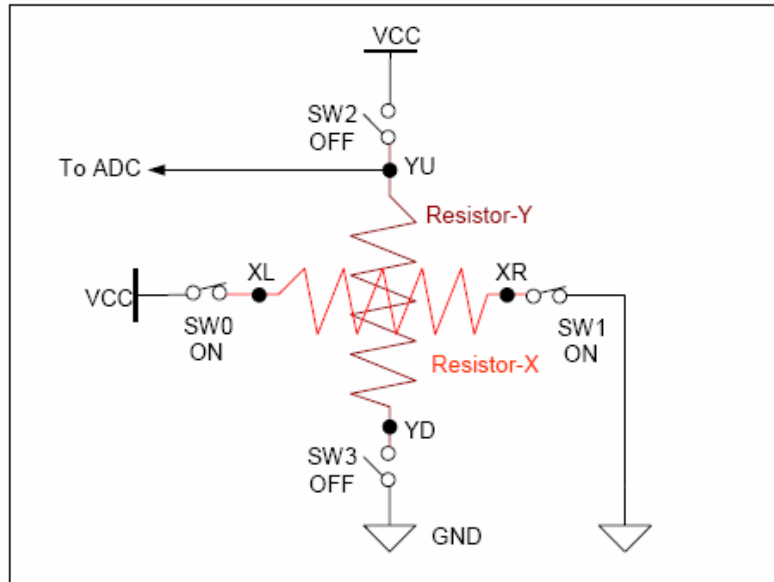


图五-3-11: 触摸屏

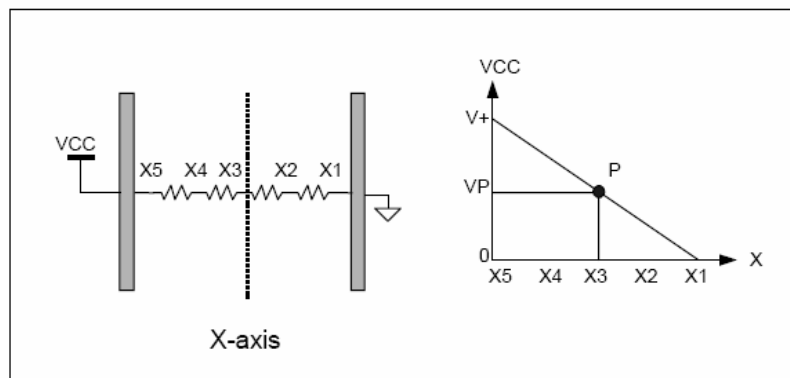
在图五-3-13触摸屏中, 设定开关SW2 与SW3 是OFF(Open), SW0 与SW1 是ON(Close), 当有外力在面板上的某一点压下去时, 由YU 点取得电压接到ADC(Analog to Digital Converter), 就可以得到被Touch 点的X 坐标相对位置。因为开关SW2 与SW3 是OFF, 因此YD 点是Floating, 所以当有外力在面板上的某一点压下去时, YU 上的电压事实上就是X 的Panel(也就是电阻)上的分压结果, 压在面板上的不同一点, 就会得到不同的分压值, 如图五-3-14所示。



