

RAiO

RA8875

文字/圖形

TFT LCD 控制器

規格書

Version 2.1

June 15, 2016

RAiO Technology Inc.

©Copyright RAiO Technology Inc. 2011, 2012, 2013, 2014, 2015, 2016

改 版 說 明		
版 本	日 期	說 明
1.0	August 01, 2011	Preliminary Version
	October 04, 2011	1. Modify Figure 7-80 、 Figure 7-81
	November 18, 2011	1. Modify Section 7-4-2 2. Modify Table 8-2
	November 30, 2011	1. Add Figure 7-82 2. Modify Figure 6-41
1.1	January 10, 2012	1. Modify Section 7-6 BTE Function (Parallel MCU interface only)
1.2	February 09, 2012	1. Add Appendix A
	February 22, 2012	1. Modify REG[21h]
	March 16, 2012	1. Modify Section 4-8 : pin description of OSC_VDD 2. Modify Section 6-1-2 : Serial I/F Protocol 3. Add Note of REG[94h], REG[98h], Figure 7-29, Figure 7-30
1.3	May 15, 2012	1. Modify Section 5-2 : the Bit1 of REG[01h] 2. Modify Table 7-8
	March 16, 2013	1. Modify Section 4-1 : pin description of PS 2. Modify Section 5-2 : REG[01h] 、 REG[16h] 3. Add Section 6-1-2-3 : SPI Sleep/Wake Up
1.4	May 3, 2013	1. Modify Section 6-1-2-1 : 3-Wire SPI Interface 2. Modify Section 6-1-2-2 : 4-Wire SPI Interface
1.5	June 18, 2013	1. Modify Figure 6-12 、 Figure 6-18
	July 17, 2013	1. Modify Section 6-8 : Example of formula for system clock
	July 31, 2013	1. Modify Section 5-2 : REG[15h]
1.6	Demcember 03, 2013	1. Modify Figure 6-37~Figure 6-40 、 Figure 6-48
1.7	March 07, 2014	1. Modify Section 7-2-4 and Section 5-5 :REG[41h]
1.8	May 29, 2014	1. Add Note 3 of Table 8-2
	August 11, 2014	1. Modify Section 4-3: PDAT[15:0]
	October 16, 2014	1. Modify Table 7-23 & Table 7-24
1.9	December 11, 2014	1. Add REG[02h] description in BTE flow step for write/read data (7-6-4-1,7-6-4-2,7-6-4-5,7-6-4-9,7-6-4-10)
2.0	April 11, 2016	1. Add Figure 6-27 、 Figure 6-29 、 Figure 6-31
2.1	June 15, 2016	1. Modify Section 7-6 : BTE Function

Chapter	Contents	Page
1.	簡介	7
2.	特性	7
3.	方塊圖	8
3-1	內部方塊圖	8
3-2	系統方塊圖	8
4.	腳位定義	9
4-1	MCU 介面	9
4-2	串列式MCU介面	10
4-3	LCD面板介面	11
4-4	串列式Flash/ROM 介面	11
4-5	觸控介面	12
4-6	鍵盤介面	12
4-7	脈寬調變介面	12
4-8	時脈與電源介面	13
5.	暫存器	14
5-1	狀態暫存器	15
5-2	系統與組態暫存器	16
5-3	LCD 顯示模式	21
5-4	工作視窗及捲動視窗設定	27
5-5	游標設定	30
5-6	BTE 引擎	34
5-7	觸控面板	41
5-8	圖形游標	43
5-9	PLL設定	44
5-10	脈波寬度調變 (PWM)	46
5-11	繪圖控制暫存器	49
5-12	直接記憶體存取 (DMA) 暫存器	55
5-13	鍵盤掃描與 IO控制暫存器	58
5-14	浮動視窗控制暫存器	60
5-15	串列式Flash控制暫存器	62
5-16	中斷控制	63
6.	硬體介面	66
6-1	MCU 介面	66
6-1-1	MCU傳輸協定	67
6-1-1-1	並列式介面的傳輸協定	67

