

# OCMJ4X9E 中文图形点阵液晶显示模块 使用说明书

---

感谢您关注和使用我们的中文系列液晶显示器产品，欢迎您提出意见和建议，我们将竭诚为您服务、让您满意。您可以浏览 <http://www.shsixian.com> 了解最新的产品与应用信息，或拨打热线电话 **021-53083613** 及向 [sx@shsixian.com](mailto:sx@shsixian.com) 邮箱发 E-mail 获取具体的技术咨询与服务。

**上海思先电子有限公司**

**Shanghai Sixian Electronics Co; Ltd.**

# 目录

1. 简介.....	3
2. 特性.....	3
3. 机械特性.....	3
4. 引脚描述.....	4
5. 脚位定义.....	5
6. 缓存器描述.....	8
7. 微控制器接口.....	20
8. 显示功能.....	31
9. 电气特性.....	36
10. 模块外型尺寸.....	43

## 1. 简介

4x9E 是一个中英文文字与绘图模式的点矩阵液晶显示模块,内建 256KByte 的 ROM 字形码,可以显示中文字型、数字符号、英日欧文等字母,并且内建 1170Byte 的显示内存,显示 144 x64 点阵的 LCD Panel,同时支持上下左右的旋转功能,另外并提供 450Byte 的卷动 Buffer,达到卷动时画面不断平移更新的效果。在文字模式中,4x9E 可接收标准中文文字内码直接显示中文,而不需要进入绘图模式以绘图方式描绘中文,可以节省许多微处理器时间,提升液晶显示中文之处理效率。

在 MPU 接口方面,4x9E 支持 8080/6800 系列之 MPU,提供 4-Bit 或 8-Bit 的数据总线接口,另外也支持 3 线、4 线及 IIC 共四种的串行接口,当显示 16x16 的中文全型字号时,可秀出 4 行 9 列的中文字,同时内建的 256Byte CGRAM 提供了自行造字的功能。因此在系统开发时,可选用成本较低的 MPU 作为搭配,不仅在快速、便捷、好用,更能节省许多的开发成本。

## 2. 特性

模块控制器为 RA8816

支持文字与绘图两种混和显示模式

支持 8080/6800 8/4-bit MPU 接口和 3-wire 或 4-wire 同步串行接口

支持 IIC 同步串行接口

内建 256KB 字型 ROM: 一级常用字库--中文繁体或简体、英文、日文、ASCII、欧文--拉丁语系(Latin, Latin-ext A, Latin-ext B)

支持 ASCII 8x8 英文字型, 8x16 半角, 16x16 全角简体中文或是繁体文

最大可支持 144Seg x 64Com LCD 面板: 4 行 9 列个中文字(16x16 字型), 或 8 行 x18 个英文字(8x8 字型)

内建 1170 Bytes 显示内存(Display RAM)以及 450Byte 的卷动 Buffer

支援 BIG5 或 GB 码

支持水平及垂直卷动功能

内建 256Byte SRAM 可自行造字

内建 32-Steps 亮度调整控制

电源操作范围: 有 3.3V 和 5V 两种类型模块

## 3. 机械特性

类别	标准值	单位
模块	90.0 (w) X60.0(h)X11.5(t)Max	mm
有效显示区	69.30(w)X37.20(h)	mm
点大小	0.45(w)X0.52(h)	mm
点间隙	0.02(w)X0.04(h)	mm

## 4. 引脚描述

引脚编号	引脚名称	方向	引脚功能描述
1	VSS	--	逻辑电源地 (0V)
2	VDD	--	逻辑电源正 (3.3V; 亦可生产提供用 5V 电源的模块)
3	/RST	I	复位信号, 低电平有效
4	/INT	O	中断讯号(Interrupt Signal), 中断发生为低
5	BIT4	I	并口数据位选择(Data Bit Select)
6	/CS	I	芯片选取(Chip Select), 低电平有效
7	RS	I	显存数据和缓存器选择, 1 为显存数据, 0 为缓存器
8	/WR	I	写控制讯号(Write Control or Read-Write Control)
9	/RD	I	读控制讯号(Read Control or Enable)
10	DB0	I/O	数据 0
11	DB1	I/O	数据 1
12	DB2	I/O	数据 2
13	DB3	I/O	数据 3
14	DB4	I/O	数据 4
15	DB5	I/O	数据 5
16	DB6	I/O	数据 6
17	DB7	I/O	数据 7
18	LED+	--	背光电源正 (5V)
19	LED-	--	背光电源负 (0V)

\*引脚的复用功能与具体应用请参考下一节。

## 5. 脚位定义

Pin Name	I/O	Description
<b>DB[7..0]</b> <b>DB0: SCK</b> <b>DB1: SDA/SDO</b> <b>DB2: RS/SDI</b> <b>DB3: CS</b> <b>DB[7..6]:SMOD</b>	I/O	<b>数据总线(Data Bus)</b> 负责模块与微处理器(MPU)之间做数据传送与接收。 当 MPU 为 8 位模式下, DB[7..0]全部有效, 当 MPU 为 4 位模式下, 只有 DB[3..0]为有效, 高字节 DB[7..4]无效需浮接。 当焊点 P/S = 1 时 (板上有焊点选择 P, 出厂时为 1), 此时为平行并列接口, DB[7..0] 为地址/数据传输线, 当焊点 P/S = 0 时 (板上有焊点选择 S), MPU 与 4X9E 之接口为串行模式(Serial Mode), 此时 DB[7..6](SMOD[1..0])为输入脚位用来决定哪一种串行模式设定。
		<b>SMOD : 串行接口模式</b> <b>0 0</b> : IIC 传输(2-Wire) , 使用到 SCK, SDA <b>0 1</b> : 三线式传输(3-Wire) , 使用到 SCK, SDA, CS <b>1 0</b> : 四线式传输(4-Wire, A-Type) , 使用到 SCK, SDA, RS, CS <b>1 1</b> : 四线式传输(4-Wire, B-Type) , 使用到 SCK, SDO, SDI, CS
	在串行模式下, 信号皆由 DB[3..0]来定义, 说明如下:	
	I/O	<b>SCK(DB0)</b> : 串行时脉-- Serial Clock。 <b>SDA(DB1)</b> : 双向串行数据-- Bi-direction Mode Serial Data。 <b>SDO(DB1)</b> : 串行数据输出 -- Data Out。 <b>RS(DB2)</b> : 内存/缓存器周期选择-- Memory/Register Cycle Select。 <b>SDI(DB2)</b> : 串行数据输入-- Serial Data In。 <b>/CS(DB3)</b> : 芯片选取 -- Chip Select, 低电位时动作。
		在 IIC 串行模式下, /WR、/RD 与 DB[5..2]同时作为组件的地址设定信号 IICA[5..0], 避免与系统上的其它 IIC 组件冲突。 在串行模式, 没使用到的数据线请保持浮接(NC)。
<b>/RD</b> <b>(EN)</b>	I	<b>控制讯号(Read Control or Enable)</b> 当使用 8080 系列的 MPU 时, /RD 为数据读取讯号, 在低电位动作。 当使用 6800 系列的 MPU 时, /RD 为 Enable 讯号, 在高电位动作。 在 IIC 串行模式下, /RD 作为组件的地址设定信号 IICA[4], 而三/四线式串行模式时此脚位不被使用, 请接到 VDD。
<b>/WR</b> <b>(R/W)</b>	I	<b>控制讯号(Write Control or Read-Write Control)</b> 当 MPU 为 8080 系列时, 此脚位为数据写入讯号(/WR), 于低电位动作。 当 MPU 为 6800 系列时, 此脚位为读取/写入讯号(R/W), 高电位时表示为 读取的动作, 低电位时表示为写入的动作。 在 IIC 串行模式下, /WR 作为组件的地址设定信号 IICA[5], 而三/四线式串行模式时此脚位不被使用, 请接到 VDD。

<b>(D/ C) RS</b>	<b>I</b>	<b>控制讯号(Data/Command Select or Register Select)</b> 当 MPU 为 8080 系列时, 此脚位为 Data 与 Command 的选择信号, D/ C 为 0 时表示缓存器存取周期(Command Cycle), 为 1 表示数据存取周期(Data Cycle)。 当 MPU 为 6800 系列时, 此脚位为 RS 信号, RS 为 0 时表示缓存器存取周期, 为 1 表示数据存取周期。 在串行模式此脚位不被使用, 请接到 VDD。
<b>/CS</b>	<b>I</b>	<b>芯片选取(Chip Select)</b> 当 CS 是 Low 时, 模块会处于致能, 可接受指令, 反之, 则不可接受指令。 在串行模式此脚位不被使用, 请接到 VDD。
<b>/INT</b>	<b>O</b>	<b>中断讯号(Interrupt Signal)</b> 中断信号, 连接 MPU 端以回报模块的状况。平时为 High, 设定中断发生时为 Low。
<b>BIT4</b>	<b>I</b>	<b>并口数据位选择(Data Bit Select)</b> 当 MPU 接口为并列模式时, 此脚位用来选择 MPU 接口的数据传输是 8 位或 4 位。 BIT4 = 0--- 8 位数据传输接口。 BIT4 = 1--- 4 位数据传输接口(Default)。 在串行模式此脚位不被使用, 请接到 VDD。
<b>/RST</b>	<b>I</b>	<b>外部重置信号(Reset)</b> RST = 0 : 模块将被初始化。 RST = 1 : 正常状态, 已内建 Pull-High 电阻。
<b>VSS</b>		电源负 0V
<b>VDD</b>		电源正 3.3V (亦可生产用 5V 电源的模块)
<b>LED-</b>		背光负 0V
<b>LED+</b>		背光正 5V
<b>下面三个通过板电阻位或焊点位选择其对应接口</b>		
<b>C86</b>		<b>MPU 选择(MPU Select)</b> 当 MPU 接口为并列模式时, 此脚位用来选择 MPU 接口是 8080 系列或 6800 系列。 C86 = 0 -- 8080 (默认, R17 已贴) 系列 MPU 接口。 C86 = 1 -- 6800 系列 MPU 接口(去除 R17)。 在串行模式将电阻 R17 移过 R18
<b>P/S</b>		<b>串并列选择(Parallel/Serial Select)</b> P/S = 0---MPU 接口为串行模式 (板上焊点 S), 请参考 DB[7..6] 的设定。 P/S = 1---MPU 接口为并列模式。(Default)
<b>J4</b>		背光控制选择 (默认背光开关受程序控制)

表 5-1: MPU 串、并列模式之使用脚位定义

Pin Name	I/O	并行模式				串行模式			
		8080		6800		IIC	三线式	四线式 (A-Typ)	四线式 (B-Typ)
		8Bit	4Bit	8Bit	4Bit				
DB7	I/O	DB7	—*1	DB7	—	0	0	1	1
DB6	I/O	DB6	—	DB6	—	0	1	0	1
DB5	I/O	DB5	—	DB5	—	IICA3	—	—	—
DB4	I/O	DB4	—	DB4	—	IICA2	—	—	—
DB3	I/O	DB3	DB3	DB3	DB3	IICA1	$\overline{CS}$	$\overline{CS}$	$\overline{CS}$
DB2	I/O	DB2	DB2	DB2	DB2	IICA0	—	RS	SDI
DB1	I/O	DB1	DB1	DB1	DB1	SDA	SDA	SDA	SDO
DB0	I/O	DB0	DB0	DB0	DB0	SCK	SCK	SCK	SCK
$\overline{RD}$ , EN	I	$\overline{RD}$	$\overline{RD}$	EN	EN	IICA4	1*2	1*2	1*2
$\overline{WR}$ , R/ $\overline{W}$	I	$\overline{WR}$	$\overline{WR}$	R/ $\overline{W}$	R/ $\overline{W}$	IICA5	1*2	1*2	1*2
D/ $\overline{C}$ , RS	I	D/ $\overline{C}$	D/ $\overline{C}$	RS	RS	1*2	1*2	1*2	1*2
$\overline{CS}$	I	$\overline{CS}$	$\overline{CS}$	$\overline{CS}$	$\overline{CS}$	1	1	1	1
C86	I	0	0	1	1	1	1	1	1
BIT4	I	0	1	0	1	1	1	1	1
P/ $\overline{S}$	I	1	1	1	1	0	0	0	0

Note1: “—” 表示此脚位不被使用，请保持浮接(NC)。

Note2: 串行模式时，没用到的并联接口信号要接到 1 或 VDD。

\*如要设 C86 为 1，则将电阻 R17 移过 R18。

\*P/S 为板上焊点选择。

## 6. 缓存器描述

### 6.1 缓存器总表

表 6-1: 缓存器总表

ID	缓存器简称	D7	D6	D5	D4	D3	D2	D1	D0	暂存器说明
0	DWFR	B/C	-	NW5	NW4	NW3	NW2	NW1	NW0	驱动波形设定
1	PWRR	SRST	MCLR	-	IO_IEN	KWK	IOWK	DOFF_Z	SLP	电源控制
2	SYSR	LS3	LS2	LS1	LS0	GB_EN	-	RS1	RS0	系统设定
3	MWMR	BMOD1	BMOD0	BIEN	ASCS	BOLD	INV	MD1	MD0	内存输入模式
4	CURCR	H3	H2	H1	H0	-	BLK	CR	CUR_EN	光标控制
5	X-CUR	-	-	X5	X4	X3	X2	X1	X0	光标 X 位置
6	Y-CUR	-	Y6	Y5	Y4	Y3	Y2	Y1	Y0	光标 Y 位置
7	KEYR	KSB	KDB1	KDB0	KSTB_SEL	K_AUTO	IRE	KF1/ KSTB1	KF0/ KSTB0	键盘扫描控制
	KSDR	SIRQ	KSTB1	KSTB0	KSD4	KSD3	KSD2	KSD1	KSD0	键盘扫描数据
SIRQ		AKD6	AKD5	AKD4	AKD3	AKD2	AKD1	AKD0		
8	SWSXR	-	-	-	SSX4	SSX3	SSX2	SSX1	SSX0	X 轴卷动起始点
9	SWSYR	-	-	SSY5	SSY4	SSY3	SSY2	SSY1	SSY0	Y 轴卷动起始点
A	SWRXR	-	-	-	SRX4	SRX3	SRX2	SRX1	SRX0	X 轴卷动范围
B	SWRYR	PINV	-	SRY5	SRY4	SRY3	SRY2	SRY1	SRY0	Y 轴卷动范围
C	SCOR	SL7	SL6	SL5/SR5	SL4/SR4	SL3/SR3	SL2/SR2	SL1/SR1	SL0/SR0	卷动位移量
D	ASCR	SPD3	SPD2	SPD1	SPD0	STP3	STP2	STP1	STP0	自动卷动控制