图五-3-12: 触摸屏与检测开关

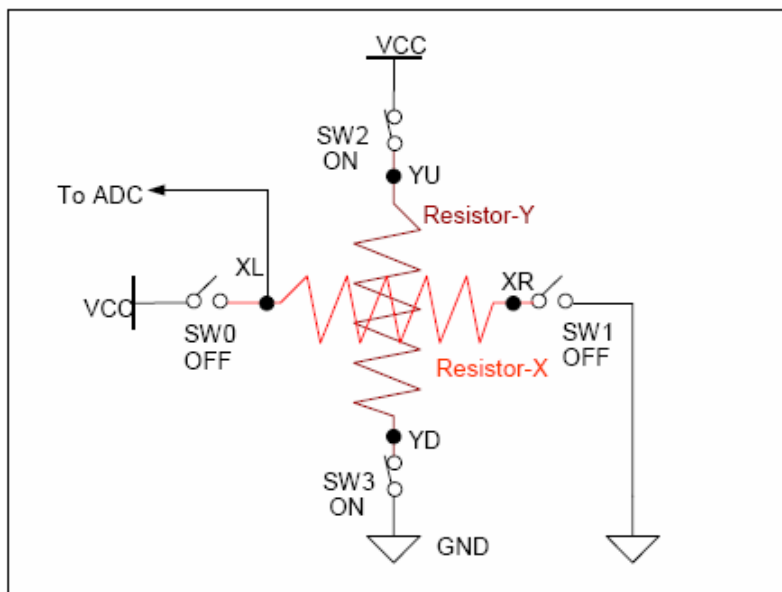


图五-3-13: 读取X坐标



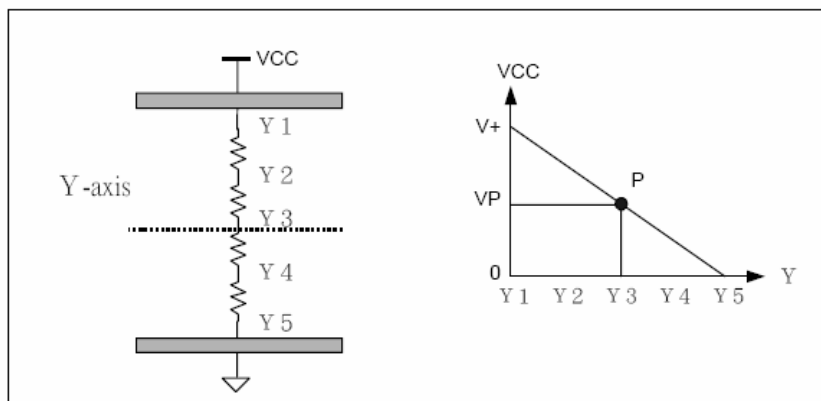
图五-3-14: Resistor-X 的分压

同理，在图五-3-15中，设定开关SW0 与SW1 是OFF(Open)，SW2 与SW3 是ON(Close)，当有外力在面板上的某一点压下去时，由XL 点取得电压接到ADC(Analog to Digital Converter)，就可以得到被Touch 点的Y 坐标相对位置。一般说来许多触摸面板都是贴在LCD 面板上面，因此在程序设计上如果重复图五-3-13 与图五-3-15的读取步骤就可以顺利得知被Touch 的点是在屏幕上的哪一位置。



图五-3-15: 读取Y坐标

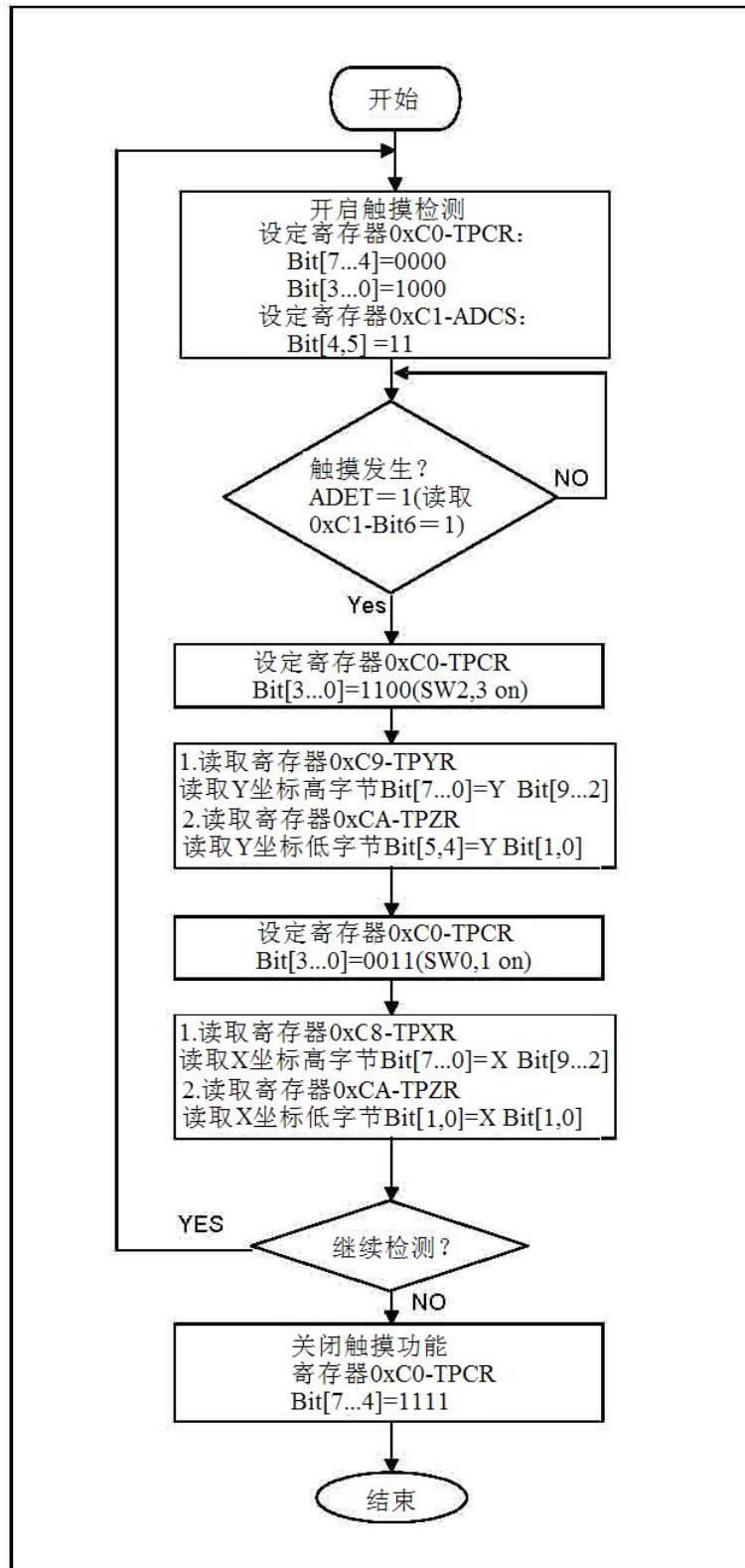
在图五-3-15中，因为开关SW0 与SW1 是OFF，因此XR 点是Floating，所以当有外力在面板上的某一点压下去时，XL 上的电压事实上就是Y 的Panel(也就是电阻)上的分压结果，压在面板上的不同一点，就会得到不同的分压值，如图五-3-16所示。