6-1-2	串列式介面的協定.....	70
6-1-2-1	3-Wire SPI 介面	70
6-1-2-2	4-Wire SPI 介面	73
6-1-2-3	SPI 睡眠 / 喚醒.....	77
6-1-2-4	IIC 介面	78
6-1-3	狀態讀取暫存器	80
6-1-4	指令寫入暫存器	81
6-1-5	記憶體讀取/寫入的操作	83
6-1-6	中斷與等待.....	85
6-1-6-1	中斷.....	85
6-1-6-2	等待.....	86
6-1-7	資料匯流排與TFT的RGB資料格式.....	87
6-1-7-1	16 位元資料匯流排.....	87
6-1-7-2	8 位元資料匯流排.....	87
6-2	驅動介面的色彩設定模式.....	88
6-3	LCD 介面.....	89
6-4	外部串列式Flash/ROM.....	91
6-4-1	外部串列式字體記憶體	94
6-4-2	外部串列式資料記憶體	95
6-5	觸控介面	96
6-6	鍵盤掃描功能.....	98
6-7	脈寬調變介面.....	98
6-8	時脈 (Clock) 與 PLL.....	99
6-9	重置功能	101
6-10	電源	103
6-10-1	電源腳位說明	103
6-10-2	電源結構	103
7.	功能描述.....	104
7-1	捲動功能	104
7-1-1	捲動視窗與捲動偏移值	104
7-1-2	水平捲動與垂直捲動.....	104
7-1-3	圖層混合捲動.....	105
7-1-3-1	圖層 1 與圖層 2 同時捲動.....	106
7-1-3-2	只捲動圖層 1.....	106
7-1-3-3	只捲動圖層 2.....	107
7-1-3-4	緩衝捲動	108
7-2	工作視窗.....	109
7-2-1	工作視窗的文字寫入.....	109
7-2-2	工作視窗的幾何圖形顯示.....	110
7-2-3	工作視窗中的DMA功能顯示.....	110
7-2-4	工作視窗的記憶體讀寫	110
7-3	游標與圖形樣板	111
7-3-1	游標種類	111
7-3-1-1	圖形游標	111

7-3-1-2	記憶體讀取游標.....	113
7-3-1-3	記憶體寫入游標.....	113
7-3-1-4	文字寫入游標.....	114
7-3-2	游標屬性.....	114
7-3-2-1	游標閃爍.....	114
7-3-2-2	游標的高度與寬度.....	115
7-3-3	圖形樣板.....	117
7-4	文字.....	118
7-4-1	內部文字記憶體.....	118
7-4-2	外部 Font ROM.....	123
7-4-3	CGRAM (Character Generation RAM 自建字庫功能).....	124
7-4-4	文字 90 度轉向.....	126
7-4-5	文字放大與通透功能.....	126
7-4-6	文字換行.....	127
7-4-7	文字全型對齊.....	127
7-5	幾何圖案繪圖引擎.....	128
7-5-1	圓形輸入.....	128
7-5-2	橢圓輸入.....	129
7-5-3	曲線輸入.....	130
7-5-4	方形輸入.....	131
7-5-5	直線輸入.....	132
7-5-6	三角形輸入.....	133
7-5-7	圓角方形輸入.....	134
7-6	BTE 引擎功能.....	135
7-6-1	選擇BTE起始點位置及圖層.....	138
7-6-2	BTE 操作說明.....	138
7-6-2-1	BTE寫入.....	138
7-6-2-2	BTE讀取.....	138
7-6-2-3	BTE移動.....	138
7-6-2-4	單色填滿.....	138
7-6-2-5	圖案填滿.....	138
7-6-2-6	BTE通透填滿.....	138
7-6-2-7	BTE通透寫入.....	139
7-6-2-8	BTE通透移動.....	139
7-6-2-9	顏色擴充.....	139
7-6-2-10	顏色移動.....	139
7-6-3	BTE記憶體存取方式.....	140
7-6-3-1	區塊記憶體讀取.....	140
7-6-3-2	線性記憶體讀取.....	140
7-6-4	BTE 功能說明.....	141
7-6-4-1	BTE寫入搭配光柵運算功能.....	141
7-6-4-2	BTE讀取功能.....	143
7-6-4-3	BTE正向移動搭配光柵運算功能.....	144
7-6-4-4	BTE反向移動搭配光柵運算功能.....	146
7-6-4-5	BTE通透性寫入功能.....	148
7-6-4-6	BTE正向通透性移動功能.....	150
7-6-4-7	圖形樣板填入搭配光柵運算功能.....	151

