

# YT24064B2 中文图形点阵液晶显示模块 使用说明书

## 目 录

1. 简介.....	3
2. 外形尺寸和引脚说明.....	4
3. 读写操作时序.....	5
4. 缓存器功能表.....	6
5. <b>ASCII</b> 字形区块.....	13
6. 指令时间.....	17

## 1. 简介

YT24064ZK 采用RA8822 做控制器,可以显示4行每行15个16x16 点阵的汉字,支持240x64 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

## 2. 外形尺寸和引脚说明

图2-1: YT24064ZK 模块外形尺寸图

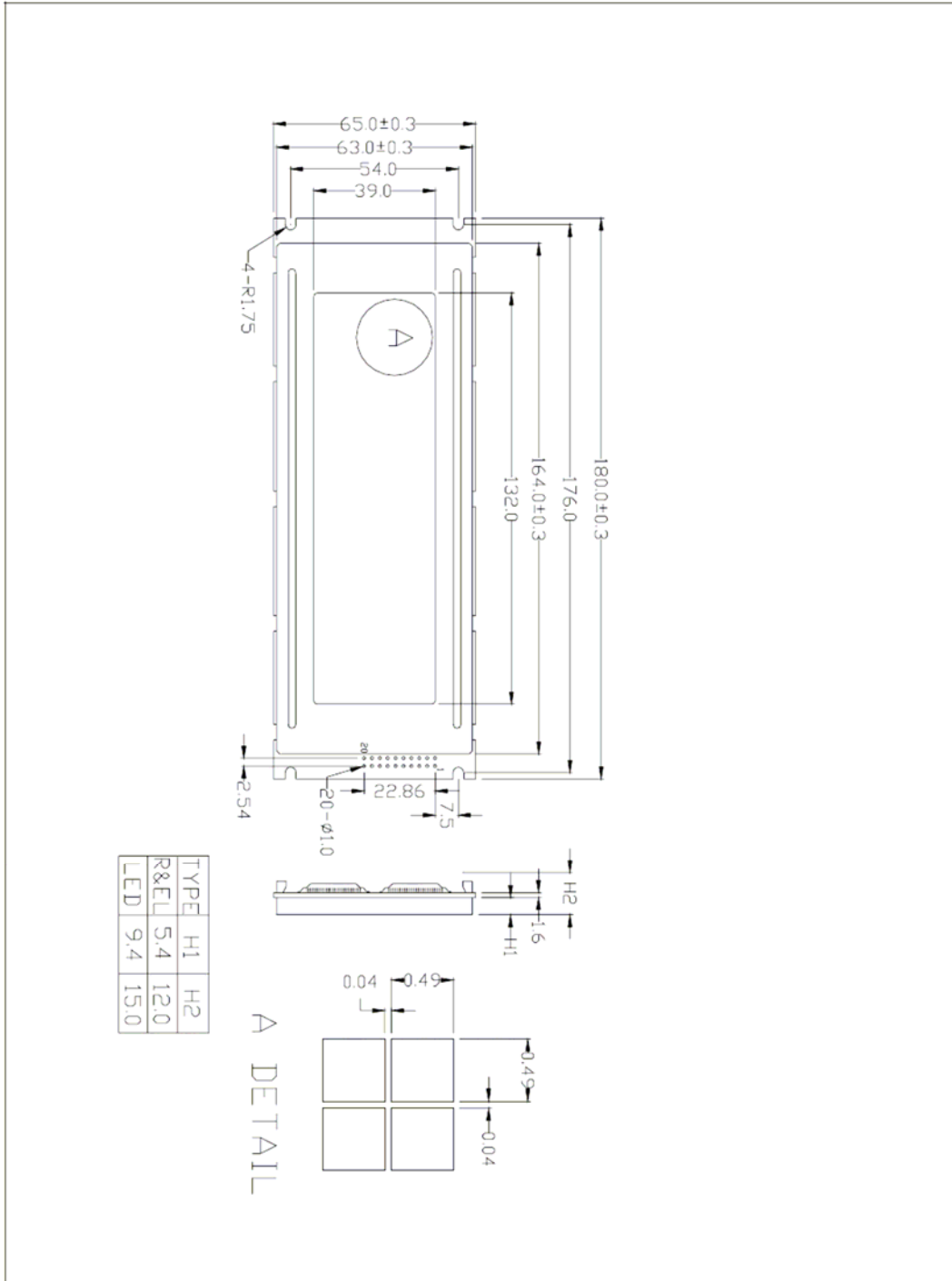


表2-1: YT24064ZK 引脚说明

引脚	名称	方向	说明
1	FG	--	外壳地
2	VSS	--	逻辑电源地 (0V)
3	VDD	--	逻辑电源正 (+5V)
4	VO	--	LCD 对比度调节电压
5	/WR	I	写入控制脚, 低有效。
6	/RD	I	读出控制脚, 低有效。
7	/CS1	I	当/CS1为低时, 模块处于致能, 可接受指令, 反之不可接收指令。
8	RS	I	H: 存取DDRAM; L: 存取缓存器。
9	NC	--	空脚
10	/RST	I	复位信号, 低有效。
11	DB0	I/O	数据 0
12	DB1	I/O	数据 1
13	DB2	I/O	数据 2
14	DB3	I/O	数据 3
15	DB4	I/O	数据 4
16	DB5	I/O	数据 5
17	DB6	I/O	数据 6
