

JD320240DZK 中文字库图形液晶显示模块 使用说明书(RA8802 控制)

感谢您关注和使用我们的 JD320240DZK 液晶显示器产品，欢迎您提出您的要求、意见和建议，我们将竭诚为您服务、让您满意。您可以浏览 www.szjingdong.COM 了解最新的产品与应用信息，或拨打热线电话 13923757716 以及向 szsjddz@163.com 邮箱发 E-mail 获取具体的技术咨询与服务。

目 录

一、简介	3
二、外形尺寸和引脚说明.....	4
三、读/写操作时序	6
四、缓存器功能表	4
五、ASCII字型区块	4
六、指令时间	17

一. 简介

JD320240DZK 采用RA8802 做控制器, 可以显示15行每行20个16x16 点阵的汉字, 支持320x240 点阵的LCD 面板, 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512KByte 的ROM 字形码, 可以显示中文字型、数字符号、英日欧文等字母, 并且内建双图层(Two Page)的显示内存。在字型方面有多种字号可供选择使用, 如16x16、32x32、48x48、64x64 及不同比例的混合显示模式, 同时内建的512Byte SRAM 提供了自行造字的功能。在文字模式中, 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。

支持文字与绘图两种混和显示模式

支持2 Page 显示模式(And, Or, Nor, Xor), 内建两个4.8K Byte 的显示RAM (Display Data RAM), 共9.6K Byte RAM, 并且可做成4 阶的显示效果。

内建512KByte ROM, 控制IC RA8822 分带繁体字库IC 和带简体字库IC, 其中标准繁体中文BIG5 码, 包含13,094 个常用与次常用字型、408 个特殊字与两组ASCII CODE, 简体字库储存7602 个标准GB 码的简体中文。模块出厂默认为使用简体字库IC

提供全角(16x16)与半角(8x16)文字显示模式

带光标、反白、闪烁功能, 且光标高度与宽度可调

支持屏幕水平卷动及垂直卷动功能

内建512Byte SRAM 可自行造字

提供中/英文文字对齐功能

显示字型可放大到32x32、48x48 或64x64, 以及混合显示模式

支持可将字型由ROM 直接读出使用

内建粗体字形与行距设定

主要技术参数和性能:

- 1、 电源 VDD: +5.0V
- 2、 显示内容: 320 (列) × 240 (行) 点
- 3、 显示类型: STN / 蓝膜 、 负显 / 1/240 Duty / 1/17BAIS
- 4、 显示视角: 6 0°CLOCK
- 5、 模块尺寸: 160.00X109.00X(MAX)12.0 (mm)
- 6、 视域窗口: 121.00X92.00 (mm)
- 7、 点大小 : 0.33X0.33 (mm)
- 8、 工作温度: -20° C ~ +70° C;
- 9、 存储温度: -30° C ~ +80° C

表2—1：JD320240DZK 引脚说明

引脚	名称	方向	说明
1	VSS	--	电源地（0V）
2	VDD	--	电源正（+3V/5V）
3	V0	--	LCD 驱动电压输入端
4	RS	I	H：存取DDRAM；L：存取缓存器。
5	/WR(R/W)	I	6800系列：读/写脚(R/W)，H：读，L：写。 8080系列：写入脚(/WR)，低有效。
6	/RD(EN)	I	6800系列：使能脚(EN)，高有效。 8080系列：读入脚(/RD)，低有效。
7	/CS1	I	当/CS1为低和CS2为高时，模块处于致能，可接受指令，反之不可接收指令。
8	CS2	I	当/CS1为低和CS2为高时，模块处于致能，可接受指令，反之不可接收指令。
9	BUSY	0	用以回应模块内部的执行使用状况，可设成高或低电平触发。
10	INT	0	用以回应模块内部的中断状况，可设成高或低电平触发。
11	/RST	I	复位信号，低有效。
12	DB0	I/O	数据 0
13	DB1	I/O	数据 1
14	DB2	I/O	数据 2
15	DB3	I/O	数据 3
16	DB4	I/O	数据 4
17	DB5	I/O	数据 5
18	DB6	I/O	数据 6
19	DB7	I/O	数据 7
20	VEE	--	LCD 驱动电源
21	A	--	LED 背光电源正（+5V）
22	K	--	LED 背光电源地（0V）