7-6-4-8	通透性圖形樣板填入功能	153
7-6-4-9	顏色擴充功能	155
7-6-4-10	通透性顏色擴充功能	159
7-6-4-11	BTE移動搭配顏色擴充功能	161
7-6-4-12	通透性BTE移動功能搭配顏色擴充	163
7-6-4-13	單色填滿功能	164
7-7	圖層混合功能	165
7-7-1	顯示圖層一	166
7-7-2	顯示圖層二	167
7-7-3	漸入漸出模式	168
7-7-4	通透模式	169
7-7-5	布林運算OR	169
7-7-6	布林運算AND	169
7-7-7	浮動視窗	170
7-8	觸控面板功能	171
7-8-1	觸控面板操作模式	172
7-8-1-1	自動模式	172
7-8-1-2	手動模式	173
7-8-2	觸控事件偵測模式	174
7-8-2-1	外部中斷模式	174
7-8-2-2	軟體輪詢模式	174
7-8-3	觸控掃描與取樣時間	175
7-9	鍵盤	176
7-10	記憶體直接存取功能	180
7-10-1	連續記憶體直接存取模式	180
使用方法 :	180
7-10-2	區塊資料記憶體直接存取模式	182
7-11	脈寬調變	183
7-12	睡眠模式	185
8. AC/DC 特性		187
8-1	最大範圍限制	187
8-2	DC 電氣特性	188
9. 產品封裝		189
9-1	封裝腳位圖	189
9-2	封裝尺寸	190
9-3	產品編號	190
Appendix A. RA8875 支援的集通字型列表		191

1. 簡介

RA8875 是一個文字與繪圖模式的雙圖層液晶顯示 (TFT-LCD) 控制器，可結合文字或2D圖形應用，最大可支援到800*480 點解析度的中小尺寸數位面板。內建 768KB 顯示記憶體可提供大多數使用者的應用一個更彈性的解決方案。此外，使用者可藉由選用外部串列式Flash 介面，選用外接集通公司的字型ROM，支援BIG5/GB/UNICODE 等編碼，可提供最大達32*32 像素之字型輸入。在圖形的使用上，RA8875 支援2D 的BTE 引擎 (Block Transfer Engine)，此功能相容於一般通用的2D BitBLT 功能，可處理大量圖形資料轉換與傳送。同時RA8875 也內建幾何圖形加速引擎 (Geometric Speed-up Engine)，提供使用者透過簡單的設定輕鬆畫出直線、矩形、圓形和橢圓的幾何圖形。為了貼近終端始用者的應用，RA8875 整合了強大的功能，如畫面捲動功能、顯示浮動視窗、圖形Pattern 及文字放大等功能，可大量節省使用者軟體開發的時間，並提昇MCU軟體的執行效率。

RA8875 提供低成本的 8080/6800 並列式MCU 介面，由於內建強大的硬體加速功能，可降低資料傳輸所需的時間並且改善效率。RA8875提供串列式 SPI/I2C 等極少量腳位的介面，內建 4-wire 的觸控面板控制器，以及2組脈波寬度調變 (PWM)，可用於調整面板背光或其他應用。相對於其他的解決方案，RA8875 是一個功能強大及低成本的彩色TFT 控制器，讓使用者順利解決軟硬體開發上的疑慮，同時達成低成本、高效能的系統方案。