18	DB7	I/O	数据 7
19	NC	--	空脚
20	VEE	--	LCD 负电压输出
21	INT	O	用以回应模块内部的中断状况, 可设成高或低电平触发。
22	BUSY	O	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。
23	A	--	LED 背光电源正 (+5V)
24	K	--	LED 背光电源地 (0V)

### 3. 读写操作时序

图3-1 是MCU 与YT24064ZK 间的系统时序图, 在YT24064ZK 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” 时是表示对缓存器下命令, 也就是对YT24064ZK 的缓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)。

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YT24064ZK 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。

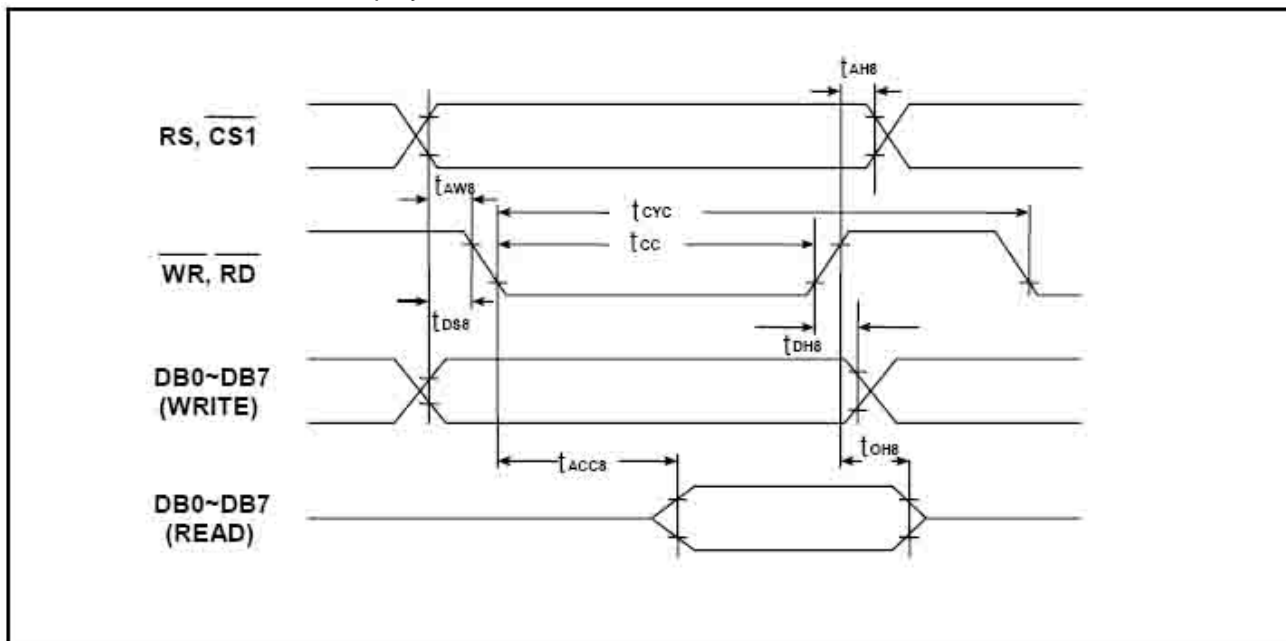


图3-1: MCU 对YT24064ZK 缓存器/DATA 进行读取/写入动作

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	$t_{D88}$	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	