三. 读/写操作时序

图3-1 是MCU 与JD320240DZK 间的系统时序图，在JD320240DZK 的定义中，/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，RS 为“L” 时表示对缓存器下命令，也就是对JD320240DZK 的缓存器进行读写的动作(Register Access Cycle)，而RS 为“H” 时表示对Display RAM 进行Data 读写的动作(Data Access Cycle)。

下面图3-1 表示如果是对缓存器进行读取动作，MCU 必须透过数据线先送出缓存器的地址，然后才能在数据线上读取缓存器的资料，如果是对缓存器进行写入动作，MCU 必须透过数据线先送出缓存器的地址，然后再送出要写入的资料。当MCU 对JD320240DZK Display RAM 进行资料的读取动作，MCU能直接在数据线上读取Display RAM 的资料，如果MCU 对Display RAM 进行资料的写入动作，MCU 则直接在数据线上送出要写入的资料。

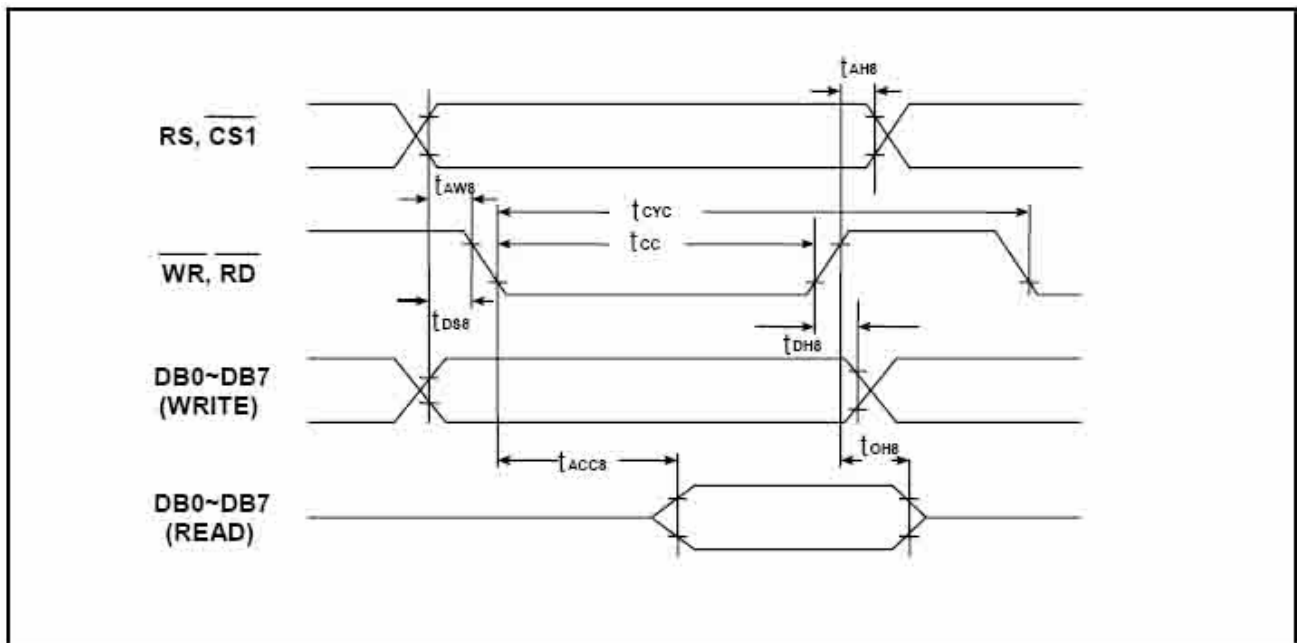


图3-1: MCU 对JD320240DZK 缓存器/DATA 进行读取/写入动作

Signal	Symbol	Parameter	Rating		Unit	Condition
			Min	Max		
RS, CS1#	t_{AH8}	Address hold time	10	--	ns	System Clock: 8MHz Voltage: 3.3V
	t_{AW8}	Address setup time	63	--	ns	
WR#, RD#	t_{cyc}	System cycle time	800	--	ns	
	t_{cc}	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	