2. 特性

- ◆ 支援文字/繪圖兩種混合顯示模式
- ◆ 內建 DDRAM 記憶體容量：768KB
- ◆ 色彩深度：256/65K 色
- ◆ 支援 8/16 bpp Generic RGB TFT 介面
- ◆ 支援 TFT 面板大小：
 - 800x480 Pixels 2 Layers @ 256 Colors.
 - 800x480 Pixels 1 Layer @ 64K Colors.
 - 480x272 Pixels 2 Layers @ 64K Colors.
- ◆ 支援 MCU 介面：
 - 8-bit 或 16-bit 的 8080/6800 系列資料匯流排 介面
 - I2C or 3 或 4-線式 SPI 介面
- ◆ 支援強大的水平和垂直區塊捲動
- ◆ 內建 10KB 字型 ROM (8x16 Dots) 及支援標準 ISO/IEC 8859-1/2/3/4 編碼
- ◆ 支援外部串列式 Flash/ROM SPI 介面
- ◆ 支援集通公司 UNICODE/BIG5/GB 系列字庫 (字型大小:16x16/24x24/32X32 Dots)
- ◆ 支援 1 倍到 4 倍字型放大 (垂直和水平)
- ◆ 支援文字垂直旋轉模式功能
- ◆ 內建 2D Block Transfer Engine (BTE) 功能，相容於 2D BitBLT 功能
- ◆ 內建幾何圖形加速繪圖引擎
- ◆ 提供可調整大小的文字寫入游標功能
- ◆ 提供 32x32 像素 的圖形游標功能
- ◆ 支援 256 個使用者自訂 8x16 字元符號
- ◆ 支援 16 個使用者自建 8x8 像素圖形 Pattern，或 4 個使用者自建 16x16 像素圖形 Pattern
- ◆ 內建 2 組脈波寬度調變(PWM)，提供 LCD 背光 調節或其他用途
- ◆ 內建 4 線式觸控面板控制器
- ◆ 提供低耗電的睡眠模式
- ◆ 內建智慧型 4x5 鍵盤控制器.
- ◆ 提供 4 個 GPO 及固定的 GPOX
- ◆ 提供 5 個 GPI 及固定的 GPIX
- ◆ 時脈來源：內建晶體振盪電路配合程式化的 PLL
- ◆ 電源操作範圍：3.0V~3.6V
- ◆ 封裝：LQFP-100pin