## 4. 缓存器功能表

YT24064ZK 缓存器功能表				
Bit	Description	Text/Graph	Default	Access
<b>REG [00h] Whole Chip LCD Controller Register (WLCR)</b>				
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)。 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	保留		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
<b>REG [01h] Misc. Register (MISC)</b>				
7	保留		1h	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	保留		1h	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		1h	R/W
3-2	保留		0h	R
1-0	系统时脉选择 0 0: 3MHz 0 1: 4MHz 1 0: 8MHz 1 1: 12MHz		0h	R/W
<b>REG [02h] Advance Power Setup Register (APSR)</b>				
7-6	保留		0h	R
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
1	Scrolling Reset for Start 1: 致能 0: 禁能		0h	R/W
0	保留		0h	R
<b>REG [03h] Advance Display Setup Register (ADSR)</b>				
7-4	保留		8h	R
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 的自动卷动 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 的卷动或是Segment 的平移模式 1: Segment 的平移 0: Common 的卷动 扩展模式下(REG[12h] 的bit[6:4] = "110" 或"111"), 此位必须设为1。		0h	R/W
<b>REG [10h] Whole Chip Cursor Control Register (WCCR)</b>				
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: 设定光标显示On 0: 设定光标显示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[80h] 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, 光标为二个字节宽度(16 个Pixel) 0: 光标固定为一个字节的宽度(8 个Pixel)	Text	1h	R/W
<b>REG [11h] Cursor Height and Lines Distance Register (CHLD)</b>				
7-4	设定光标高度		2h	R/W
3-0	设定行与行的距离		2h	R/W
<b>REG [12h] Memory Access Mode Register (MAMR)</b>				
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	Graph	1h	R/W

	0: 先垂直移动再水平移动			
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。 Page1 Page2 灰度 ----- 0 0 Level1 1 0 Level2 0 1 Level3 1 1 Level4 1 1 0: 扩展模式(1), 此功能已屏蔽 1 1 1: 扩展模式(2), 此功能已屏蔽		1h	R/W
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 Page0 是用于支持自行造字功能, 若用到特殊字, 是字库内没 有的字型, 可利用 Page0 内建字库内容, 来提升MPU 的存取 效率。		1h	R/W
<b>REG [20h] Active Window Right Register (AWRR)</b>				
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
<b>REG [30h] Active Window Bottom Register (AWBR)</b>				
7-0	设定工作窗口(Active window)底边位置 → Common-Bottom		EFh	R/W
<b>REG [40h] Active Window Left Register (AWLR)</b>				
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
<b>REG [50h] Active Window Top Register (AWTR)</b>				
7-0	设定工作窗口(Active window)顶边位置 → Common-Top		0h	R/W
<b>REG [21h] Display Window Right Register (DWRR)</b>				
7-6	保留		0h	R
5-0	设定显示窗口(Display Window)右边位置 → Segment-Right $\text{Segment-Right} = (\text{Segment Number} / 8) - 1$ 对YT24064ZK LCD Panel 为240x64, 则此缓存器的值为: $(240 / 8) - 1 = 29 = 1Dh$		27h	R/W
<b>REG [31] Display Window Bottom Register (DWBR)</b>				
7-0	设定显示窗口(Display Window)底边位置 → Common-Bottom $\text{Common-Bottom} = \text{LCD Common Number} - 1$ 对YT24064ZK LCD Panel 为240x64, 则此缓存器的值为: $64 - 1 = 63 = 3Fh$		EFh	R/W
<b>REG [41] Display Window Left Register (DWLR)</b>				
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
<b>REG [51] Display Window Top Register (DWTR)</b>				
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				