四. 缓存器功能表

JD320240DZK 缓存器功能表				
Bit	Description	Text/Graph	Default	Access
REG [00h] Whole Chip LCD Controller Register (WLCR)				
7-6	电源模式(Power Mode) 11: 正常模式(Normal Mode) 模块的所有功能都可以使用(Available)。 10: 等待模式(Standby Mode) 01: 睡眠模式(Sleep Mode) 除了缓存器的读写外, 其他LCD的显示和DDRAM、ROM的存取将被禁止。 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, 模块将回复至正常模式。		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	设置自动重置功能 当此位处于致能(设为1)时, 若RA8802/8820 无法在2 毫秒内得到一完整指令或数据时, 则RA8802/8820 会忽视这个指令或数据 1: 致能自动重置功能 0: 禁能自动重置功能		0h	R
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 1: 屏幕开启 0: 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[80h] 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: 屏幕反白显示, DDRAM 内的资料会被全部反相	Text/Graph	1h	R/W
REG [08h] Misc. Register (MISC)				
7	保留		0h	R
6	CLK_OUT 致能控制 1: 致能 0: 禁能 注: 此功能测试使用, 不提供给用户		1h	R/W
5	切换窗口模式 1: 工作窗口 (Active window) 0: 显示窗口 (Display window)		0h	R
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		0h	R/W
3-2	保留		0h	R
1-0	系统时脉选择 0 0: 1MHz 0 1: 1MHz		0h	R/W

	1 0: 4MHz 1 1: 8MHz			
REG [10h] Whole Chip Cursor Control Register (WCCR)				
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	1h	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	1h	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	0h	R/W
2	光标显示On/Off 设定 1: 设定光标显示On 0: 设定光标显示Off	Text/Graph	0h	R/W
1	光标闪烁控制 1: 光标闪烁, 闪烁时间由缓存器[80h] BTMR 来决定 0: 光标不闪烁	Text/Graph	0h	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, 光标为二个字节宽度(16 个Pixel) 0: 光标固定为一个字节的宽度(8 个Pixel)	Text	0h	R/W
REG [18h] Cursor Height and Lines Distance Register (CHLD)				
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 [20h] Active Window Right Register (AWRR)				
7-6	保留		0h	R
5-0	设定工作窗口(Active window)右边位置 → Segment-Right		XXh	R/W
REG [30h] Active Window Bottom Register (AWBR)				
7-0	设定工作窗口(Active window)底边位置 → Common-Bottom		XXh	R/W
REG [40h] Active Window Left Register (AWLR)				
7-6	保留		0h	R
5-0	设定工作窗口(Active window)左边位置 → Segment-Left		XXh	R/W
REG [50h] Active Window Top Register (AWTR)				
7-0	设定工作窗口(Active window)顶边位置 → Common-Top		XXh	R/W
Note: REG [20h, 30h, 40h, 50h] 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 (REG[20h and 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。				
REG [28h] Display Window Right Register (DWRR)				
7-6	保留		0h	R
5-0	设定显示窗口(Display Window)右边位置 □ Segment-Right Segment_ Right = (Segment Number / 8) – 1 RA8802: 如果LCD Panel 为320x240, 则此缓存器的值为: (320 / 8) – 1 = 39 = 27h RA8820: 如果LCD Panel 为240x160, 则此缓存器的值为:		XXh	R/W

	$(240 / 8) - 1 = 29 = 1Dh$			
REG [38] Display Window Bottom Register (DWBR)				
7-0	设定显示窗口(Display Window) 底边位置 □ Common_Bottom Common_Bottom = LCD Common Number -1 RA8802: 如果LCD Panel 为320x240, 则此缓存器的值为: $240 - 1 = 239 = EFh$ RA8820: 如果LCD Panel 为240x160, 则此缓存器的值为: $160 - 1 = 159 = 9Fh$		XXh	R/W
REG [48] Display Window Left Register (DWLR)				
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 [58] Display Window Top Register (DWTR)				
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
注: 缓存器的设定, 请遵照以下的规范: 1. DWRR ≥ AWRR ≥ CPXR ≥ AWLR ≥ DWLR 2. DWBR ≥ AWBR ≥ CPYR ≥ AWTR ≥ DWTR				
REG [60h] Cursor Position X Register (CPXR)				
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)				
7-0	设定光标Common 地址		0h	R/W
REG [80h] Blink Time Register (BTMR)				
7-0	光标闪烁时间设定 闪烁时间 = [80h] Bit[7..0] × (1/Frame_Rate) Frame Rate 的设定是通常依照LCD 面板所提供的最佳值。		23h	R/W
REG [90h] Shift Clock Control Register (SCCR)				
7-0	设定 XCK 讯号周期 SCCR = (SCLK × DBW) / (Column × Row × FRS) SCLK: RA8802/8820 系统频率(System Clock) (单位: Hz) DBW: LCD Driver 的Data Bus 宽度(单位: Bit) Column: LCD 面板的Segment 大小(单位: Pixel) Row: LCD 面板的Common 大小 (单位: Pixel) FRS: LCD 面板的Frame Rate(单位: Hz)		--	R/W
REG [A0h] Interrupt Setup & Status Register (INTR)				
7	忙碌状况指示 1: RA8802/8820 为忙碌状态, MCU 需暂时等候到忙碌状态终止。 0: RA8802/8820 为闲置状态, 随时可接受MCU 存取。		0h	R (Read Clear)
6	触控屏幕侦测 1: 触控屏幕有侦测到接触(Touch) 0: 触控屏幕未侦测到接触(Touch)		0h	R (Read Clear)
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: 光标列等于缓存器INTY 0: 光标列不等于缓存器INTY		0h	R (Read Clear)
3	忙碌中断屏蔽 1: 致能BUSY 去产生中断输出 0: 禁能BUSY 去产生中断输出		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, 或致能BUSY。		0h	R/W