图五-3-16: Resistor-X 的分压

图五-3-18的流程图则是说明RA8803触摸式面板读取的控制方式，与触摸式面板有关的缓存器为TPCR、TPXR、TPYR 与TPSR (ADCS)，在使用触摸式面板时必须先将触摸屏功能开启，缓存器TPCR 的Bit-7 与Bit-6 设为“1”，同时TPCR 的Bit[3..0] 设为“1000”，也就是Switch SW3 为On的状态，然后程序可以侦测缓存器TPSR 的Bit-6 是否为“1”，如果缓存器TPSR 的Bit-6 为“1”，则表示触摸式面板目前被Touch，请参考图6-8。在侦测阶段时，缓存器TPCR 的Bit-7 与Bit-6 可以先为“0” (ADC Disable)，如果程序侦测到缓存器TPSR 的Bit-6 为“1”，表示触摸式面板目前被Touch，然后再将ADC Enable -- 缓存器TPCR 的Bit-7与Bit-6 设为“1” 也可以，如此可避免触摸式面板未被Touch 而让ADC 动作产生不

必要的耗电。



图五-3-17: 触摸屏的检测流程

REG [C1h] ADC Status Register (TPSR/ADCS)

Bit	Description	Default	Access
-----	-------------	---------	--------

7	ADC 数据转换完成指示 1: ADC 数据转换已完成 0: ADC 数据转换未完成	0h	R/W
6	触摸事件的检测指示 1: 有被触摸 0: 没被触摸	0h	R/W
5-4	必须在系统使用时设为 “1 “	0h	R/W
3-2	设定ADC 的时脉转换速度 0 0: SCLK/32 0 1: SCLK/64 1 0: SCLK/128 1 1: SCLK/256	2h	R/W
1-0	保留	2h	R/W

REG [C8h] Touch Panel Segment High Byte Data Register (TPXR)

Bit	Description	Default	Access
7-0	储存触控屏幕行的高字节 (bit9~2) 数据	80h	RO

REG [C9h] Touch Panel Common High Byte Data Register (TPYR)

Bit	Description	Default	Access
7-0	储存触控屏幕列的高字节 (bit9~2) 数据	80h	RO

REG [CAh] Touch Panel Segment/Common Low Byte Data Register (TPZR)

Bit	Description	Default	Access
7-6	储存触控屏幕行的低字节 (bit1~0) 数据	0h	RO
5-4	保留	0h	--
3-2	储存触控屏幕列的低字节 (bit1~0) 数据	0h	RO
1-0	保留	0h	--

X1	连接于电阻式触控屏幕的左边端点XL。
X2	连接于电阻式触控屏幕的右边端点XR
X3	连接于电阻式触控屏幕的上边端点YU
X4	连接于电阻式触控屏幕的下边端点YD

REG [E0h] Pattern Data Register (PNTR)