<b>REG [60h] Cursor Position X Register (CPXR)</b>				
7-6	保留		0h	R
5-0	设定光标Segment 地址		0h	R/W
<b>REG [61h] Begin Segment Position Register (BGSg)</b>				
7-6	保留		0h	R
5-0	显示 Segment 开始的位置		0h	R/W
<b>REG [70h] Cursor Position Y Register (CPYR)</b>				
7-0	设定光标Common 地址		0h	R/W
<b>REG [71h] Shift action range, Begin Common Register (BGCM)</b>				
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
<b>REG [72h] Shift action range END Common Register (EDCM)</b>				
7-0	在水平移动模式下, 设定区块移动的结束Common 位置		EFh	R/W
<b>REG [80h] Blink Time Register (BTMR)</b>				
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
<b>REG [81h] Frame Rate Polarity Change at Common_A Register (FRCA)</b>				
7-0	保留		0h	R
<b>REG [90h] Shift Clock Control Register (SCCR)</b>				
7-0	设定屏幕更新周期 $SCCR = (SCLK \times DW) / (Seg \times Com \times FRM)$ SCLK: 系统频率(System Clock) (单位: Hz) DW: 4 (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), 通常为70Hz 限制条件SCCR $\geq$ 4 对YT24064ZK: Seg = 240; Com = 64		4h	R/W
<b>REG [91h] Frame Rate Polarity Change at Common_B Register (FRCB)</b>				
7-0	保留		0h	R
<b>REG [A0h] Interrupt Setup &amp; Status Register (INTR)</b>				
7	Key Scan 中断旗标 1: Key Scan 有侦测到按键输入 0: Key Scan 没有侦测到按键输入		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	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, 或致能BUSY。 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, 或致能BUSY。 0: 禁能INTY 中断		0h	R/W

<b>注:</b> Bit3~Bit0 的任一 Bit 被设为 "1" 将使得中断讯号功能 (INT) 和忙碌讯号功能(BUSY) 被致能 (Enable), 而 INT、BUSY 的触发准位由缓存器 [01h] 的 Bit4 决定。				
<b>REG [A1h] Key Scan Controller Register (KSCR)</b>				
7	Key Scan 的致能控制位 1: 致能 0: 禁能		0h	R/W
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
<b>注:</b> YT24064ZK 的 Key Scan 功能已屏蔽				
<b>REG [A2h] Key Scan Data Register (KSDR)</b>				
7-0	Key Scan KC[7~0] 的输出值		0h	RO
<b>REG [A3h] Key Scan Data Expand Register (KSER)</b>				
7-0	Key Scan KR[7~0] 的输入值		0h	RO
<b>REG [B0h] Interrupt Column Setup Register (INTX)</b>				
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。如不使用此功能请将此缓存器设成FFh。		27h	R/W
<b>REG [B1h] Interrupt Row Setup Register (INTY)</b>				
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。如不使用此功能请将此缓存器设成FFh。		EFh	R/W
<b>REG [C0h] Touch Panel Control Register (TPCR)</b>				
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	触控屏幕扫瞄 1: 禁能 0: 致能		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
<b>注:</b> YT24064ZK 的触控屏幕功能已屏蔽				
<b>REG [C1h] ADC Status Register (TPSR/ADCS)</b>				
7	ADC 数据转换完成指示 1: ADC 数据转换已完成		0h	R/W

	0: ADC 数据转换未完成			
6	触摸事件的侦测指示 1: 有被触摸 0: 没被触摸		0h	R/W
5	此位必须在系统使用时设为“1”		0h	R/W
4	此位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
<b>REG [C8h] Touch Panel Segment High Byte Data Register (TPXR)</b>				
7-0	储存触控屏幕行的高字节(bit9~2)数据		80h	RO
<b>REG [C9h] Touch Panel Common High Byte Data Register (TPYR)</b>				
7-0	储存触控屏幕列的高字节(bit9~2)数据		80h	RO
<b>REG [CAh] Touch Panel Segment/Common Low Byte Data Register (TPZR)</b>				
7-6	储存触控屏幕行的低字节(bit1~0)数据		0h	RO
5-4	保留		0h	R
3-2	储存触控屏幕列的低字节(bit1~0)数据		0h	RO
1-0	保留		0h	R
<b>REG [D0h] LCD Contrast Control Register (LCCR)</b>				
7	LCD 亮度控制(DAC 功能) 1: 禁能 0: 致能		1h	R/W
6-5	保留		0h	R
4-0	设定DAC 输出电流I <sub>out</sub> 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)		Fh	R/W
注: YT24064ZK 的 LCD 亮度软件控制功能已屏蔽				
<b>REG [E0h] Pattern Data Register (PNTR)</b>				
7-0	(1) 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’。 (2) Display Times of Gray Mode 在灰阶模式下(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。		0h	R/W
<b>REG [F0h] Font Control Register (FNCR)</b>				
7	字型ROM 的转换电路控制 1: 致能 0: Bypass 注: 用户建立字型ROM 时使用, 暂不提供此功能。		1h	R/W
6	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		0h	R/W
5-4	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)		1h	R/W
3	填写PNTR (REG [E0h]) 的数据到DDRAM 1: 开始写入	Graph	0h	R/W