	0: 禁能INTX 中断			
0	设定缓存器[B8h]INTY 是否发生中断 1: 致能INTY 中断, 或致能BUSY。 0: 禁能INTY 中断		0h	R/W
REG [B0h] Interrupt Column Setup Register (INTX)				
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 [B8h] Interrupt Row Setup Register (INTY)				
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)				
7	触控屏幕功能启动 1: 致能 0: 禁能		1h	R/W
6	触控屏幕数据输出 1: 致能触控屏幕数据输出 0: 禁能触控屏幕数据输出		1h	R/W
5	保留		0h	R
4	触控状态位指示 1: 触控未发生 0: 触控发生		1h	R/W
3-0	触控屏幕控制位 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)			R/W
REG [C8h] Touch Panel Segment High Byte Data Register (TPXR)				
7-0	储存触控屏幕的行、列相对位置数据		00h	RO
REG [D0h] LCD Contrast Control Register (LCCR)				
7	LCD 亮度控制(DAC 功能) 1: 禁能 0: 致能		1h	R/W
6	DAC 写入致能 1: 不允许MCU 把数据写入Bit[4..0] 0: 允许MCU 把数据写入Bit[4..0]		1h	R/W
5	重置LCD 亮度调整控制功能 1: 正常操作 0: DAC 重置, 设定Iout 为 0μA		1h	R/W
4-0	设定DAC 输出电流Iout 的值(LCD 亮度控制) 0 0 0 0 0b → 0μA±0.2 uA (Min. Current) : : 1 1 1 1 1b → 540μA±140 μA (Max. Current)		0h	R/W
REG [E0h] Pattern Data Register (PNTR)				
7-0	设定写入到DDRAM 的数据 当缓存器[F0h]的bit3 为 '1', 模块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'。		0h	R/W
REG [F0h] Font Control Register (FNCR)				
7	字型ROM 的转换电路控制 1: 致能 0: 禁能		1h	R/W
6	内部/外部字型ROM 选择 1: 选择外部字型ROM 0: 选择内部字型ROM		0h	R/W
5-4	字型ROM 的语系选择 01: 选择繁体 (BIG5) 字型		00h	R/W

	10: 选择简体 (GB) 字型			
3	填写PNTR (REG [E0h]) 的数据到DDRAM 1: 开始写入 0: 未动作	Graph	0h	R/W
2	文字码的类别选择 1: 输入ASCII 码, 输出为符号等 0: 输入GB/BIG5 码, 输出为中文字	Text	0h	R/W
1-0	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		00h	R/W
注: 中文内码不论是GB 或BIG5 码都是由两个Byte 组成, 但是英文及一些符号ASCII 码只由一个Byte 组成 (00h~FFh), 通常模块将送到Display RAM 的Data(00h~9Fh)视为ASCII 码, 也就半角文字(8x16), 大于等于“A0h” 的视为全角码(如繁简中文)的高位, 必须再送一次低位内码, 才能显示全角字型。如果使用者有用到A0h~FFh 的ASCII 码, 则MCU 在送Data(ASCII 码)到Display RAM之前必须将缓存器[F0h] 的Bit2 设成“1”。				
REG [F1h] Font Size Control Register (FVHT)				
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 1 0: 三倍 1 1: 四倍		0h	R/W
3-0	保留		Fh	R