E	SCCR	SCR_IM D1	SCR_IM D0	SCR_M D	SBUF	SCR_DI R1	SCR_DI R0	SCR_IN TEN	AUTO_S CR	卷动控制
F	ISR	BF	-	-	-	IO_I	SCR_I	KI	BI	中断状态显示
10	CSTR	BR2	BR1	BR0	CT4	CT3	CT2	CT1	CT0	对比调整
11	DRCR_A	BOFF	EN_R	EN_G	ROFF	IDIR	-	CDIR	SDIR	驱动控制
12	DRCR_B	CK_BS1	CK_BS0	RR2	RR1	RR0	IRS	HD1	HD0	驱动控制
13	BLTR	BLK_EN	PBK_EN	-	INV	BLT3	BLT2	BLT1	BLT0	闪烁设定
14	IODR	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0	I/O 埠方向设定
15	IODAR	IOD7	IOD6	IOD5	IOD4	IOD3	IOD2	IOD1	IOD0	I/O 端口资料
16	ELCR	EL_EN	-	-	-	ELT3	ELT2	ELT1	ELT0	冷光控制
17	CGMI	-	-	-	-	-	UMI2	UMI1	UMI0	造字选择
18	CGMD	CGMD7	CGMD6	CGMD5	CGMD4	CGMD3	CGMD2	CGMD1	CGMD0	造字资料

### 6.2 缓存器内容描述

MPU 对模块有两种命令周期(Command Cycle)，一为缓存器读写周期(Register Cycle, RS = 0)，另一为内存读写周期(Memory Cycle, RS = 1)。在进行缓存器读写周期时，MPU 必须先告之模块要读的哪一个缓存器进行读或写，因此 MPU 会将第一笔数据(Data)传给模块，模块收到

后将此数据存入索引缓存器，同时认定下一笔数据就是要对索引缓存器内所指定的缓存器进行读或写。

#### IR (Index Register), 索引缓存器

RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	ID4	ID3	ID2	ID1	ID0

ID[4..0]: 此为模块的索引缓存器，固定在进行缓存器读写的第一个周期被用到，用来设定 MPU 在缓存器读写的第二周期是对模块的哪一个缓存器进行读写。

由索引缓存器可以看出它可以指定到 32 个缓存器(00H~1Fh)，但是模块只用到 22 个缓存器，这些缓存器在 RESET 后的起始数据(Initial Data)都是 00h。

#### Memory Data (RAMD), 内存

RW	RS	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0/1	1	D7	D6	D5	D4	D3	D2	D1	D0

当 RS = 1，表示 MPU 对模块进行内存读写周期(Memory Cycle)，如果写入(/RW = 0)数据到内存(Display RAM 或 ICON RAM)，写入的资料会根据 MD[1..0](REG[03h] bit1-0)的设定而有不同的定义(例如写入 GB 码或是 ASCII 码，或是 Bit-Map)。如果读取(/RW = 1) 内存数据，在不同的模式下也有不同的读法:

1. 全型字模式：以由左上到左下(16 列)，而后再从右上到右下(16 列)的方式读取字型，共 32Byte。
2. 半型字模式：由左上到左下(16 列)依序读取，共 16 Byte。
3. 小 ASCII 模式：由左上到左下(8 列)，共 8 Byte。
4. 图型模式：由屏幕左上到右上，一次 1 个 Byte(8 Pixel)，依序读取，而后换行。

#### [00h] Driver Waveform Register (DWFR), 驱动波形设定缓存器

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	B/C	--	NW5	NW4	NW3	NW2	NW1	NW0

B/C: 选择驱动输出的波形。0 ---B-Type 波型，1 --- C-Type 波型。

NW[5..0]: 用来设定内部 FRAME 信号在扫描到第几个 Segment(Row)时转态，只适用于 B/C = 1 (C-Type 波型)时。

#### [01h] Power Control Register (PWRR), 电源控制缓存器

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	SRST	MCLR	--	IO_IEN	KWK	IOWK	DOFF_Z	SLP

**SRST**: 软件重置(Reset)。1---全部缓存器将被设成起始状态，显示内存数据不变，此设定后必须经过 50usec 模块才可以接受新的命令，0---无作用。

**MCLR**: 清除内存。1---显示内存将被全部写入“00h”，此设定后必须经过 50usec 模块才可以接受新的命令，0---无作用。如果 MCLR 与 SRST 同时被设成“1”，模块芯片将先写入“00h”到显示内存，然后再进行软件重置。

**IO\_IEN**、**KWK**、**IOWK**: 不使用，全设为 0。

**DOFF\_Z**: 显示关闭(Display Off)。0---模块显示器(Panel)画面将被关闭，1---LCD 显示器(Panel)

画面将被开启。

**SLP**：睡眠模式设定。1---模块进入睡眠模式，此时时脉与显示器将被关闭。

0---模块 Wake Up。

**[02h] System Register (SYSR), 系统设定缓存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	LS3	LS2	LS1	LS0	GB_EN	--	RS1	RS0

**LS[3..0]**：设定 Panel 的显示行数(Segment)，模块的 Segment 为 144。LS3~LS0 设为 1000。

**GB\_EN**：设定 GB 码或 BIG5 码。1---GB 码，0 ---BIG5 码。

**RS[1..0]**：设定 Panel 的显示列数(Common)或 Duty，模块的 Common 为 64 (不包括 Icon 的 Common)。RS1、RS0 均设为 1。

**[03h] Memory Write Mode Register (MWMR), 内存输入模式缓存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	BMOD1	BMOD0	BIEN	ASCS	BOLD	INV	MD1	MD0

**BMOD[1..0]**：设定写入内存的范围。

表 6-2

BMOD1	BMOD0	写入记忆体范围
0	0	正常显示范围
0	1	显示范围 + 卷动 Buffer 区域
1	x	卷动 Buffer 区域

**BIEN**：忙碌中断控制。1---允许忙碌(内部写入内存之动作)后发生中断，0---不允许忙碌中断发生。

**ASCS**：选择 ASCII 表。0---选择 ASCII 表 1(Table1)，1---选择 ASCII 表 2(Table2)，请参考第 6-9 节。

**BOLD**：选择粗体字。0---选择正常字写入显示内存，1---选择以粗体字写入显示内存。

**INV**：选择反向字。0---选择正常字写入显示内存，1---选择以反向字写入显示内存。

**MD[1..0]**：选择内存输入模式。

表 6-3

MD1	MD0	输入模式
0	0	图型模式
0	1	小 ASCII 模式(8X8)
1	0	大 ASCII 模式(8X16)
1	1	全型字模式(16X16)

在全型字模式下(MD[1..0] = 11)，若写入第一个 Byte 小于 80h，则自动判断为 ASCII 码，显

示大 ASCII 字型，但是对大于 80h 的 ASCII 则必须在大 ASCII 模式(MD[1..0] = 10)才能显示。

**[04h] Cursor Control Register (CURCR), 光标控制寄存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	H3	H2	H1	H0	--	BLK	CR	CUR_EN

**H[3..0]**：光标高度设定。

表 6-4

H3	H2	H1	H0	高度 (Pixel)
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

在小 ASCII 模式(8X8)时，H3 不被使用，光标高度设定只能由 1~8 个像素(H[3..0] = x000b~x111b)。

**BLK**：光标闪烁选择。0---光标不闪烁，1---光标闪烁。

**CR**：光标归位设定。0---无动作，1---光标归位，光标会回到最左上方。

**CUR\_EN**：光标显示设定。0---光标不显示，1---光标显示

**[05h] Cursor Position Register of X (X-CUR), 光标 X 位置寄存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	--	--	X5	X4	X3	X2	X1	X0

X[5..0] : 以半角字宽度(8 Pixel)为单位, 设定光标于 Segment (X 轴)的位置。因为模块的 Segment 有 144Pixel, 因此 X[5..0] 的设定范围为 0~11h, 当 X[5..0] = 20h 及 21h 时光标是指到水平卷动 Buffer 的位置。

**[06h] Cursor Position Register of Y (Y-CUR), 光标 Y 位置寄存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	--	Y6	Y5	Y4	Y3	Y2	Y1	Y0

Y[6..0] : 以像素高度(Pixel)为单位, 设定光标于 Common (Y 轴)的位置。因为模块的 Common 有 64Pixel, 因此 Y[6..0] 的设定范围为 0~3Fh, 当 Y[6..0] = 40h~4Fh 时光标是指到垂直卷动 Buffer 的位置。如果 Y[6..0] = 50h, 光标将定在 COMS(Icon)的位置。

**[08h] Scroll Window Start X Register (SWSXR), X 轴卷动起始点寄存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	--	--	--	SSX4	SSX3	SSX2	SSX1	SSX0

**SSX[4..0] :** 设定卷动窗口的范围其 Segment (X 轴)的起始点, 以半型字(8 Pixel)为单位。

**[09h] Scroll Window Start Y Register (SWSYR), Y 轴卷动起始点寄存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	--	--	SSY5	SSY4	SSY3	SSY2	SSY1	SSY0

**SSY[5..0] :** 设定卷动窗口的范围其 Common (Y 轴)的起始点, 以 Pixel 为单位。

**[0Ah] Scroll Window Range X Register (SWRXR), X 轴卷动范围寄存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	--	--	--	SRX4	SRX3	SRX2	SRX1	SRX0

**SRX[4..0] :** 设定要卷动的窗口范围其 Segment (X 轴)的 Offset, 以半型字(8 Pixel)为单位。

**[0Bh] Scroll Window Range Y Register (SWRYR), Y 轴卷动范围寄存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	PINV	--	SRY5	SRY4	SRY3	SRY2	SRY1	SRY0

**PINV :** 屏幕显示反向区域设定。0 → 全屏幕显示反向, 1 → 区域(Partial)显示反向。

**SRY[5..0] :** 设定要卷动的窗口范围其 Common (Y 轴)的 Offset, 以 Pixel 为单位。

**[0Ch] Scroll Offset Register (SCOR), 卷动位移量寄存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	SL7	SL6	SL5/SR5	SL4/SR4	SL3/SR3	SL2/SR2	SL1/SR1	SL0/SR0

**SL[7..0] :** 水平方向卷动的位移量(以 Pixel 为单位), 当 SCR\_MD (REG[0Eh]bit 5) 为 0 时动作。

**SR[5..0] :** 垂直方向卷动的位移量(以 Pixel 为单位), 当 SCR\_MD (REG[0Eh]bit 5) 为 1 时动作。

当设定为自动卷动时, 此寄存器可以用来设定 Common 或 Segment 的卷动起始位置。

当设定为手动卷动时, 此寄存器的位移量设定不可以超过卷动设定范围, 也就是寄存器 08h~0Bh 的设定范围。

**[0Dh] Auto-Scroll Control Register (ASCR), 自动滚动控制寄存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	SPD3	SPD2	SPD1	SPD0	STP3	STP2	STP1	STP0

**SPD[3..0]** : 设定自动滚动的速度。

表 6-5

SPD3	SPD2	SPD1	SPD0	滚动时间
0	0	0	0	1 Unit
0	0	0	1	3 Units
0	0	1	0	5 Units
0	0	1	1	7 Units
0	1	0	0	17 Units
0	1	0	1	19 Units
0	1	1	0	21 Units
0	1	1	1	23 Units
1	0	0	0	129 Units
1	0	0	1	131 Units
1	0	1	0	133 Units
1	0	1	1	135 Units
1	1	0	0	145 Units
1	1	0	1	147 Units
1	1	1	0	149 Units
1	1	1	1	151 Units

1 Unit = 1 Frame Times

**STP[3..0]** : 设定自动滚动时, 每次的位移像素。

表 6-6

STP3	STP2	STP1	STP0	位移像素 (Pixel)
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

#### [0Eh] Scroll Control Register (SCCR), 卷动控制寄存器

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	SCR_IM D1	SCR_IM D0	SCR_MD	SBUF	SCR_DI R1	SCR_DI R0	SCR_INT EN	AUTO_S CR

**SCR\_IMD[1..0] :**

**SCR\_IMD[1..0] :**

0X: 自动卷动启动时, 设定每卷动 1 像素后发出中断(SCR\_INTEN 必须为 1)。

10: 自动卷动启动时, 设定每卷动 8 像素后发出中断(SCR\_INTEN 必须为 1)。

11: 自动卷动启动时, 设定每卷动 16 像素后发出中断(SCR\_INTEN 必须为 1)。

**SCR\_MD :**卷动模式设定。0---选择手动卷动模式, 并且将卷动的 Offset 值归 0, 1---选择自动卷动模式。

**SBUF :** 卷动 Buffer 设定。0---卷动 Buffer 禁能, 卷动时不包含卷动 Buffer, 仅包含屏幕显示所及的部分。1---卷动 Buffer 致能, 卷动时包含卷动 Buffer。

**SCR\_DIR[1..0] :** 卷动方向设定。

表 6-7

表 5-12

SCR_DIR1	SCR_DIR0	卷动方向
0	0	由左至右(水准)
0	1	由右至左(水准)
1	0	由上至下(垂直)
1	1	由下至上(垂直)

**SCR\_INTEN:** 卷动中断设定。0 → 不发动卷动中断，1 → 自动卷动时，当卷动 1、8 或 16 个像素后，会发出中断通知 MPU(非自动卷动时，此功能无效)。

**AUTO\_SCR :** 自动卷动模式设定。0 → 停止自动卷动模式。若要跳出自动卷动模式，或是进行新的显示资料写入，则 Bit5 的 SCR\_MD 必须先清除为 0，以免影响之后进行的画面显示。1 → 进行自动卷动模式。

#### [0Fh] Interrupt Status Register (ISR), 中断状态缓存器

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	BF	--	--	--	IO_I	SCR_I	KI	BI

**BF :** 忙碌旗标，1---显示内存处于忙碌状态(Data Write)，0---显示内存处于闲置状态(Write 完成)。

**IO\_I :** I/O 口中断，不使用，不必理会。

**SCR\_I :** 卷动中断，1---卷动完成后中断发生，0---无卷动中断发生。

**KI :** 键盘扫描中断，不使用，不必理会。

**BI :** 忙碌中断，1---内部写入内存之动作全部完成后中断发生，0---无忙碌中断发生。

#### [10h] Contrast Adjust Register (CSTR), 对比调整缓存器

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	BR2	BR1	BR0	CT4	CT3	CT2	CT1	CT0