	0: 未动作 当为 '1'，模块内部将自动读取PNTR 的Data，然后填写到 DDRAM 内(Range:[AWLR, AWTR] ~[AWRR, AWBR]), 之后此位会被自动清除为 '0'。			
2	强制为ASCII 解码 1: 所有输入的Data, 都视为ASCII 解码(00~FFh) 0: 模块会先检验输入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)	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		2h	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"。				
<b>REG [F1h] Font Size Control Register (FVHT)</b>				
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

## 5. 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	ç	ê	ë	ë	ï	î	ï	Ä	Å	
1001	È	æ	ø	ô	ö	õ	û	ü	ö	Ü	Φ	£	¥	ℳ	ƒ	
1010	ã	ĩ	õ	ũ	ñ	Ñ	º	º	¿	¿	¼	¼	½	½	»	»
1011	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒
1100	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒
1101	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒	▒
1110	α	β	Γ	π	Σ	σ	μ	τ	Φ	θ	Ω	δ	∞	∅	∈	π
1111	≡	±	≥	≤	∫	∫	÷	∞	°	°	√	n	²	■		

图 5-1: 内建 ASCII 区块 0

5.2 ASCII 字形区块1

b5-b0 b7-b4	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	€	‚	ƒ	„	…	†	‡	ˆ	%	Š	‹	œ	Ž			
0001		‘	’	“	”	•	–	—	™	š	›	œ	ž	ÿ		
0010		ı	φ	£	¥	€	ı	§	©	®	«	»	–	®	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		à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î
1100	Ŕ	Ŗ	Ÿ	Š	Œ	Č	Ć	Ĉ	Ċ	Ď	Đ	Ě	İ	İ	İ	İ
1101	ŕ	ŗ	ÿ	š	œ	č	ć	ĉ	ċ	ď	đ	ě	ı	ı	ı	ı
1110	ŕ	ŗ	ÿ	š	œ	č	ć	ĉ	ċ	ď	đ	ě	ı	ı	ı	ı
1111	đ	ň	ň	ő	ő	ő	÷	ř	ř	ů	ů	ů	ů	ý	t	ı

图 5-2: 内建 ASCII 区块 1

5.3 ASCII 字形区块2

b3-b0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
b7-b4																
0000																
0001																
0010		°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
0011		°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
0100	A	A	A	A	A	A	A	A	E	E	E	E	I	I	I	I
0101		N	O	O	O	G	O	x	G	U	U	U	U	U	S	B
0110	ä	ä	ä	ä	ä	ä	ä	ä	ë	ë	ë	ë	ï	ï	ï	ï
0111		n	o	o	o	g	o	÷	g	u	u	u	u	u	s	'
1000																
1001																
1010		A	K	R	x	ï	l	s	°	s	e	g	f	-	z	'
1011		°	°	°	°	°	°	°	°	°	°	°	°	°	°	°
1100	A	A	A	A	A	A	A	Æ	İ	Ç	È	È	È	È	İ	İ
1101	Đ	N	Ö	K	Ö	Ö	Ö	x	Ø	U	U	U	U	U	U	B
1110	ä	ä	ä	ä	ä	ä	ä	æ	ı	ç	è	è	è	è	ï	ï
1111	đ	n	ö	k	ö	ö	ö	÷	ø	u	u	u	u	u	ü	'

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

## 6. 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/SYS\_CLK=125ns，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125ns X 3 lock=375ns，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： $3+(Com \times Seg)/8$