五. ASCII 字形区块

5.1 ASCII 字形区块0

b5-b0 b7-b4	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000		☺	☻	♥	♣	♠	♣	♠	◐	◑	♂	♀	♪	♫	✂	
0001	▶	◀	↕	!!	¶	§	=	±	↑	↓	→	←	↔	▲	▼	
0010		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
1000	Ç	ü	é	â	ä	ã	ä	ç	ê	ë	ë	ï	î	ï	Ä	Å
1001	Ê	æ	ø	ô	ö	ö	û	ü	ÿ	Ö	Ü	φ	£	¥	ℳ	℔
1010	ā	ī	ō	ū	ñ	Ñ	≡	≈	¿	¬	¬	½	¼	⅓	⅔	⅕
1011	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒
1100	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒
1101	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒
1110	α	β	Γ	π	Σ	σ	μ	τ	Φ	Θ	Ω	δ	∞	∅	∈	∩
1111	≡	±	≥	≤	∫	∫	÷	≈	°	.	.	√	n	2	■	

图 5-1: 内建 ASCII 区块 0

5.2 ASCII 字形区块1

b8-b0 b7-b4	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	€		, f	...	† ‡	ˆ	% Š	< œ	Ž							
0001		‘ ’ “ ”	•	–	—	˜	™ Š	> œ	Ž	Ÿ						
0010		ı	Ɔ	£	¥	ı	§	©	®	«	»	–	®			
0011	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
0100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
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	Ì	Í	Î
1011	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
1100	Ŕ	Ŗ	Ű	Ų	Ŵ	Ŷ	Ÿ	Š	Š	Š	Š	Š	Š	Š	Š	Š
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1110	ŕ	ŗ	ŭ	ŵ	ŷ	Ź	Ż	Ź	Ż	Ź	Ż	Ź	Ż	Ź	Ż	Ź
1111	đ	ñ	ň	õ	ô	õ	ö	÷	ř	ů	ű	ű	ű	ű	ű	t

图 5-2：内建 ASCII 区块 1

5.3 ASCII 字形区块2

b3-b0 b7-b4	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000																
0001																
0010		H	h	£	¥		Â	ä	ˆ	ı	Ş	Ğ	İ	–	Ž	
0011		°	h	²	³		µ	ñ	·	ı	ş	ğ	ı	ˆ	ž	
0100		À	Á	Â	Ã	Ä	Å	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
0101		Ñ	Ō	Ŏ	Ğ	Ö	×	Ğ	Ū	Ū	Ū	Ū	Ū	Ū	Ū	ß
0110		à	á	â	ã	ä	å	ç	è	é	ê	ë	ì	í	î	ï
0111		ñ	õ	ö	ö	ö	÷	ğ	û	û	û	û	û	û	û	ˆ
1000																
1001																
1010		À	Á	Â	Ã	Ä	Å	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1011		°	h	²	³		µ	ñ	·	ı	ş	ğ	ı	ˆ	ž	
1100		À	Á	Â	Ã	Ä	Å	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101		Ð	Ñ	Ō	Ŏ	Ğ	Ö	×	Ø	Ū	Ū	Ū	Ū	Ū	Ū	ß
1110		à	á	â	ã	ä	å	ç	è	é	ê	ë	ì	í	î	ï
1111		đ	ñ	õ	ö	ö	÷	ø	û	û	û	û	û	û	û	ˆ

图 5-3：内建 ASCII 区块 2

5.4 ASCII 字形区块3

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																

图 5-4：内建 ASCII 区块 3

六. 指令时间

说明在做读/写或是各种模式下写到内存所需的时间。可依使用者所设定的不同系统频率(SYS_CLK)，来决定各个指令所需要的时间。例如，SYS_CLK=8MHz，每个Clock 的时间= $1/\text{SYS_CLK}=125\text{ns}$ ，而写入缓存器所需的Clock 为3 个机械周期，所以对缓存器做读取或是写入时所需的时间约为 $125\text{ns} \times 3 \text{ lock}=375\text{ns}$ ，用此方式来计算指令所需的时间。

下列是说明各个指令动作所需的机械周期时间：

写入缓存器的时间为3 个机械周期

读取缓存器的时间为3 个机械周期

写入内存的时间为3 个机械周期

在绘图模式下写入内存的时间为3 个机械周期

在中文字型下写入一个字到内存的时间为35 个机械周期

在ASCII 字型下写入一个字到内存的时间为19 个机械周期

硬件清除屏幕所需的机械周期时间，公式： $3+(\text{Com} \times \text{Seg})/8$