**BR[2..0] :** 设定 LCD Bias(以 144x65 做计算标准)。

**BR2~BR0:** 全部设置为 1，1/9Bias。

**CT[4..0] :** 设定显示对比度(Contrast)，共 32 阶，为避免显示品质不佳或显示画面较淡，通常依据液晶玻璃材质、电源电压及尺寸大小设定 CT[4:0]，调整出最佳的显示品质。

表 6-8

BR2	BR1	BR0	Bias
0	0	0	1/5
0	0	1	1/6
0	1	0	1/7
0	1	1	1/8
1	0	0	1/9
1	0	1	1/9
1	1	0	1/9
1	1	1	1/9

**[11h] Driver Control Register1 (DRCR\_A), 驱动控制寄存器 A**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	BOFF	EN_R	EN_G	ROFF	IDIR	-	CDIR	SDIR

**BOFF**、**EN\_R**、**EN\_G**、**ROFF** 为内部升压等设定，全部设为 1。

**IDIR** : Icon 顺序排列选择, 0 → Icon 顺序排列固定不变, 1 → Icon 顺序排列随 SDIR 而变。

**CDIR** : Common 顺序排列选择。0 → 接脚 COM0~63 代表 Common 0 ~ 63, 1 → 接脚 COM0~63 代表 Common 63~0。

**SDIR** : Segment 顺序排列选择。0 → 接脚 SEG0~143 代表 Segment 0 ~ 143, 1 → 接脚 SEG0~143 代表 Segment 143~0。

**[12h] Driver Control Register (DRCR\_B), 驱动控制寄存器 B**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	CK_BS1	CK_BS0	RR2	RR1	RR0	IRS	HD1	HD0

**CK\_BS[1..0]** : 升压电路(Booster)时脉设定。假设系统 RC 振荡时脉 = 100KHz。

表 6-9

CK_BS1	CK_BS0	升压电路(Booster)时脉
0	0	SYS_CLK/2 → 50KHz
0	1	SYS_CLK/4 → 25KHz
1	0	SYS_CLK/6 → 16.7KHz
1	1	SYS_CLK/8 → 12.5KHz

**RR[2..0]** : 电压调整器(Regulator) 阻值比, 也就是参考电压 VREF 与 V0 的比值。调整会影响对比度。

表 6-10

RR2	RR1	RR0	电压调整器 阻值比 (Resistor Ratio)
0	0	0	X3
0	0	1	X3.5
0	1	0	X4
0	1	1	X4.5
1	0	0	X5
1	0	1	X5.5
1	1	0	X6
1	1	1	X6.4

**IRS:** 选择电压调整器使用的分压电阻。设为 1，使用内部分压电阻。

**HD[1:0]:** LCD 驱动器的驱动电流设定，通常液晶玻璃尺寸大时需设定较大的驱动电流，避免显示品质不佳或显示画面较淡。

表 6-11

HD1	HD0	驱动电流
0	0	小(Min) ↓ 大(Max)
0	1	
1	0	
1	1	

[13h] Blink Timer Register (BLTR), 闪烁设定缓存器

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	BLK_EN	PBK_EN	--	INV	BLT3	BLT2	BLT1	BLT0

**[13h] Blink Timer Register (BLTR), 闪烁设定缓存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	BLK_EN	PBK_EN	--	INV	BLT3	BLT2	BLT1	BLT0

**BLK\_EN** : 闪烁功能设定。0 → 闪烁功能关闭, 1 → 闪烁功能开启。

**PBK\_EN** : 全屏幕或区域闪烁设定。0 → 选择全屏幕闪烁, 1 → 选择区域闪烁, 闪烁的区域是根据滚动窗口 (Scroll Window) 来决定, 也就是由缓存器 SWSXR、SWSYR、SWRXR 及 SWRYR 的设定来决定, 当区域闪烁关闭时请将上面滚动窗口的缓存器清除为 0, 以免影响之后进行的画面显示。注意, 只有当 BLK\_EN= 1 时, 才会开启闪烁功能。

**INV** : 屏幕显示反向设定。0 → 屏幕显示正常, 1 → 屏幕显示反向, 反向区域由缓存器 SWRYR(REG[0Bh]) 的 Bit7(PINV) 决定。

**BLT[3..0]** : 闪烁时间设定, 以 Frame 为单位。

表 6-12

BLT3	BLT2	BLT1	BLT0	闪烁时间 (Frames)
0	0	0	0	8
0	0	0	1	16
0	0	1	0	24
0	0	1	1	32
0	1	0	0	40
0	1	0	1	48
0	1	1	0	56
0	1	1	1	64
1	0	0	0	72
1	0	0	1	80
1	0	1	0	88
1	0	1	1	96
1	1	0	0	104
1	1	0	1	112
1	1	1	0	120
1	1	1	1	128

**[14h] I/O Direction Control Register (IODR), I/O 端口方向设定缓存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0

**OE[7..0]** : 设定 I/O 埠的输出或输入。0 → 相对应之 I/O 埠为输入(Input), 1 → 相对应之 I/O 埠为输出(Output)。

当有背光亮灭控制功能时，OE0 为 1 时，背光开，为 0 时，背光关。  
如无背光亮灭控制功能，不必理会。

**[17h] CGRAM Register (CGMI), 造字选择寄存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	-	-	-	-	--	UMI2	UMI1	UMI0

**UMI[2..0]**：选择自行造字时所要造的字，模块允许使用者造 8 个 16x16 的全型字，其对映的字型码为 FFF0h~FFF7h。

表 6-13

UMI2	UMI1	UMI0	对映的字型码
0	0	0	FFF0h
0	0	1	FFF1h
0	1	0	FFF2h
0	1	1	FFF3h
1	0	0	FFF4h
1	0	1	FFF5h
1	1	0	FFF6h
1	1	1	FFF7h

**[18h] CGRAM Data Register (CGMD), 造字数据寄存器**

RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	CGMD7	CGMD6	CGMD5	CGMD4	CGMD3	CGMD2	CGMD1	CGMD0

**CGMD[7..0]**：此寄存器用来传递或读取 16x16 的全型字数据。在设定完寄存器[17h]后，使用者将所要造的 16x16 全型字 Bit Map 以连续 32Byte 写入此寄存器，日后要显示此全型字只要透过 MPU 写入两个 Byte 的对映字型码即可。

**6-3 寄存器初始化程序：**

```
void lcd_initial(void)
```

```
{ //初始化,
```

```
/*对寄存器的写操作是先写寄存器的地址，后写八位寄存器数据，如下面函数的实参。*/
```

```
lcd_regwrite(0x00,0x00); // Driver Waveform Register (DWFR)
```

```
lcd_regwrite(0x01,0x02); // Power Control Register (PWRR) 开显示
```

```
lcd_regwrite(0x02,0x8f); // System Register (SYSR) 设定 GB 码，设定显示范围
```

```
lcd_regwrite(0x03,0x00); // Memory Write Mode Register (MWMR)
```

```
lcd_regwrite(0x04,0x00); // Cursor Control Register (CURCR) 光标不显示
```

```
lcd_regwrite(0x0e,0x00); // Scroll Control Register (SCCR), 滚动控制寄存器
```

```
lcd_regwrite(0x0f,0x00); // Interrupt Status Register (ISR), 中断状态寄存器
```

```
lcd_regwrite(0x10,0xef); // Contrast Adjust Register (CSTR) 对比度设定
```

```

lcd_regwrite(0x11,0xf0); // Driver Control Register1 (DRCCR_A) 设定 SEG 和 COM 的方向
lcd_regwrite(0x12,0x3f); //Driver Control Register (DRCCR_B)
lcd_regwrite(0x13,0x00); // Blink Timer Register (BLTR) 闪烁功能关闭
lcd_regwrite(0x14,0x01); // I/O Direction Control Register (IODR) 开背光
}

```

## 7. 微控制器接口

### 7-1 并行接口

模块的 MPU 接口可支持 Intel 8080 系列与 Motorola 6800 系列的 4 位或 8 位微处理器, 使用者可以透过板上 **R17**, **R18** 去选择模块的 MPU 界面是 8080 或者是 6800 的兼容系统, 如果贴上 R17 (4.7k), 去掉 R18, 则模块的 MPU 接口将定义成与 8080 兼容的接口。反之, 去掉 R17, 贴上 R18 (4.7k), 则将定义成与 6800 兼容的接口。而 BIT4 脚位可用来选择所连接的 MPU 为 4/8 位, 如果 BIT4 接到 GND, 则表示 MPU 选用 8 位接口, BIT4 接到 VDD, 则表示 MPU 选用 4 位接口, 在 4 位接口时, 数据总线(Data Bus) 只有用到 DB[3..0]。

### 7-2 串行接口

模块除了支持上述的并行接口外, 也可以支持一种 3 线串行接口、两种 4 线(A-Type, B-Type)串行接口或 IIC 串行接口, 使用者可以透过 P/S 这根脚位去选择模块的 MPU 接口是并行或者是串行接口, 当脚位 P/S = 1 时, 此时为平行并行接口, DB[7..0] 为地址/数据传输线, 而当脚位 P/S = 0 时, MPU 与 模块之接口为串行模式(Serial Mode), 此时数据总线的 DB[7..6]被当成 SMOD[1..0] 的输入设定脚位用, 来决定哪一种串行模式设定, 如表 5-1 所示。在串行接口时许多数据传输线(Data Bus)被用来当作串行控制信号, 请参考前面第 3 节的说明。而图 6-3 ~ 6-6 是 MPU 与 RA8816 之间的串行接口接口图。

表 7-1

SMOD	串行界面模式
0 0	IIC 传输(2-Wire), 使用到 SCK, SDA, IICA[5..0]
0 1	三线式传输(3-Wire), 使用到 SCK, SDA, $\overline{CS}$
1 0	四线式传输(4-Wire, A-Type), 使用到 SCK, SDA, RS, $\overline{CS}$
1 1	四线式传输(4-Wire, B-Type), 使用到 SCK, SDO, SDI, $\overline{CS}$

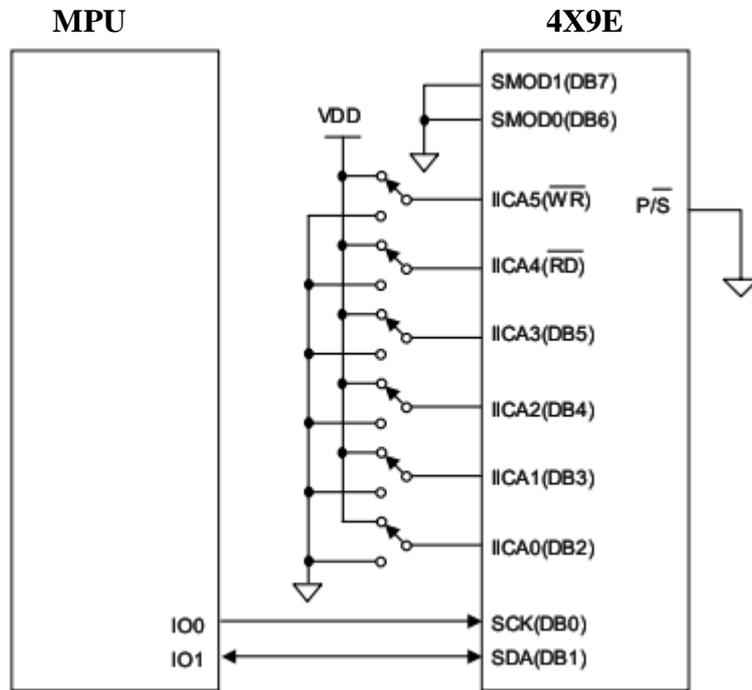


图 7-2: IIC 的 MPU 界面图

在 IIC 串行模式下, WR、RD 与 DB[5..2]组成的 IICA[5..0]作为 IIC Bus 的组件地址选择信号, 只要设定与系统上的其它 IIC 组件不发生冲突即可, 虽然 IIC 可以指定 128 个组件, 但是模块只提供 64 个地址选择(IICA[5..0]), 同时模块将 IICA[6]由内部固定为 0, 因此组件地址选择只能由 00~3Fh。另外使用 4 线式(B-Type)的 MPU 接口在 SCK 上要加一 Pull-Hi 电阻, 如图 5-4。