Bit	Description	Default	Access
-----	-------------	---------	--------

7-0	<p>(1) 设定写入到DDRAM 的数据 当缓存器[F0h]的bit3 为 ‘1’，HS240128S1内部将自动读取本缓存器[E0h] 的Data, 然后全部填写到DDRAM 内, 之后缓存器[F0h]的bit3 被清除为 ‘0’。</p> <p>(2) Display Times of Gray Mode 在于灰阶模式下(缓存器MAMR bit[6..4] = 000) ，此缓存器用来控制灰阶显示效果，“1”与“0”的数目代表显示比率，请参考下面的说明。</p>	0h	R/W
-----	--	----	-----

REG [F0h] Font Control Register (FNCR)

Bit	Description	Text/Graph	Default	Access
7	<p>字型ROM 的转换电路控制 1: 致能 0: Bypass</p>		1h	R/W
6	<p>字型ROM 的地址空间选择 当bit5~4 设定 “00”→ ROM Mode0, 该位可以用来选择上或下的256KB ROM 的地址空间。 1: 选择下部256KB 字型ROM 0: 选择上部256KB 字型ROM</p>		0h	R/W
5-4	<p>字型ROM 的字型选择 0 0: 选择简体 (GB) 字型 (256KB, Mode0) 0 1: 选择繁体 (BIG5) 字型 (512KB, Mode1) 1 0: 选择简体 (GB) 字型 (512KB, Mode2)</p>		1h	R/W
3	<p>填写PNTR 的数据到DDRAM 1: 开始写入 0: 未动作 当FDA 为 ‘1’，HS240128S1 内部将自动读取PNTR 的Data ，填写到DDRAM 内(Range:[AISR, AICR] ~ [AXSR, AXCR]), 之后此位会被自动清除为 ‘0’。</p>	Graph	0h	R/W
2	<p>强制为ASCII 解码 1: 所有输入的Data, 都以ASCII 解码(00~FFh) 0: RA8803/22 会先检视输入Data 的第一个字节, 介于00~9Fh, 视为ASCII (半角字) A0~FFh, 视为GB/BIG5 (全角字)</p>	Text	0h	R/W (Auto Clear)
1-0	<p>4 种ASCII 区块选择 0 0: ASCII 选择区块0, Latin_1 0 1: ASCII 选择区块1 , Latin_2 1 0: ASCII 选择区块2 , Latin_3 1 1: ASCII 选择区块3, Latin_4</p>		2h	R/W

寄存器[F0h]是用来设定与字型ROM 相关的功能，Bit6 就是如前面所述用来选择显示的字形是使用内部字型ROM 或是外部字型ROM。当使用者HS240128S1时，必须将Bit[5..4]设成“10”才能正确显示简体字型。ASCII 区块选择是为了提供更多的可以选择 ASCII 图案，可以做一些小动画，比如鼠标等待、时钟运行等。而当使用 ASCII 显示英文和数字的时候，必须选择区块 0，否则无法正常显示。

b5-b0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
b7-b4																
0000	€		ƒ
0001		€	ƒ
0010		ı	€	ƒ
0011	°	ı	€	ƒ
0100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ï	Î	Ì	Í
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																
1010	°	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ï	Î	Í
1011	°	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ï	Î	Í	Ð
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
1111	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

ASCII区块1

b3-b0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
b7-b4																
0000																
0001																
0010		À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î
0011	°	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î
0100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
0101		Ñ	Ò	Ó	Ô	Õ	Ö	×	Ü	Ý	Û	Ü	Ý	ß		
0110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
0111	ñ	ò	ó	ô	õ	ö	÷	ü	ý	Û	Ü	Ý	ß			
1000																
1001																
1010		À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î
1011	°	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ü	Ý	Û	Ü	Ý	ß		
1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
1111	đ	ñ	ò	ó	ô	õ	÷	ø	ü	ý	Û	Ü	Ý	ß		

ASCII区块2

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

ASCII区块3

REG [F1h] Font Size Control Register (FVHT)

Bit	Description	Default	Access
7-6	设定字型水平的大小 0 0: 一倍 0 1: 二倍 1 0: 三倍 1 1: 四倍	0h	R/W
5-4	设定字型垂直的大小 0 0: 一倍 0 1: 二倍	0h	R/W

	1 0: 三倍 1 1: 四倍		
3-0	保留	Fh	R/W

HS240128S1 内建有512KByte 的中文显示字型ROM(Font ROM)，全角16x16 中文与8x16 的ASCII 半型字型。除了内建的8x16 和16x16 的字号外，还提供字型放大的功能，可利用REG[F1h]bit7~4 的设定，将显示字号放大到32x32 或48x48，64x64。下图是表示16x16 的字型放大到32x32。