3. 方塊圖

3-1 內部方塊圖

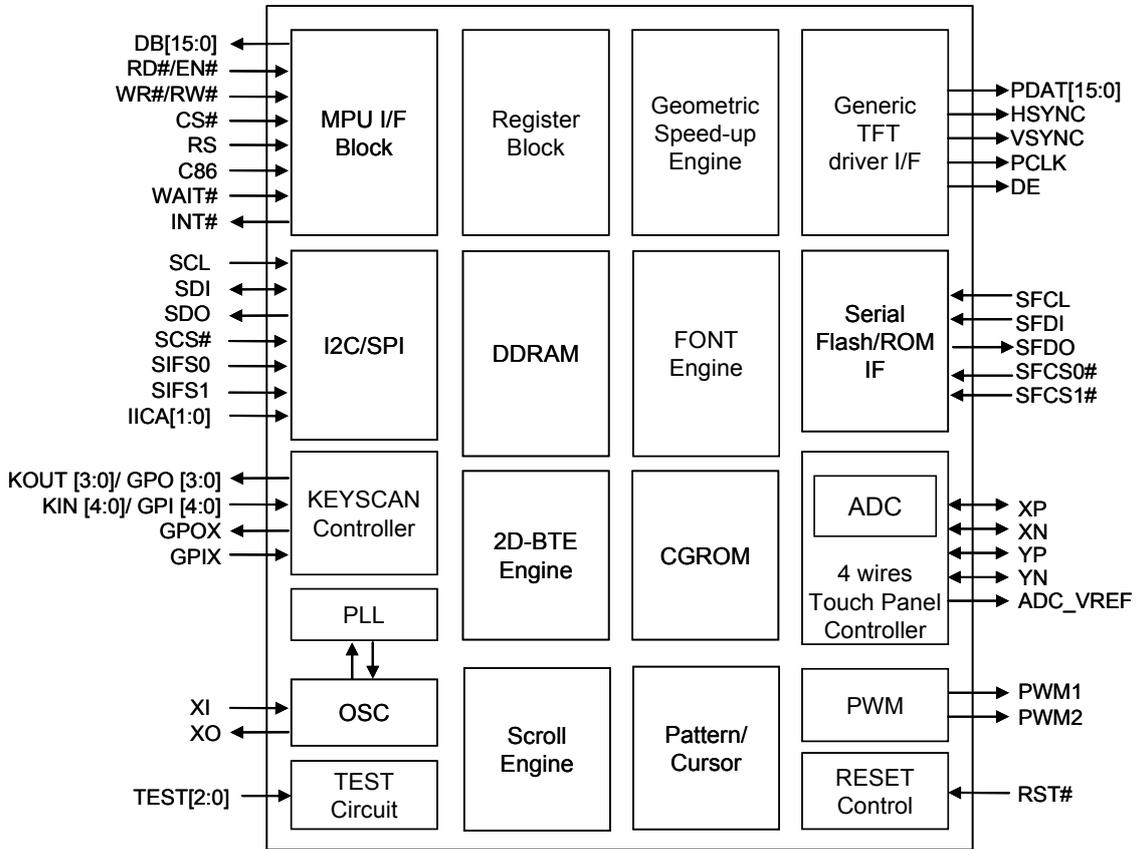


圖 3-1 : RA8875 內部方塊圖

3-2 系統方塊圖

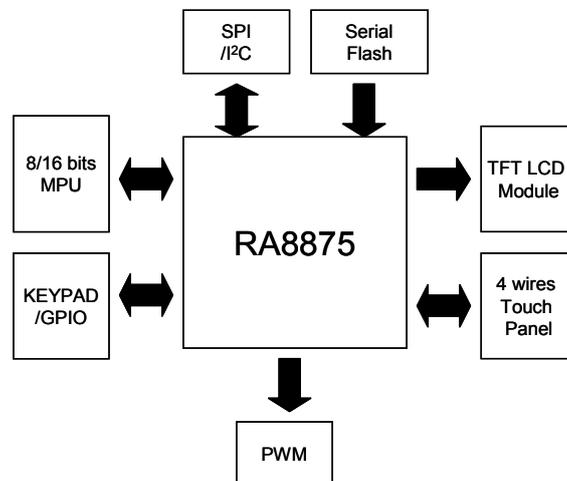


圖 3-2 : RA8875 系統方塊圖

4. 腳位定義

4-1 MCU 介面

接腳名稱	I/O	腳位說明															
DB[15:0]	I/O	<p>資料匯流排 (Data Bus) 負責 RA8875 及微處理器 (MCU) 之間的資料傳送與接收。 當設定暫存器號碼及資料時，會使用到 DB[7:0]。 當寫入資料到顯示記憶體時，依據資料匯流排模式的設定會使用 DB[15:0]。 當使用8-bit資料匯流排模式下，DB[15:8] 應保持接下拉/上拉電阻至GND或VDD。</p>															
RD# (EN)	I	<p>致能 / 讀取控制訊號 當 MCU 為 8080 系列，此腳位為資料讀取訊號 (RD#)，於低電位動作。 當 MCU 為 6800 系列，此腳位為致能訊號 (EN)，於高電位動作。</p>															
WR# (RW#)	I	<p>寫入 / 讀-寫控制訊號 (Write/Read-Write) 當 MCU 為 8080 系列，此腳位為資料寫入訊號 (WR#)，於低電位動作。 當 MCU 為 6800 系列時，此腳位為資料讀取 / 寫入訊號 (RW#)，於高電位時表示讀取動作，於低電位時表示寫入動作。</p>															
CS#	I	<p>晶片選取控制訊號 低準位動作的晶片選取輸入信號。</p>															
RS	I	<p>指令 / 資料選擇控制訊號 此腳位用於區分進行指令週期或資料週期。當 RS = 0 時，RA8875 為資料讀取或寫入週期。當 RS = 1 時，RA8875 為狀態讀取或指令寫入週期，當於 8080 介面時，通常此腳位和微處理器的 A0 相接。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>RS</th> <th>WR#</th> <th>存取週期</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>資料寫入 (Data Write)</td> </tr> <tr> <td>0</td> <td>1</td> <td>資料讀取 (Data Read)</td> </tr> <tr> <td>1</td> <td>0</td> <td>指令寫入 (CMD Write)</td> </tr> <tr> <td>1</td> <td>1</td> <td>狀態讀取 (Status Read)</td> </tr> </tbody> </table>	RS	WR#	存取週期	0	0	資料寫入 (Data Write)	0	1	資料讀取 (Data Read)	1	0	指令寫入 (CMD Write)	1	1	狀態讀取 (Status Read)