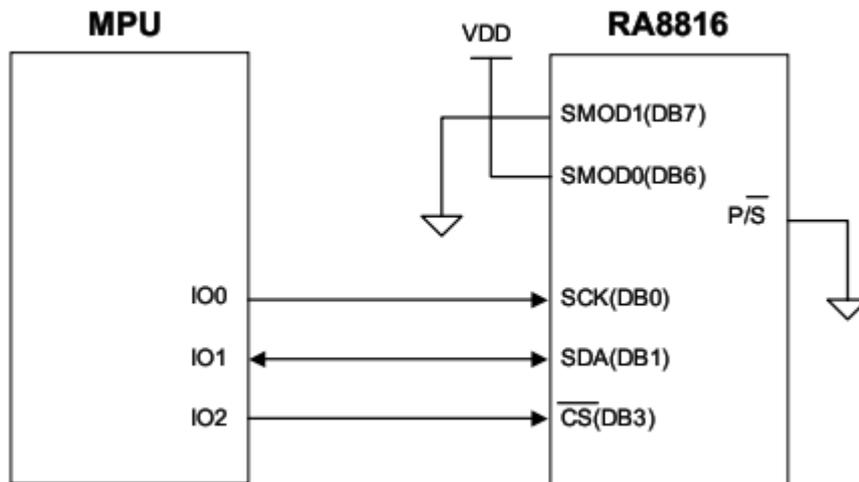


图 7-3: 3 线式的 MPU 界面图

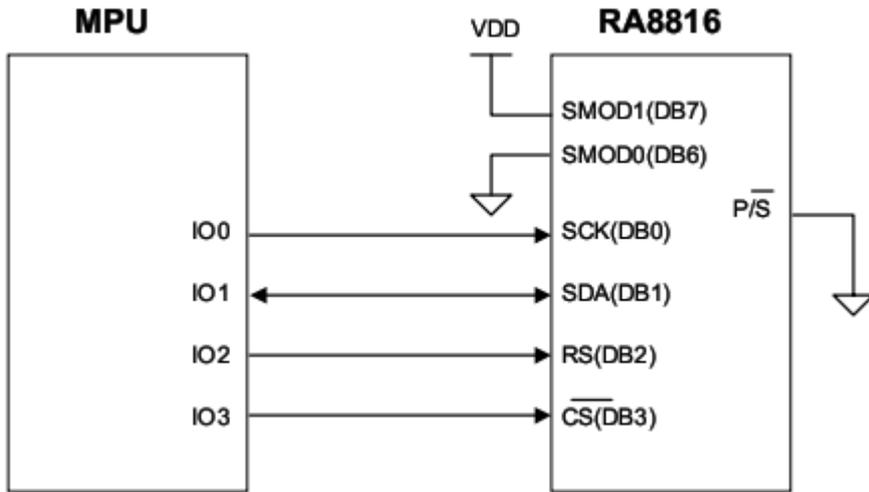


图 7-4: 4 线式(A-Type)的 MPU 界面图

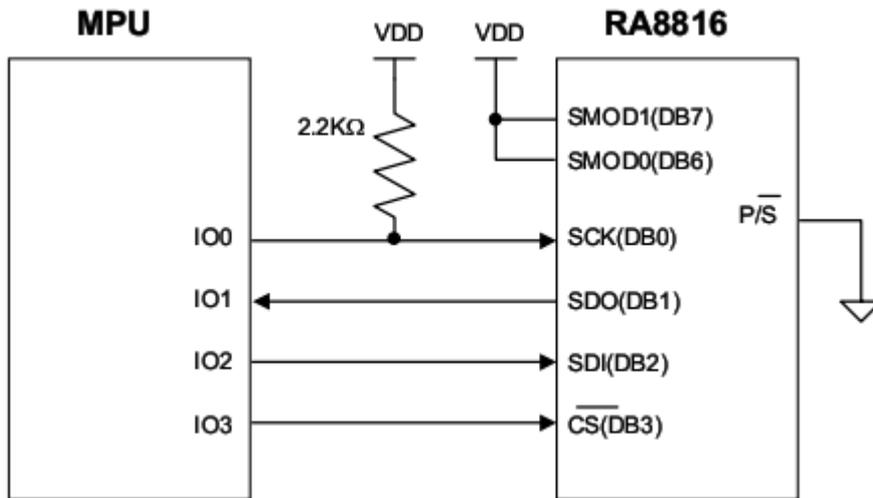


图 7-5: 4 线式(B-Type)的 MPU 界面图

### 7-3 缓存器读写

MPU 对模块只有两种控制时序，一为 MPU 对缓存器的读写，另一为 MPU 对内存的读写。如前面第四节所述，在进行缓存器读写周期时，MPU 必须先告之 模块要对的哪一个缓存器进行读或写，因此 MPU 传给 RA8816 的第一笔数据(Data)代表要对模块的哪一个缓存器进行读或写，第二笔数据才是真正代表写入该缓存器的数据，或是代表由缓存器读出的数据。由于缓存器内容控制着所有 RA8816 的动作，因此对缓存器的读写就如同对模块下命令(Command)一样，所以也可以说缓存器的读写周期就是命令周期(Command Cycle)。

图 6-7 与 6-8 为以 8080 MPU(8-Bit) 对 RA8816 缓存器进行读写的时序图，图 6-9 与 6-10 为以 6800 MPU(8-Bit) 对 RA8816 缓存器进行读写的时序图，图 6-11 到 6-14 为以串行模式对模块缓存器进行读写的时序图。

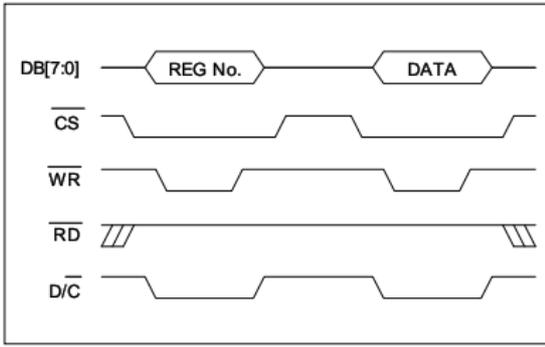


图 6-7：以 8080 8-Bit 对缓存器写入的时序图

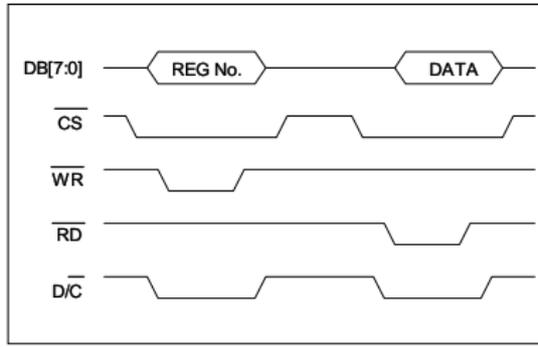


图 6-8：以 8080 8-Bit 对缓存器读取的时序图

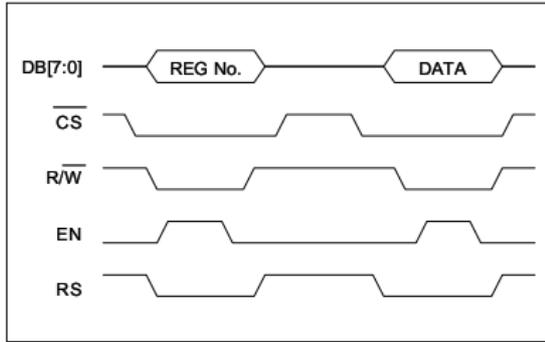


图 6-9：以 6800 8-Bit 对缓存器写入的时序图

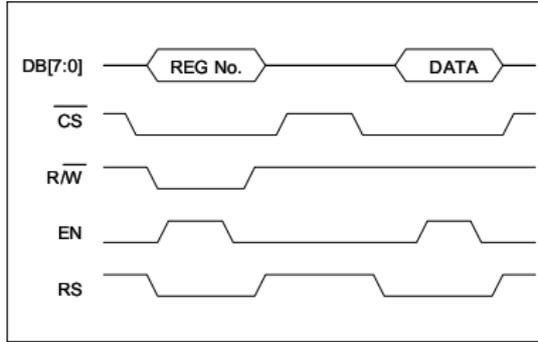


图 6-10：以 6800 8-Bit 对缓存器读取的时序图

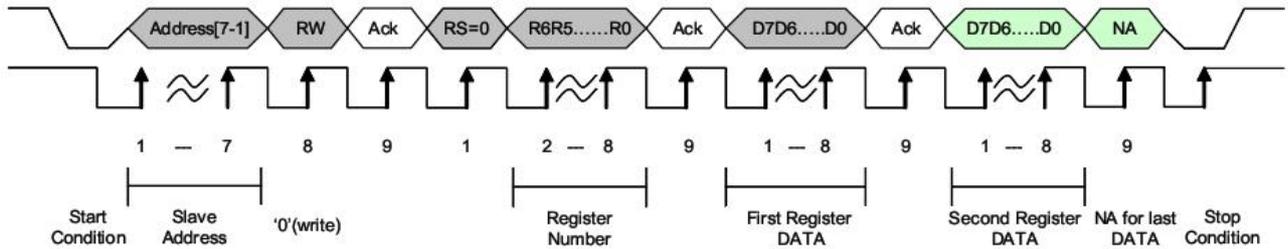


图 6-11a：I2C 串行接口对缓存器写的时序图

- From Master to Slave
- From Slave to Master
- Optional

Ack : Acknowledge(SDA LOW)  
 NA : Not Acknowledge(SDA HIGH)  
 RW : Master Read(1) / Write(0) Command  
 RS : Select Register(0) / Memory(1)

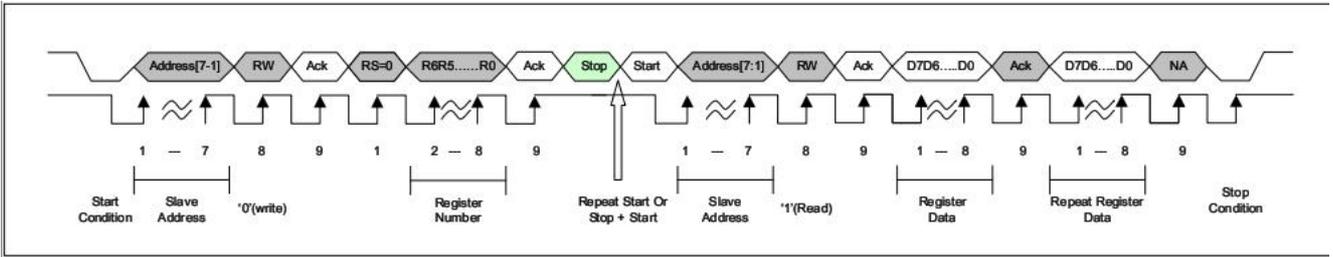


图 6-11b: IIC 串行接口对缓存器读的时序图

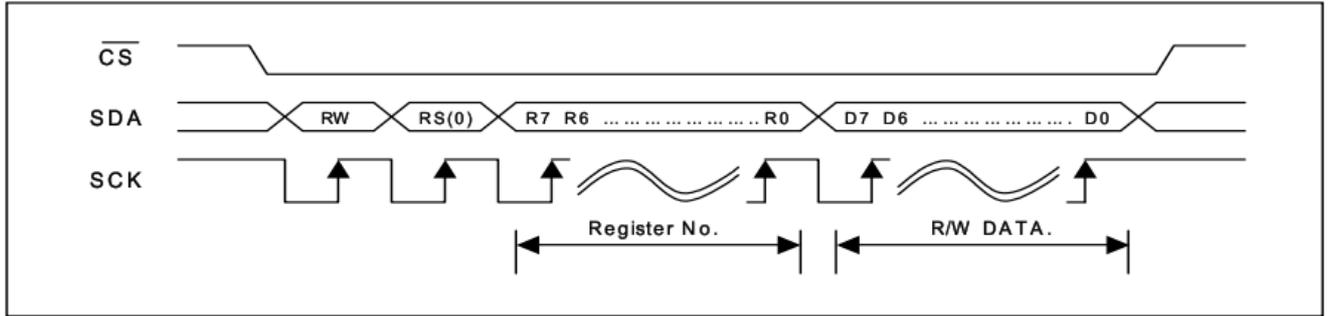


图 6-12: 3 线串行接口对缓存器读写的时序图

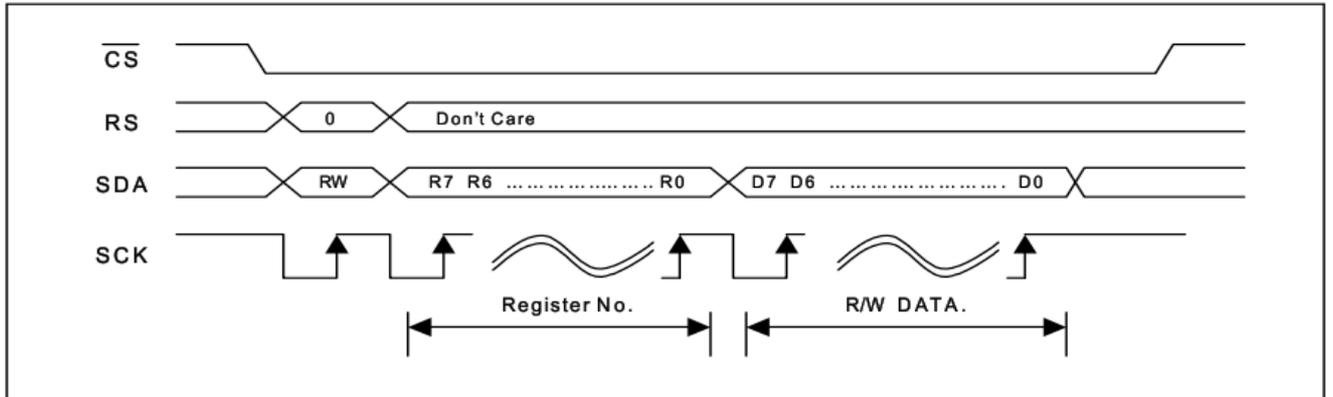


图 6-13: 4 线串行接口(A-Type)对缓存器读写的时序图

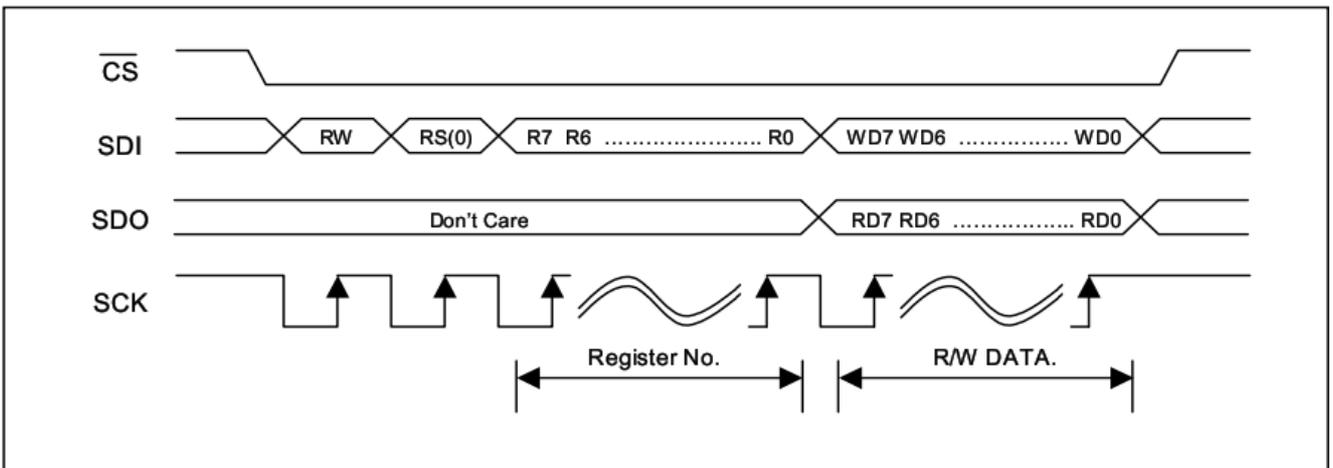


图 6-14: 4 线串行接口(B-Type)对缓存器读写的时序图

### 7-4 内存读写

MPU 对模块的另一时序为对内存的读写(Memory Write), 通常代表对显示内存做数据的传递,

一个 Memory Write 代表写入 Data 到目前光标所对映的显示内存地址，之后光标地址会自动加一，再一个 Memory Write 动作，Data 将被写入到新光标所对映的显示内存地址。由于都是资料的读写的动作，因此对内存的读写周期也可以解释为数据周期(Data Cycle)。

图 6-15 与 6-16 为以 8080 MPU(8-Bit) 对模块内存进行读写的时序图，图 6-17 与 6-18 为以 6800 MPU(8-Bit) 对模块内存进行读写的时序图，图 6-19 到 6-22 为以串行模式对模块内存进行读写的时序图。