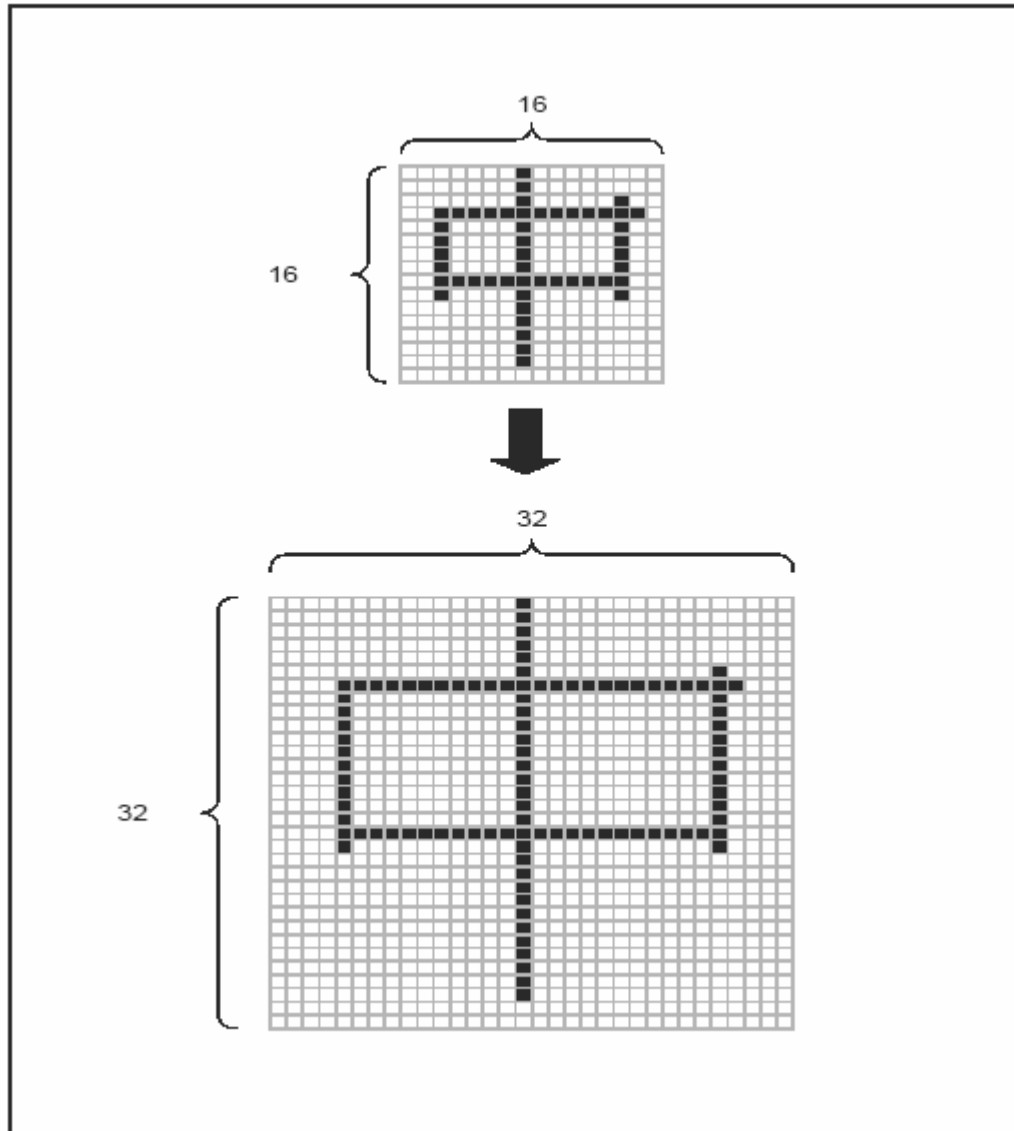


图5-13: 字体放大效果

屏幕显示与 DDRAM 地址映射关系（DDRAM 地址表）

	SEG1~SEG8	SEG9~SEG16	SEG205~SEG212	SEG213~SEG240
COM1	00H	01H	1CH	1DH
COM2	1EH	1FH	3AH	3BH
.....
COM127	2274H	2275H	2290H	2291H
COM128	2292H	2293H	257FH	2580H

示例接口电路：（直接访问）