RS	WR#	存取週期															
0	0	資料寫入 (Data Write)															
0	1	資料讀取 (Data Read)															
1	0	指令寫入 (CMD Write)															
1	1	狀態讀取 (Status Read)															
C86	I	<p>MCU 介面選擇 0 : Intel 8080 系列 MCU 介面。 1 : Motorola 6800 系列 MCU 介面。</p>															
PS	I	<p>並列式/串列式 I/F 選取輸入訊號 0 : 選取並列式 8080/6800 介面。 1 : 選取串列式 3/4-wire SPI 或 IIC 介面。 當選擇 MCU 介面動作時，必須在指令/資料週期確認前，正確設定輸入 PS 腳位訊號，當使用串列式介面時，建議將 DB[15:0], RD#(EN) , WR#(RW#) , CS# , RS, C86 腳位接到 VDD。</p>															

接腳名稱	I/O	腳位說明
INT#	O	中斷訊號 RA8875 發出給 MCU 的外部中斷信號。
WAIT#	O	等待訊號 (Wait Signal Output) 此為反應 RA8875 內部的執行使用狀況。當 WAIT# 為 Low 時表示 RA8875 正處於忙碌狀態，無法接受來自 MCU 指令。把此腳位接到 MCU I/O 腳位時，它能用輪詢機制 (Polling) 來監控內部的狀況。

4-2 串列式MCU介面

接腳名稱	I/O	腳位說明
SCL	I	SPI 時脈 包含串列式 3-wire、4-wire 或 IIC 介面時脈。 若不使用時，請將接腳接至 VDDP。
SDI	I/O	IIC 資料/4-wire SPI 資料輸入 4-wire SPI 介面：當使用串列介面時，為資料輸入信號。 3-wire SPI 介面：不使用，請接至 VDDP。 IIC 介面：當使用串列介面時，為資料輸入/資料輸出信號。 當不使用時，請將接腳接至 VDDP。
SDO	I/O	3-wire SPI 資料/4-wire SPI 資料輸出 4-wire SPI 介面：當使用串列介面時，為資料輸出信號。 3-wire SPI 介面：當使串列介面時，為資料輸入/資料輸出。 IIC 介面：不使用，此接腳請保持浮接。 當不使用時，請保持浮接。
SCS#	I	SPI 晶片選擇 當選擇接腳為 3-wire 或 4-wire 串列介面。 IIC 介面：當不使用時，請將接腳接至 VDDP。 當不使用時，請將接腳接至 VDDP。
IICA[1:0]	I	IIC I/F: IIC 位置選擇 其他介面：當不使用時，請勿保持浮接。
SIFS[1:0]	I	串列介面選擇 00：不使用。 01：3-Wire SPI。 10：4-Wire SPI。 11：IIC。 若串列介面不使用時，請將 SIFS1 and SIFS0 接腳都連接至 GND。