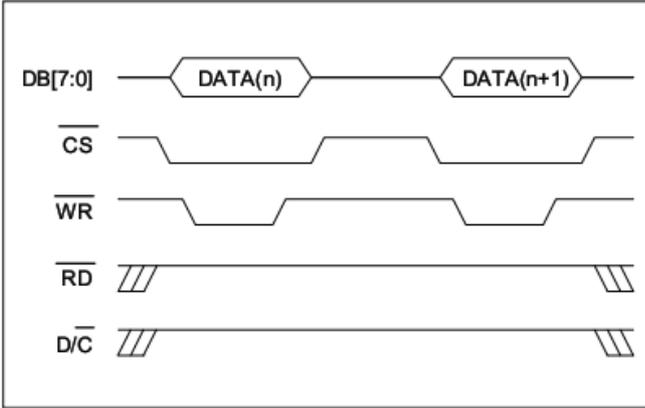


图 6-15: 以 8080 8-Bit 对内存写入的时序图

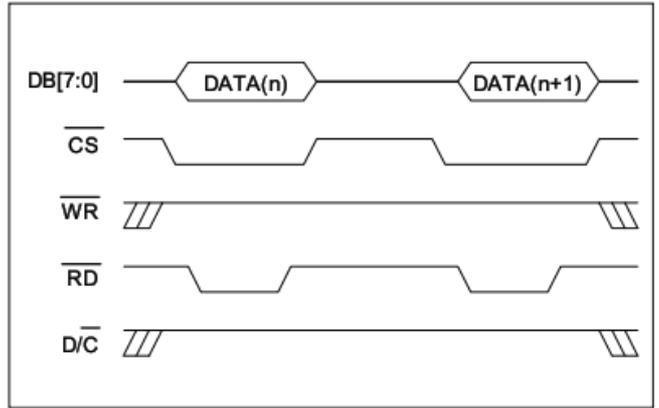


图 6-16: 以 8080 8-Bit 对内存读取的时序图

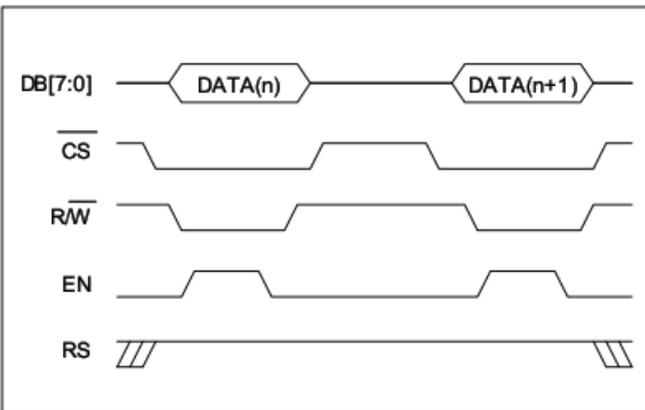


图 6-17: 以 6800 8-Bit 对内存写入的时序图

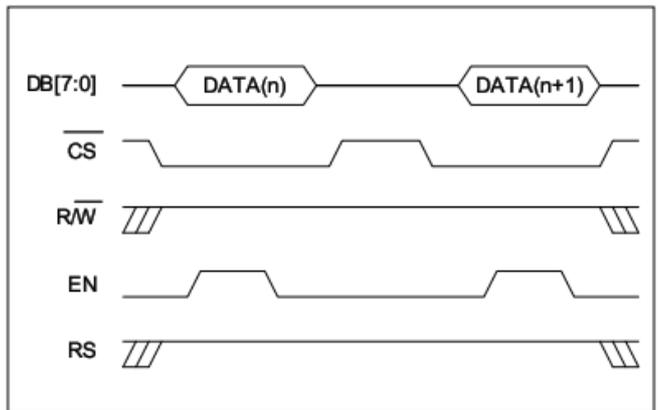


图 6-18: 以 6800 8-Bit 对内存读取的时序图

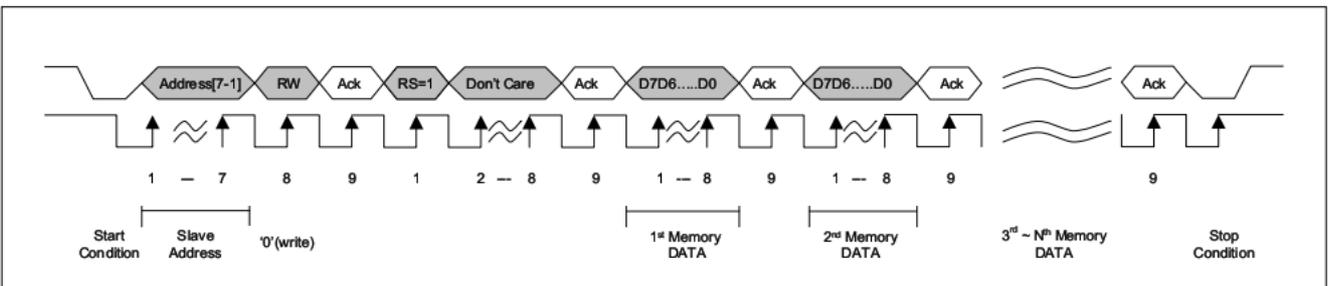


图 6-19a: IIC 串行接口对内存写的时序图

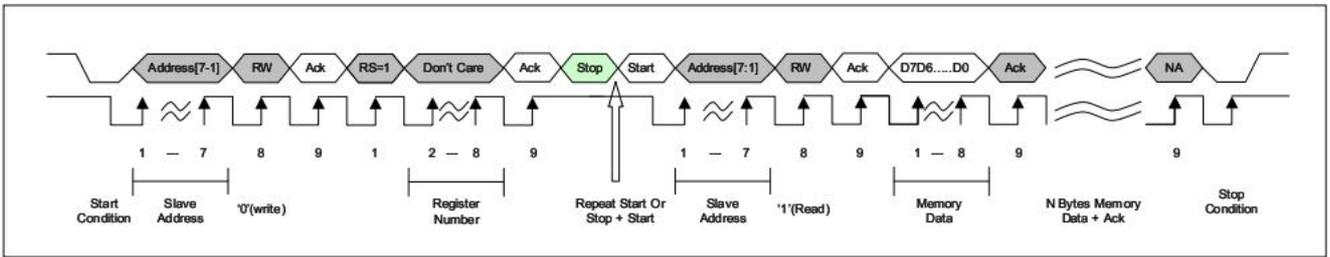


图 6-19b: IIC 串行接口对内存读的时序图

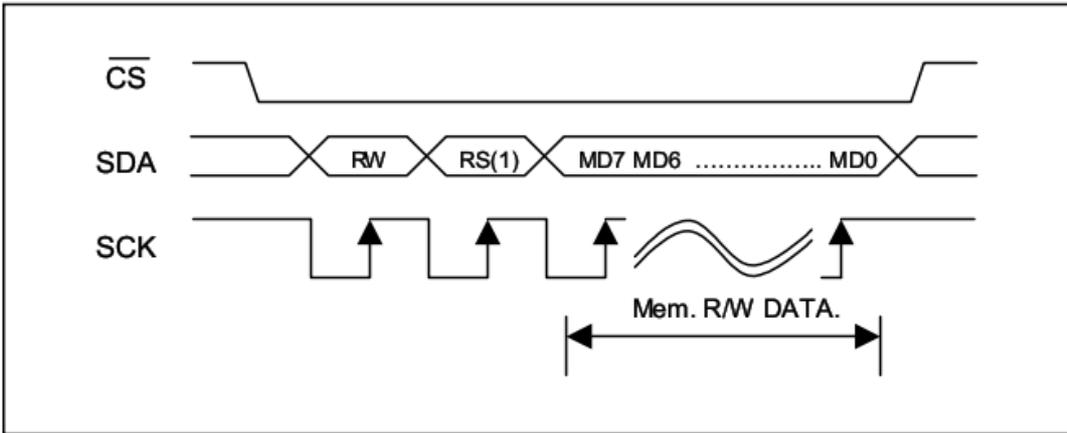


图 6-20: 3 线串行接口对内存读写的时序图

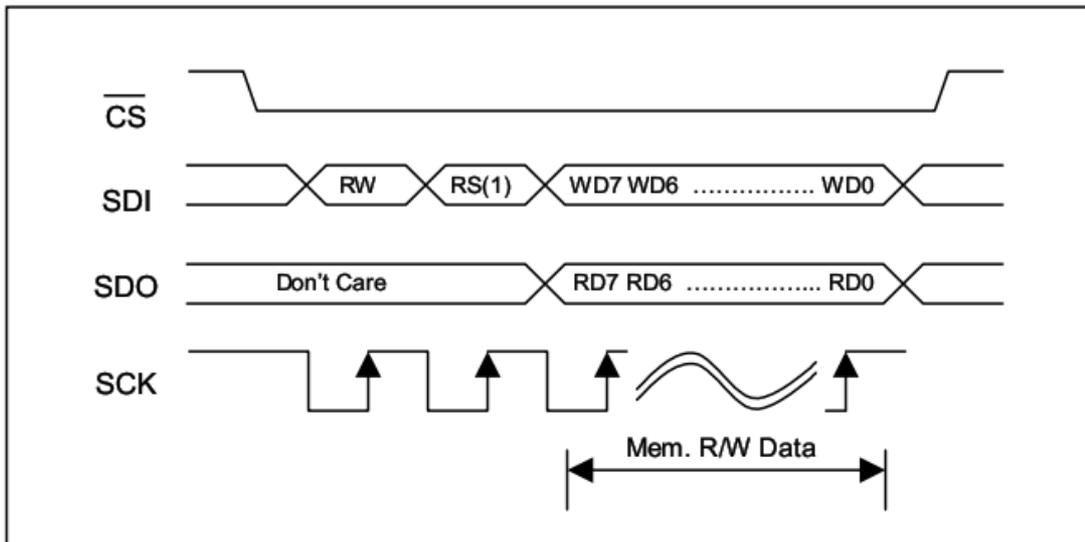


图 6-22: 4 线串行接口(B-Type 对内存读写的时序图

### 7-5 内存

模块内含三个内存，包括用来储存字型(字库)的 256Kbyte Font ROM、1170Byte 的显示内存(Display RAM)，以及供使用者自行造自的字型产生内存(CGRAM)。Font ROM 储存了中文繁体或简体字型数据，也包括英文、日文、欧文--拉丁语系(Latin, Latin-ext A, Latin-ext B)及 ASCII 表。在文字模式下，当模块收到 MPU 传送来的标准码，就会将其所对应的字型数据由 Font ROM 送到显示内存，显示内存的数据也会一直不断的被 LCD 控制电路读出，同时送到

LCD 驱动器，让讯息显示在 LCD Panel 上。因此 MPU 不用花费计算的时间以绘图模式描绘中文，可以节省 MPU 许多时间，提升液晶显示中文之处理效率，同时缩短软件开发时间。模块内建的中文字库有两种，-S 代表简体字库，-T 代表繁体字库，字库与字码的对映表请参考应用手册。

模块的 LCD 显示范围为 144x64 点，因此需要 $(144 * 64) / 8 = 1152\text{Byte}$  的显示内存 (Display RAM)，同时为了达到显示画面移动的效果，模块的显示内存还增加了滚动 Buffer，让显示画面在左右移动或上下移动时不会有迟钝的感觉。

字型产生内存(CGRAM)则是由使用者写入特殊的字型数据，让 MPU 以后可以用固定的码 (FFF0h~FFF7h)通知模块，将使用者自己建立的字型显示在 LCD Panel 上。

## 7-6 系统时脉

模块的系统时脉是由内部 RC 电路产生，此系统时脉约为 55KHz。当电源控制缓存器 (PWRR)的 Bit0 - SLP 被设定为 1 时，RA8816 将进入睡眠模式，此时系统时脉将被关闭。

## 7-7 中断

模块提供一中断信号线 (INT) 用来表示有三种中断讯息可能发生：

当内部写入内存之动作全部完成后，允许中断发生。

滚动中断，当滚动 1、8 或 16 个像素完成后，会发出中断。

这两种中断都可以单独被致能或禁能，而中断的设定与中断讯息可有由缓存器来控制与读取。中断信号线平时为 High，设定中断发生时为 Low，MPU 侦测到此中断后必须清除中断状态缓存器，中断信号线才会回到 High。如果不使用硬件中断信号线，MPU 可以用询问的方式读取中断状态缓存器，以得到中断讯息。

## 7-8 ASCII 区块选择设定

模块内建三个 ASCII 区块，包含许多文字、特殊符号或图形可供使用者直接取用，事实上这三个 ASCII 区块的数据是储存在 256Kbyte 的 Font ROM 内，如图 6-33~6-35，每个 ASCII 表的左边数值为 High Nibble，上边数值为 Low Nibble。此三个 ASCII 区块的选用方式由内存输入模式缓存器(MWMMR)的 MD1 与 MD0 来设定。如果使用者需要特殊符号或图形，亦可经由调整 ROM Code 来建立。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	☉	☼	☽	☾	☿	♁	♂	♃	♄	♅	♆	♇	♈	♉	♊	♋	♌
1	♍	♎	♏	♐	♑	♒	♓	♈	♉	♊	♋	♌	♍	♎	♏	♐	♑
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~		
8	☺	ü	é	â	ä	à	á	☺	è	ë	è	ì	î	ï	ä	å	
9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ö	ü	¢	£	¥	℞	ƒ	
A	á	í	ó	ú	ñ	ñ	á	â	ä	ä	é	ë	é	é	í	î	
B	ï	ï	ó	ö	ò	ó	ö	ú	ù	ù	œ	œ	œ	œ	œ	œ	
C	㊦	㊧	㊨	㊩	㊪	㊫	㊬	㊭	㊮	㊯	㊰	㊱	㊲	㊳	㊴	㊵	
D	㊶	㊷	㊸	㊹	㊺	㊻	㊼	㊽	㊾	㊿	㊿	㊿	㊿	㊿	㊿	㊿	
E	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	
F	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	㊿	

图 6-33: 小 ASCII 表(Table 0)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		⊗	⊙	⌈	⌋	⌈	⌋	▪	▣	=	-	-	ƒ	⊙		
1	▶	◀	⏪	⏩	⏪	⏩	⏪	⏩	⏪	⏩	⏪	⏩	⏪	⏩	⏪	⏩
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8	đ	À	á	ĸ	Ĝ	..	Ĉ	Ŝ	Ū		Š	ı	Ž	Ě	ž	ž
9	Ń	ñ	Č	†	ğ	"	ĉ	ŝ	ű	č	ş	ı	Ť	ę	ț	ž
A	ã	ı	Œ	£	€	¥	Š	Œ	Š	©	®	«	¬	-	®	-
B	°	±	²	³	ž	µ	¶	·	ž	'	°	»	œ	œ	ÿ	ı
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
F	ø	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

图 6-34: 大 ASCII 表(Table 1)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	☒	☑	⌈	⌋	⌈	⌋	⌈	⌋	▪	▣	=	-	-	▣	☑	
1	▶	◀	▮	▮	▮	▮	▮	▮	▮	▮	▮	▮	▮	▮	▮	▮
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8	⌘	,	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
9	◀	☒	☑	☑	☑	=	≡	±	+	-	×	÷	≈	π	Σ	∞
A		Ё	Ъ	Г	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ		Ў	Ц
B	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
C	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
D	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
E	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F	№	ё	ъ	ѓ	є	ѕ	і	ї	ј	љ	њ	ќ	ќ	ѕ	ў	ц

图 6-35: 大 ASCII 表(Table 2)

图 6-33 是小 ASCII 的 Table, 每个 ASCII 由 8x8 的点组成, 在小 ASCII 的显示模式下, 模块 LCD 可以秀出 8 列 ASCII 文字, 每列有 18 个字。图 6-34、6-35 是大 ASCII 的 Table, 在大 ASCII 的的显示模式下, 则可以秀出 4 列 ASCII 文字, 每列有 18 个字。

### 7-9 电源控制(Power Control)

模块提供两种电源模式(Power Mode), 一种为工作时的正常模式(Normal Mode), 一种为睡眠模式(Sleep Mode), 在正常模式时, 可写入"0"到电源控制缓存器(PWRR)的 Bit 0, 让 RA8816 进入睡眠模式, 睡眠模式启动时, 会关闭 LCD 显示功能, 将 RC 振荡器关闭, 因此整个 IC 仅耗损静态电流。

**模块唤醒(Wake-UP):** 透过 MPU 写入"1"到电源控制缓存器(PWRR)的 Bit 0, 让模块进入

正常模式。当唤醒模块时，模块 IC 先将 RC 振荡器起振，因此必须要等待一段起振时间(约 250ms)，之后 MPU 才能正常的进行数据存取，LCD Driver 也才能动作。

## 8. 显示功能

### 8-1 文字模式

模块内建 256KB 字型 ROM，包括一级常用字库--中文繁体或简体、英文、日文、ASCII、欧文--拉丁语系(Latin, Latin-ext A, Latin-ext B)，在文字模式下可以支持全角(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 或是 8x8 的点矩阵组成，如图 7-1 所示：

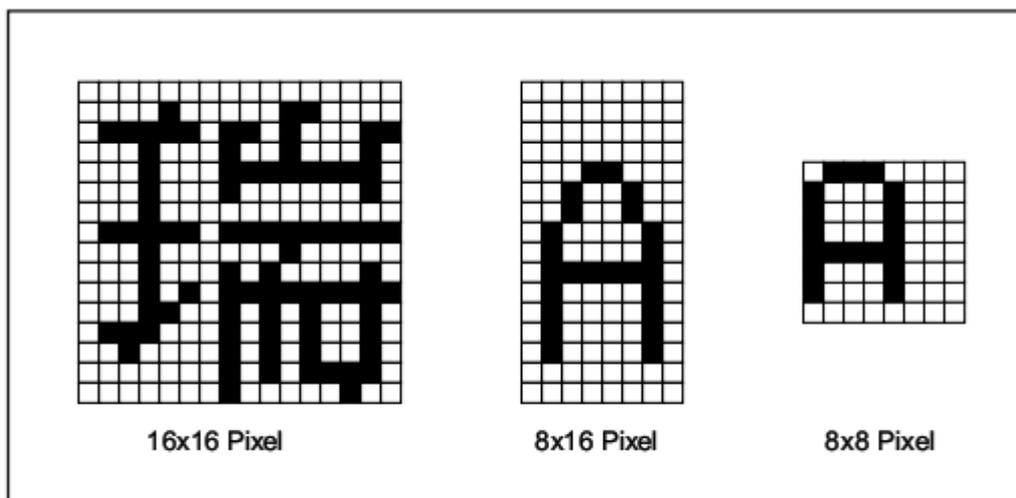


图 7-1：全角与半角文字

模块的中文显示方式是在文字模式操作，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。因为中文字码属于全型字模式，占两个 Byte，所以如果 MPU 接口是 8-Bit，则 MPU 必须分两次将中文字码的 High Byte & Low Byte 写入模块，而英文或数字码只占一个 Byte，因此只要将内码一次写入 模块既可。模块显示范围为 144 点 x 64 点，若以显示文字为例，全角字型（16x16）即是 9 行 x 4 列，半角字型（8x16）则可以显示到 18 行 x 4 列，半角字型（8x8）则可以显示到 18 行 x 8 列。

使用者可以透过内存输入模式缓存器(MWMMR)，设定写入显示内存的输入模式是图型模式、小 ASCII 模式(8x8)、大 ASCII 模式(8x16)、全型字模式(16X16)，在文字模式下也可以选择显示粗体字、反向字或正常字。

### 8-2 绘图模式

模块的绘图模式是以字符映像(bit map)方式填入图形数据在显示内存(Display RAM)上，将内存输入模式缓存器(MWMMR)的[MD1, MD0] 设成"00" 选择图型模式，之后进行内存的写入动作，数据将会被写入到目前光标所对映的显示内存地址。

模块显示像素(Pixel)范围为 144 点 x 64 点，因此需要约 1170 Byte 的 Display Data RAM (DDRAM) 来储存欲显示的每个像素点，在 DDRAM 里，只有在显示范围内的对应资料会被显示于 LCD 面板上，不在显示范围内的则会被忽略掉。当模块在显示图形的时候，是以字符映像(Bit Map)的方式写入 DDRAM，若 DDRAM 的某个位置被填满为 '1' 时，相对于 LCD 面板的位置会被显示出亮点，由图 7-3 可看出，在 DDRAM 上所储存之像素资料，会对应到显示屏幕(LCD)

上，而构成文字、符号或图形之显示效果。

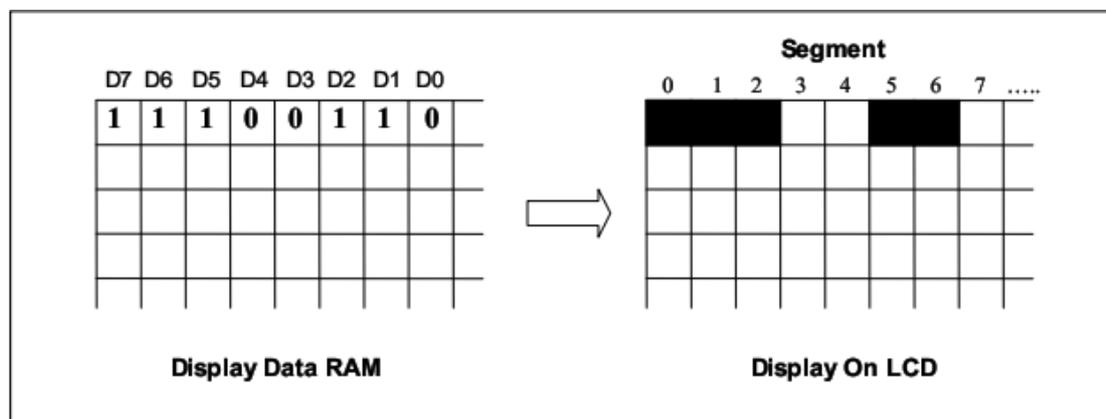


图 7-3: 显示资料到 LCD 显示的映像

模块提供自动清除 DDRAM 的功能，只要将电源控制缓存器(PWRR)的 MCLR 设成 1，模块将于极短的时间内将数据“00”全部写入 DDRAM 内。此外，在绘图模式下，模块透过闪烁设定缓存器(BLTR)，也可以设定闪烁功能及屏幕显示反向功能，闪烁的范围可以根据滚动窗口(Scroll Window)的大小来决定。

## 8-3 光标设定

### 8-3-1 光标位置与移位

模块支持 144(Segment) x 65(Common)的 Panel Size，但是光标的 Segment 地址是以每 8-Bit 为单位，例如想在 Panel 的左上角第三个全角位置秀出“制”，则必须设定光标位置缓存器 X-CUR = 04h，Y-CUR = 00h，同理，想在 Panel 的左上角第二行第一个全角位置秀出“器”，则必须设定光标缓存器 X-CUR= 00h，Y-CUR = 10h，请参考图 7-4。

模块不论文字或是绘图模式，都是使用缓存器 X-CUR 与 Y-CUR 来设定光标的地址，一旦光标地址设定，则不论在文字或是绘图模式，写入 Data 光标都会自动移位。而光标的移位是以显示窗口为边界。

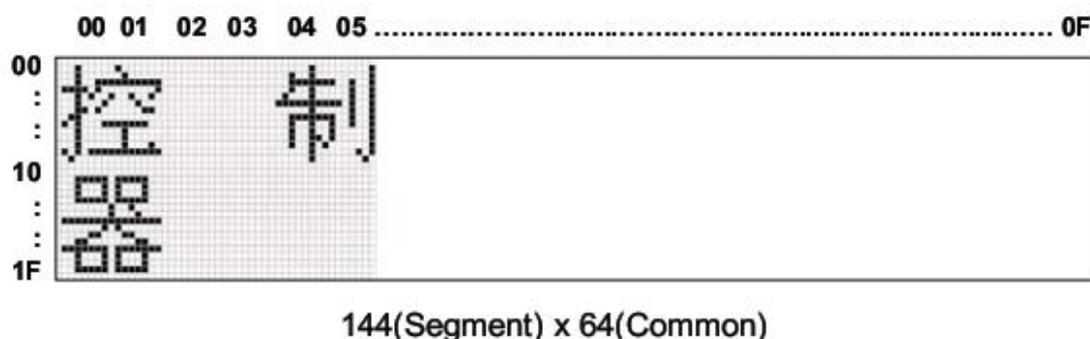


图 7-4: 模块光标位置设定的显示范例

### 8-3-2 光标显示与闪烁

模块可以透过光标控制缓存器(CURCR)来控制光标显示的开(光标显示)或关(光标不显示)、光标闪烁与否, 光标闪烁时间则由闪烁设定缓存器(BLTR)来决定, 可以由 8 Frames 到 128 Frames。

### 8-3-3 光标高度与宽度

模块在做文字显示时, 光标的高度依不同使用者的需要, 提供了 1~16 Pixel 的高度设定, 请参考图 7-5。同时提供一光标归位的设定, 让光标回到最左上方。

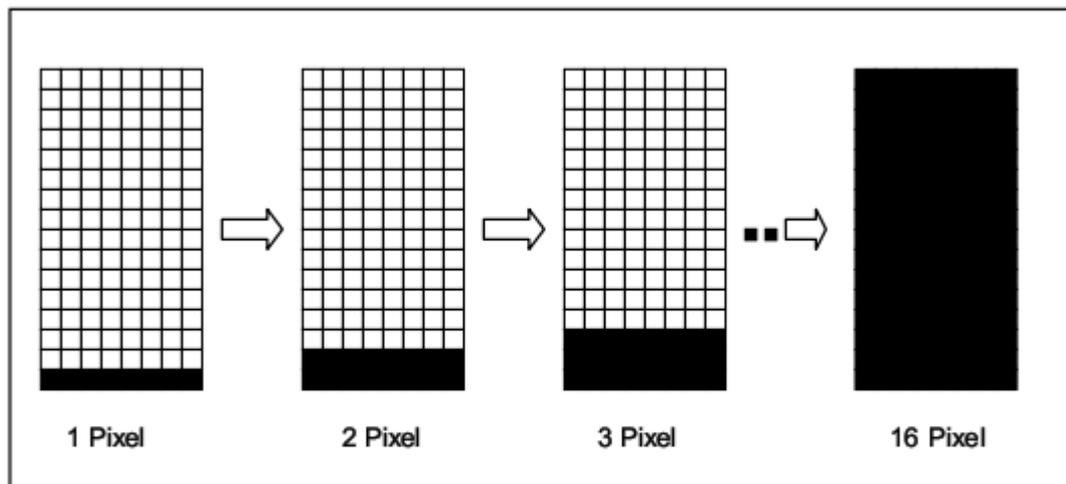


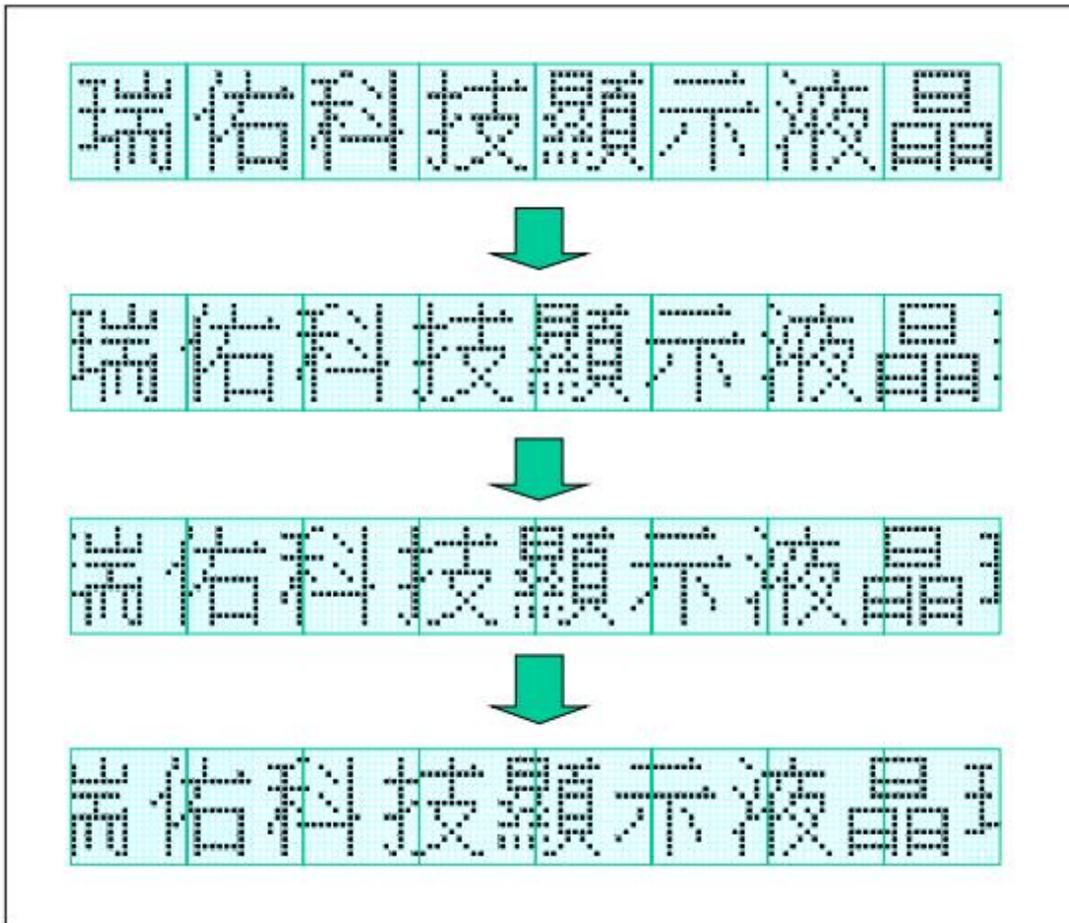
图 7-5: 光标高度之设定

### 8-4 显示窗口

模块的显示窗口(Display Window)就是实际 LCD 面板的大小, 可以由系统设定缓存器(SYSR)来设定, 显示范围为 144(Segment)x64(Common), 另外提供一 Common(Com S)当作 Icon 使用, 因此可以有 144 个 Icon, 在读写 Com S 时, 游标 Y 位置缓存器(Y-CUR)须要设定成 50h, 再由游标 X 位置缓存器(X-CUR)决定选择的 Icon。