4-3 LCD面板介面

接腳名稱	I/O	腳位說明												
PDAT[15:0]	○	LCD 面板資料匯流排 (LCD Panel Data Bus) 資料匯流排的輸出端連接到 Source Driver。RA8875 暫存器 (REG[10h]) 支援 256/64K 色彩深度，使用者在做不同的設定時可連接對應的 RGB 匯流排。不使用的接腳請保持浮接。 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>色彩深度</th> <th>紅</th> <th>綠</th> <th>藍</th> </tr> </thead> <tbody> <tr> <td>256</td> <td>PDAT[15:13]</td> <td>PDAT[10:8]</td> <td>PDAT[4:3]</td> </tr> <tr> <td>64K</td> <td>PDAT[15:11]</td> <td>PDAT[10:5]</td> <td>PDAT[4:0]</td> </tr> </tbody> </table>	色彩深度	紅	綠	藍	256	PDAT[15:13]	PDAT[10:8]	PDAT[4:3]	64K	PDAT[15:11]	PDAT[10:5]	PDAT[4:0]
		色彩深度	紅	綠	藍									
256	PDAT[15:13]	PDAT[10:8]	PDAT[4:3]											
64K	PDAT[15:11]	PDAT[10:5]	PDAT[4:0]											
HSYNC	○	HSYNC Pulse Generic TFT 介面信號。												
VSYNC	○	VSYNC Pulse Generic TFT 介面信號。												
PCLK	○	Pixel Clock Generic TFT 介面信號。												
DE	○	Data Enable Generic TFT 介面信號。												

4-4 串列式Flash/ROM 介面

接腳名稱	I/O	腳位說明
SFCL	○	外部串列 Flash/ROM 時脈 串列 Flash/ROM SPI 介面時脈。
SFDI/SIO0	I/O	外部 Flash/ROM SPI 資料輸入 單一模式(Singal mode)：串列 Flash/ROM SPI I/F 資料輸入。在 RA8875 其初始狀態為輸出。 雙倍模式(Dual mode)：此信號被用在雙向輸入/輸出資料 #0(SIO0)。
SFDO/SIO1	I/O	外部 Flash/ROM SPI 資料輸出 單一模式：串列 Flash/ROM SPI 介面資料輸出信號。對於 RA8875 來說，此信號為輸入(預設值)。 雙倍模式：此信號被用在雙向輸入/輸出資料#1(SIO1)。
SFCS0#	○	外部 Flash/ROM SPI 晶片致能 0 串列 Flash/ROM SPI 介面 #0.的晶片致能腳位。
SFCS1#	○	外部 Flash/ROM SPI 晶片致能 1 串列 Flash/ROM SPI 介面 #1.的晶片致能腳位。

4-5 觸控介面

接腳名稱	I/O	腳位說明
YN	A	觸控面板控制訊號 (YN Signal for Touch Panel) 4-wire 觸控面板控制訊號。
YP	A	觸控面板控制訊號 (YP Signal for Touch Panel) 4-wire 觸控面板控制訊號。
XN	A	觸控面板控制訊號 (XN Signal for Touch Panel) 4-wire 觸控面板控制訊號。
XP	A	觸控面板控制訊號 (XP Signal for Touch Panel) 4-wire 觸控面板控制訊號。
ADC_VREF	A	觸控面板控制器的 ADC 參考電壓 (TP ADC Reference Voltage) 此腳位為 ADC 參考電壓。其準位等於 $0.5 \cdot V_{DD}$ ，此參考電壓可以選擇內部產生，或是外部輸入。

4-6 鍵盤介面

接腳名稱	I/O	腳位說明
KOUT[3:0]/ (GPO[3:0])	O	鍵盤介面的掃描信號/ GPOs (一般性輸出信號) 此為鍵盤介面的掃描線輸出，其輸出架構為 Open-drain (初始狀態)。 此信號可透過暫存器設定為 GPOs。若不使用時，請保持浮接。
KIN[4:0]/ (GPI[4:0])	I	鍵盤介面的資料信號/ GPIs (一般性輸入信號) 此為鍵盤資料輸入(預設)，請外加提升電阻。 KIN[4:0]可透過暫存器設定為 GPI[4:0]。若不使用時，請接到 GND。
GPOX	O	Extra GPO (General Purpose Output) 此為額外的 GPO 信號，若不使用時，請保持浮接。
GPIX	I	Extra GPI (General Purpose Input) 此為額外的 GPI 信號，若不使用時，請接到 GND。