### 8-5 水平卷动

模块针对某些显示画面的需求, 提供了水平卷动(Scroll)的功能, 透过缓存器的设定, 它可以允许使用者指定水平卷动的范围, 同时也可以指定每次卷动的位移量是多少 Pixel 及卷动的的速度。如图 7-7 所示, 为向左作水平卷动每次 2 Pixel 的效果。



**图 7-7：水平卷动**

模块除了允许向左或向右作水平卷动，也可以利用水平卷动 Buffer，达到向左或向右作水平移动(Shift)的效果，例如向左作水平移动时，可以先将文字或 Data 存入在水平卷动 Buffer，然后 Buffer 内的数据就会随着水平移动陆续移出，在水平移动 16 个 Pixel 后，藉由 MPU 软件再将文字或 Data 存入在水平卷动 Buffer，如此循环动作，可以让 LCD Panel 显示出向左移动的效果，如图 7-8 所示，为向左作水平移动每次 8 Pixel 的效果，其中右边的灰色区域代表水平卷动 Buffer，不会在 LCD Panel 的显示范围内。

有关卷动的详细使用方式及缓存器设定，请参考应用手册或例程。

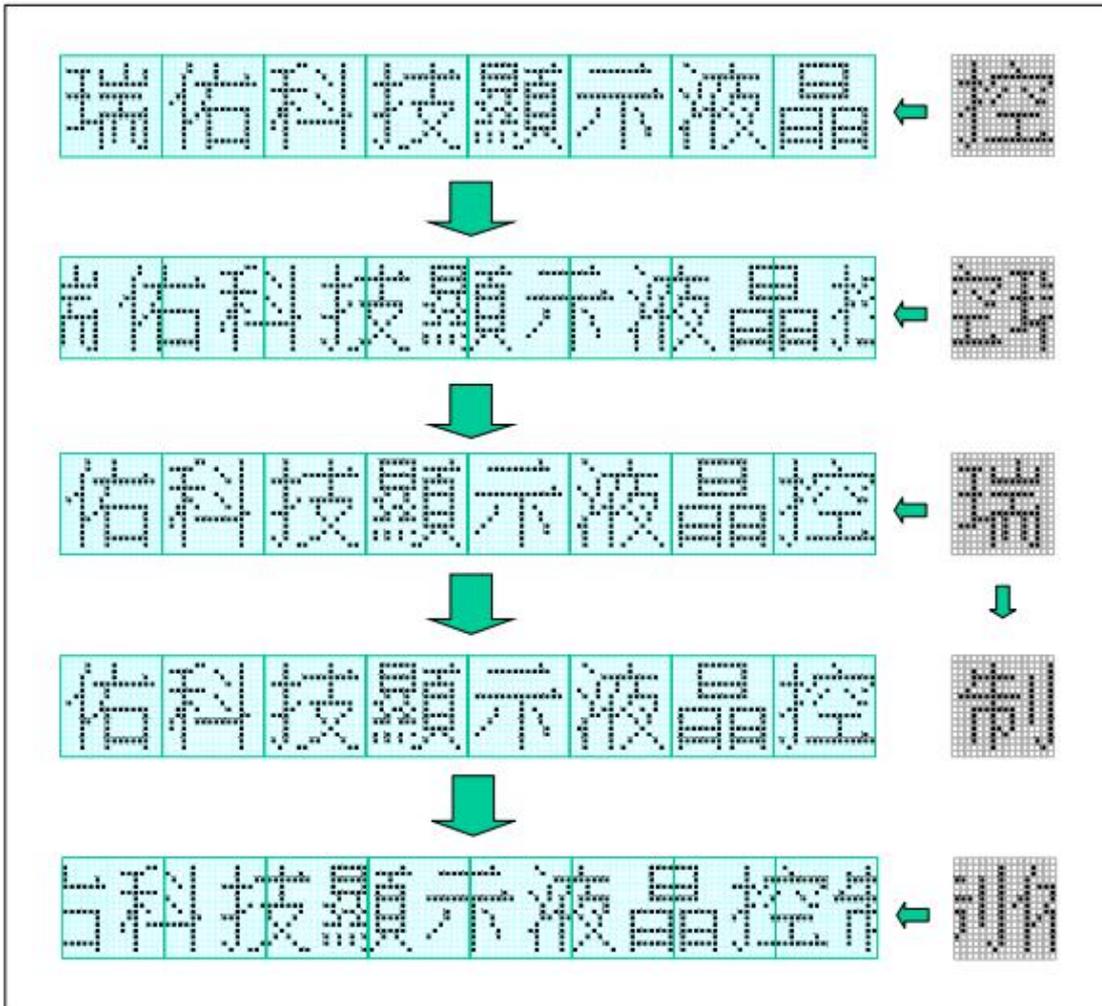


图 7-8: 水平移动

### 8-6 垂直卷动

4X9E 也提供了垂直卷动(Scroll) 与垂直移动(Shift)的功能, 同样的透过缓存器的设定, 它可以允许使用者指定垂直卷动的范围、每次卷动的位移量是多少 Pixel 及卷动的的速度。如图 7-9 所示, 为向上作垂直卷动每次 2 Pixel 的效果。

## 9. 电气特性

### 9-1 Absolute Maximum Ratings

表 9-1

Parameter	Symbol	Rating	Unit
Supply Voltage Range	$V_{DD}$	-0.3 to 6.5	V
Input Voltage Range	$V_{IN}$	-0.3 to $V_{DD}+0.3$	V
External VLCD Voltage Range	$V_{LCD}$	-0.3 to 14	V
Operation Temperature Range	$T_{OPR}$	-40 to 85	°C
Storage Temperature Range	$T_{ST}$	-55 to 125	°C

本模块使用内部升压，用户不必理会 EXTERNAL VLCD。

### 9-2 DC 特性

表 9-2

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Operating Voltage	$V_{DD}$	2.8	3.3	3.8	V	
VLCD Voltage	$V_{LCD}$	6	–	12	V	
Input High Voltage	$V_{IH}$	$0.8 \times V_{DD}$	–	$V_{DD}$	V	
Input Low Voltage	$V_{IL}$	0	–	$0.2 \times V_{DD}$	V	
Output High Voltage	$V_{OH}$	$0.8 \times V_{DD}$	–	$V_{DD}$	V	
Output Low Voltage	$V_{OL}$	0	–	$0.2 \times V_{DD}$	V	
Input Leakage Current	$I_{IL}$	-1	–	+1	μA	$V_{IN} = V_{DD}$
Output Leakage Current	$I_{OL}$	-3	–	+2	μA	$V_{IN} = V_{DD}$
Oscillator Frequency	$F_{CL}$	85	95	105	Khz	
Operating Mode Current (Normal Mode Current)	$I_{SB}$	0.29	0.50	1.1	mA	Min.:*1 Typ.:*2 Max.:*3
Display Off Current	$I_{DISPLAY}$	–	197	–	μA	
Sleep Mode Current	$I_{SLEEP}$	–	0.2	–	μA	

\*本模块使用内部升压，用户不必理会 VLCD Voltage。这里的  $V_{DD}$  为模块 IC 的电压。本类模块有普通为 3.3V。亦可生产提供 5V 电压的模块。

### 9-3 Timing 特性

#### 9-3-1 并行接口

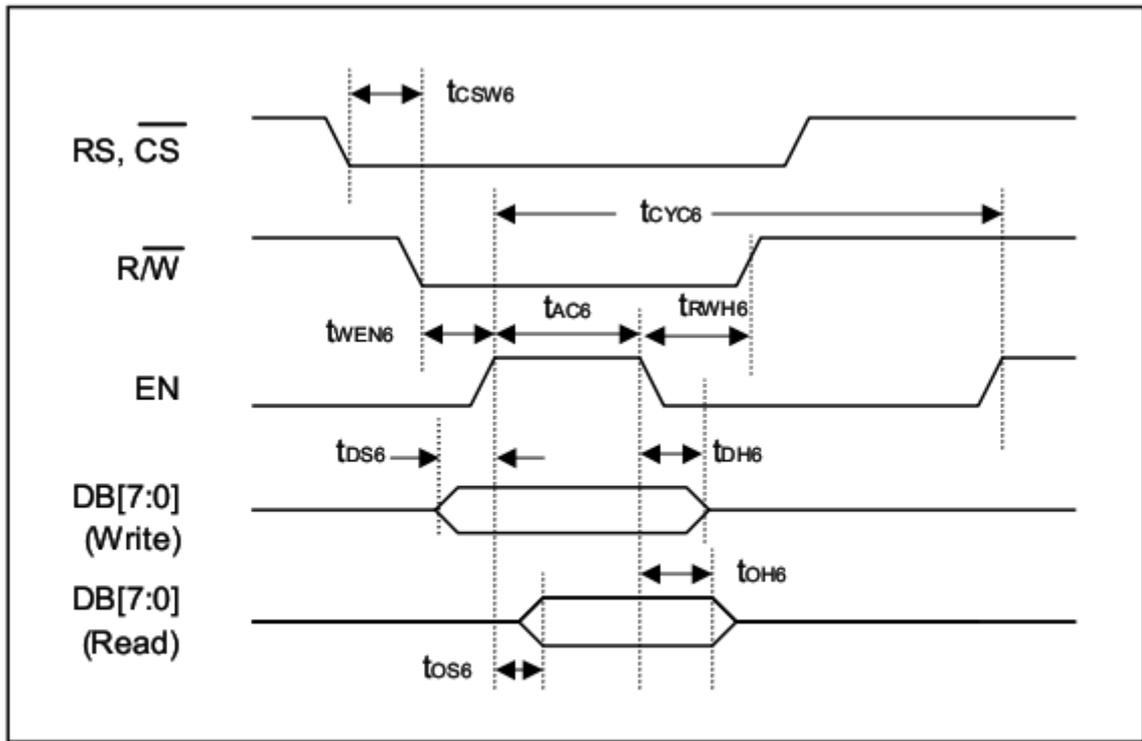


图 9-1: 6800 MPU 时序

## 6800 MPU Interface

表 9-3

Item	Signal	Symbol	Condition	Rating		Unit
				Min.	Max.	
Address Setup Time	RS, $\overline{CS}$	$t_{csw6}$		0	--	ns
Read/Write Setup Time	$\overline{R/W}$	$t_{wen6}$		10	--	
Read/Write Hold Time		$t_{rwh6}$		10	--	
Enable Access Time	EN	$t_{ac6}$		90	--	
Access Cycle Time		$t_{cyc6}$	Command Cycle	200	--	
			Data Cycle	400	--	
Write Data Setup Time	DB[7..0]	$t_{ds6}$		10	--	
Write Data Hold Time		$t_{dh6}$		10	--	
Read Data Access Time		$t_{os6}$		30	50	
Read Data Hold Time		$t_{oh6}$		10	--	

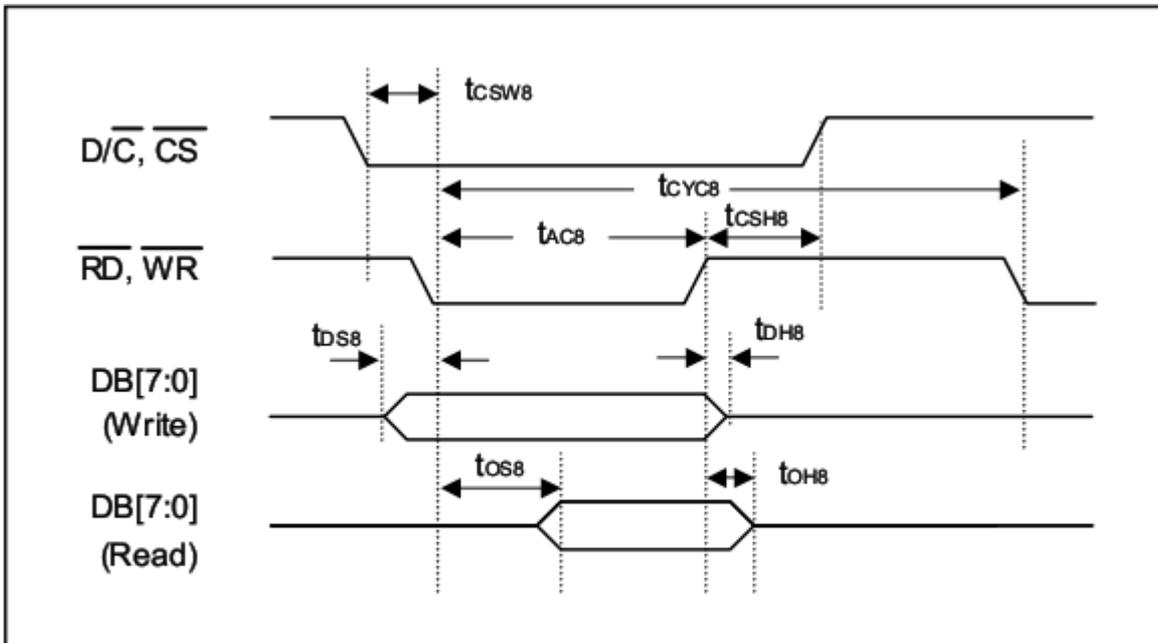


图 9-2: 8080 MPU 时序

8080 MPU Interface

表 9-4

Item	Signal	Symbol	Condition	Rating		Unit
				Min.	Max.	
Address Setup Time	$\overline{RS}, \overline{CS}$	$t_{CSW8}$		10	--	ns
Address Hold Time		$t_{SH8}$		10	--	
Read/Write Access Time	$\overline{RD}, \overline{WR}$	$t_{AC8}$		90	--	
Access Cycle Time		$t_{CYC8}$	Command Cycle	200	--	
			Data Cycle	400	--	
Write Data Setup Time	$DB[7..0]$	$t_{DS8}$		10	--	
Write Data Hold Time		$t_{DH8}$		10	--	
Read Data Setup Time		$t_{OS8}$		30	50	
Read Data Setup Time		$t_{OH8}$		10	--	

**8080MPU 读写时序例程:**

```
//例程测试环境: 用 AT89S52, 晶振 12M
void lcd_regwr(uchar regnada) //写缓存器
{
```

```
    lcd_regnada = regnada;
    lcd_cs = 0; // chip enable.
    lcd_rd = 1; //
    lcd_rs = 0; // rs = 0;
    lcd_wr = 0; // wr = 0;
    ;
    lcd_wr = 1; // wr = 1;
    lcd_rs = 1; // rs = 1;
```

```

    lcd_cs =1; // chip disable.
}

//.....
void lcd_datawrite(uchar wrdata) small
{
    //写资料到 DDRAM

    chk_busy();          //check busy
    lcd_data = wrdata;
    lcd_cs =0; // chip enable.
    lcd_rd = 1; //
    lcd_rs = 1; // rs = 1;
    lcd_wr = 0; // wr = 0;
    ;
    lcd_wr = 1; // wr = 1;
    lcd_rs = 1; // rs = 1;
    lcd_cs =1; // chip disable.
}
//.....
uchar lcd_regread(uchar regname) small
{
    //读缓存器
    uchar reg_rddata;
    lcd_regwr(regname);
    lcd_data = 0xff;
    lcd_cs =0; // chip enable.
    lcd_wr = 1; // wr = 1;
    lcd_rs = 0; // rs = 0;
    lcd_rd = 0; // rd = 0;
    ;
    reg_rddata = lcd_data;
    lcd_rd = 1; // rd = 1;
    lcd_rs = 1; // rs = 1;
    lcd_cs =1; // chip disable.
    return(reg_rddata);
}

//.....
void chk_busy (void)          //忙检测
{
    unsigned char temp;
    do
    {
        temp = lcd_regread(0x0f);
    }while(temp & 0x80);
}

```

}

9-3-2 串行接口

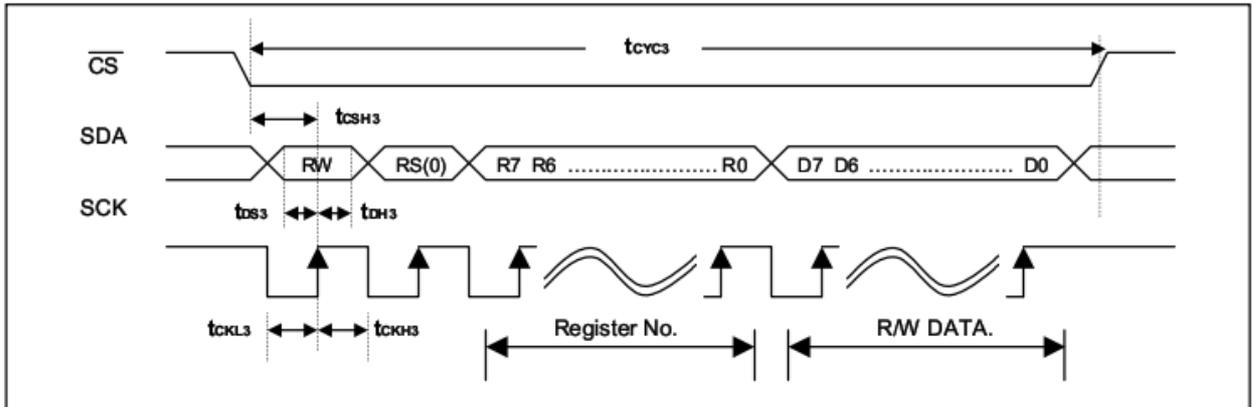


图 9-3: 3-Wire 时序

3-Wire Serial Interface

表 9-5

Item	Signal	Symbol	Condition	Rating		Unit
				Min.	Max.	
Access Time	$\overline{\text{CS}}$	$t_{\text{cyc3}}$		3.6	—	$\mu\text{s}$
$\overline{\text{CS}}$ Setup Time		$t_{\text{csh3}}$		20	—	ns
Clock Low Pulse Width	SCK	$t_{\text{ckL3}}$		100	—	
Clock High Pulse Width		$t_{\text{ckH3}}$		100	—	
Data Setup Time	SDA	$t_{\text{ds3}}$		20	—	
Data Hold Time		$t_{\text{dH3}}$		10	—	

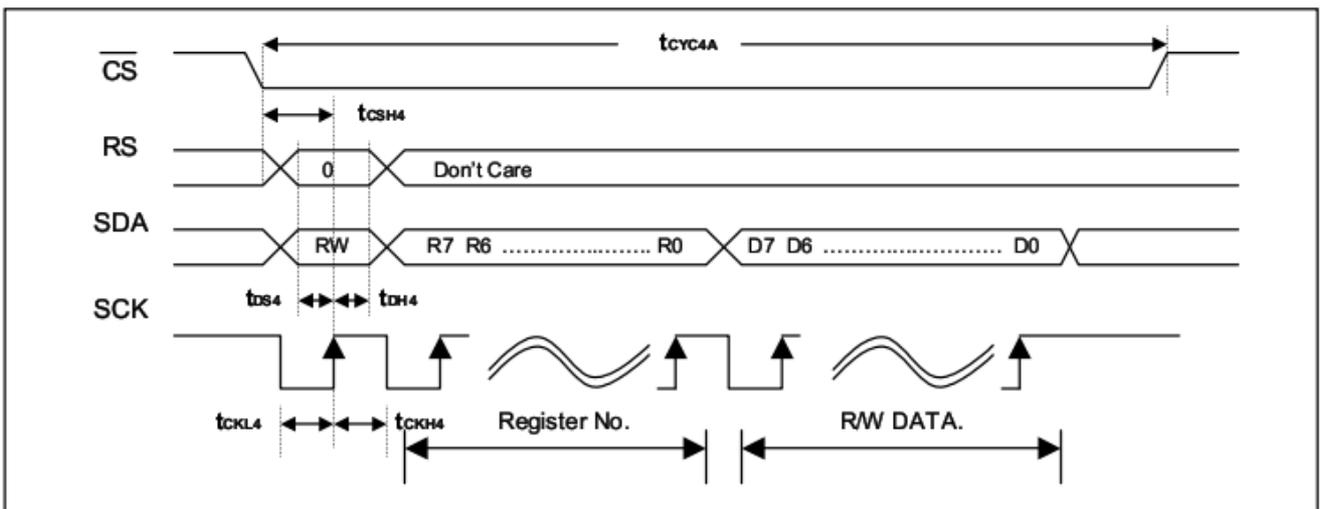


图 9-4: 4-Wire(A-Type) 时序

## 4-Wire(A-Type) Serial Interface

表 9-6

Item	Signal	Symbol	Condition	Rating		Unit
				Min.	Max.	
Access Time	$\overline{\text{CS}}$	$t_{\text{CYC4A}}$		3.4	--	$\mu\text{s}$
CS Setup Time		$t_{\text{CSH4}}$		20	--	ns
Clock Low Pulse Width	SCK	$t_{\text{CKL4}}$		100	--	
Clock High Pulse Width		$t_{\text{CKH4}}$		100	--	
Data Setup Time	SDA, RS	$t_{\text{DS4}}$		20	--	
Data Hold Time		$t_{\text{DH4}}$		10	--	

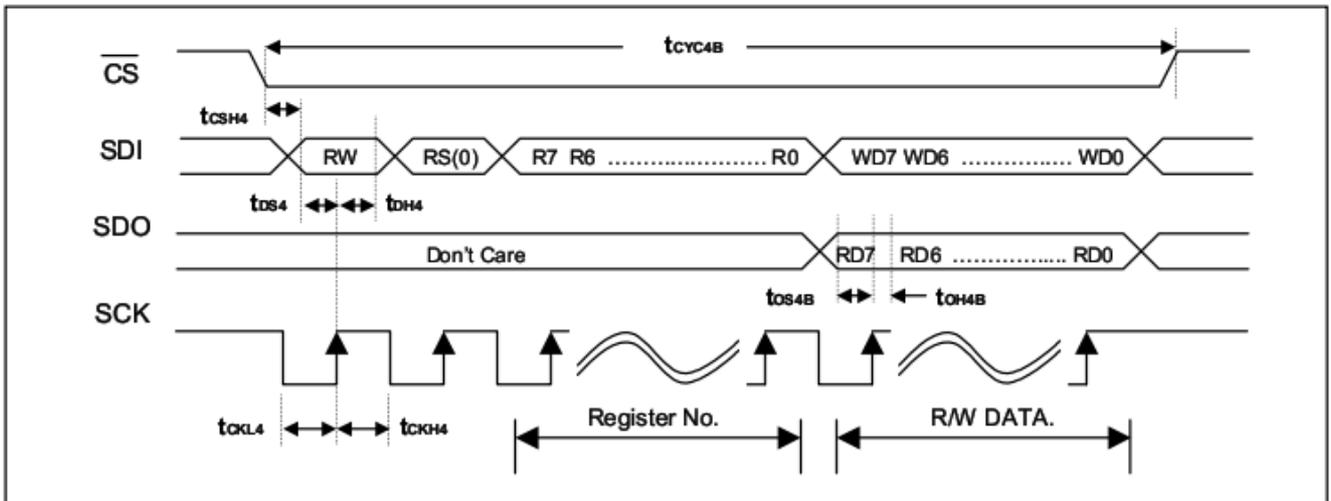


图 9-5A: 4-Wire(B Type) 时序

## 4-Wire(B-Type) Serial Interface

表 9-7A

Item	Signal	Symbol	Condition	Rating		Unit
				Min.	Max.	
Access Time	$\overline{\text{CS}}$	$t_{\text{CYC4A}}$		3.6	--	$\mu\text{s}$
CS Setup Time		$t_{\text{CSH4}}$		20	--	ns
Clock Low Pulse Width	SCK	$t_{\text{CKL4}}$		100	--	
Clock High Pulse Width		$t_{\text{CKH4}}$		100	--	
Data Write Setup Time	SDI	$t_{\text{DS4}}$		20	--	
Data Write Hold Time		$t_{\text{DH4}}$		10	--	
Data Read Setup Time	SDO	$t_{\text{OS4B}}$		20	--	
Data Read Hold Time		$t_{\text{OH4B}}$		10	--	

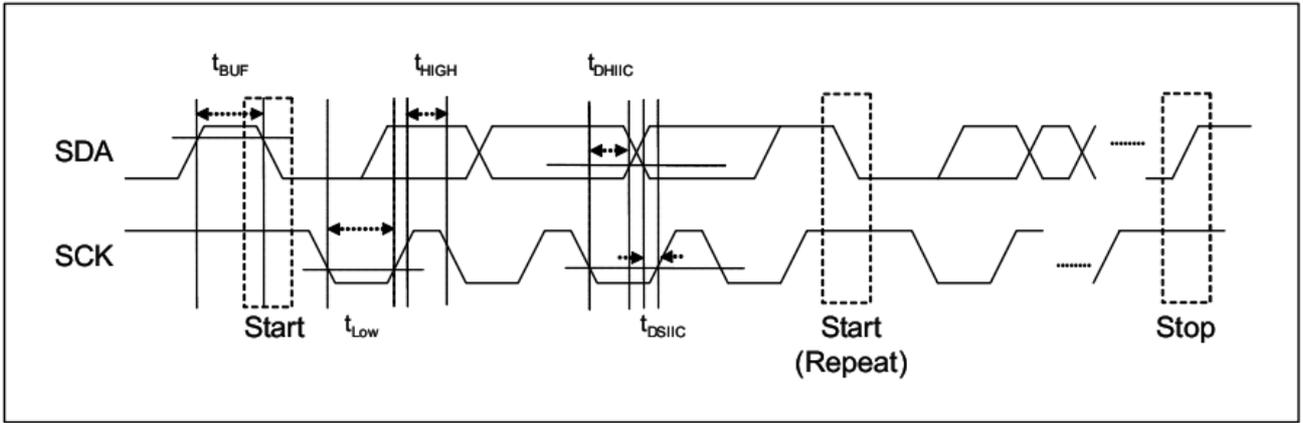


图 9-5B: IIC 时序

IIC Serial Interface

表 9-7B

Item	Signal	Symbol	Rating		Unit
			Min	Max	
SCK Clock Frequency	SCK	$f_{SCL}$	--	500	KHz
Bus Free Time Between STOP and START	SCK/SDA	$t_{BUF}$	1	--	us
LOW Period of SCK Clock	SCK	$t_{Low}$	200	--	ns
HIGH Period of SCK Clock	SCK	$t_{High}$	200	--	ns
Data Setup Time	SCK/SDA	$t_{DSIC}$	100		ns
Data Hold time	SCK/SDA	$t_{DHIC}$	100		ns

9-3-3 Reset 界面

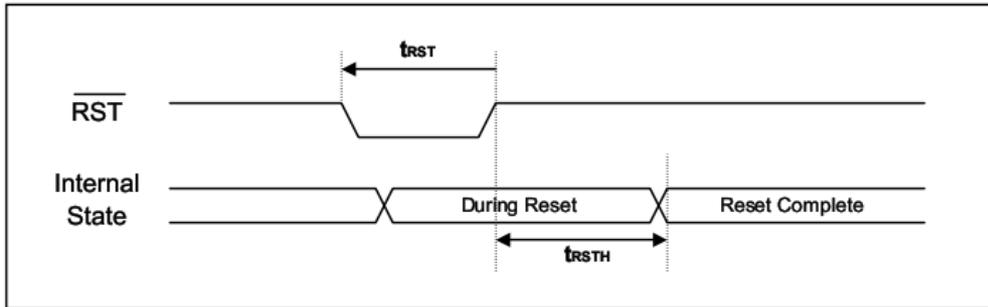


图 9-6 : Reset 时序

表 9-8

Item	Signal	Symbol	Condition	Rating		Unit
				Min.	Max.	
Reset Pulse Width	$\overline{RST}$	$t_{rST}$		30	--	ms
Reset Complete Hold Time	$\overline{RST}$	$t_{rSTH}$	$F_{CL} = 100KHz$ (Internal RC Oscillator)	150	--	ms