4-7 脈寬調變介面

接腳名稱	I/O	腳位說明
PWM1	O	此為 PWM 信號輸出#1。
PWM2	O	此為 PWM 信號輸出#2。

4-8 時脈與電源介面

接腳名稱	I/O	腳位說明
XI	I	Crystal 輸入腳位 (Crystal Input Pin) 此為內部晶體振盪電路的輸入端，用來連接外部晶體振盪器，以產生 RA8875 的內部時脈 (Clock) 信號。
XO	O	Crystal 輸出腳位 (Crystal Output Pin) 此為內部晶體振盪電路的輸出端，用來連接外部晶體振盪器，以產生 RA8875 的內部時脈 (Clock) 信號。
RST#	I	重置訊號 (Reset Signal Input) 此腳位為 RA8875 低電位動作之硬體重置輸入訊號。為了提升抗雜訊的能力，此腳位為 Schmitt-Trigger 輸入。當輸入準位變低時，能確保此腳位不會被觸發。
TEST[2:0]	I	測試模式輸入訊號 (Test Mode Input) 此腳位為測試訊號，必須保持接地 (GND)。
VDDP	P	RA8875 I/O 的 VDD 電源輸入 (IO VDD) 此腳位為 3.3V 輸入。
CORE_VDD	P	RA8875 內部核心電路的 VDD 電源 (CORE VDD) 此腳位為 1.8V 核心電路電源輸入。
LDO_OUT	P	LDO VDD 輸出 (LDO VDD Output) 1.8V 的 LDO 電源輸出，必須於輸出端接旁路電容以避免電源的噪訊。
LDO_GND	P	LDO GND 此為 LDO 電路的接地端。
OSC_VDDP	P	OSC IO VDD 此為錯開的 OSC 3.3V IO 電源。
OSC_VDD	P	OSC VDD 此為 OSC 1.8V 電源輸出，供內部振盪器電路使用。建議在靠近信號端接旁路電容。
OSC_GNDP	P	OSC IO GND 此為隔開的 OSC IO 端接地信號。
OSC_GND	P	OSC GND OSC 的接地信號，此接腳在 IC 內部和 OSC_GNDP 連結。
ADC_VDD	P	ADC VDD ADC 3.3V 電源信號。
ADC_GND	P	ADC GND ADC 接地信號。
GND	P	GND IO 信號及核心電路接地端。

5. 暫存器

RA8875 的MCU 介面有 4 種週期 (Cycle) 類型，請參考表 5-1。暫存器的設定或讀取功能是由這些週期所組成的。RA8875 包括一個狀態暫存器及數十個指令暫存器。狀態暫存器是一個唯讀的暫存器，只能透過「狀態讀取」週期讀取。指令暫存器可用於存取大部分的功能，可透過指令寫入週期及資料寫入週期進行存取。「指令寫入」週期設定暫存器的號碼，而「資料寫入」週期則設定暫存器的寫入資料。當讀取特定的指令暫存器時，MCU 需要先下「指令寫入」週期然後再下「資料讀取」週期。「指令寫入」週期對程式設定暫存器數量，而「資料讀取」週期讀取暫存器的資料。指令暫存器分為 15 個類別，請參考表 5-2，且大部分都可讀或寫。下面章節將對所有暫存器的內容進行說明。

表 5-1 : MCU 週期類型

週期類型	RW#	RS	說明
指令寫入	0	1	暫存器號碼寫入週期
狀態讀取	1	1	狀態讀取週期
資料寫入	0	0	對應的暫存器資料/記憶體資料寫入週期，跟隨著指令寫入週期
資料讀取	1	0	對應的暫存器資料/記憶體資料讀取週期，跟隨著指令寫入週期

表 5-2 : 指令暫存器類別

No.	指令暫存器類別	暫存器位址
1	系統與組態暫存器	[01h], [02h], [04h], [10h] ~ [1Fh]
2	LCD 顯示控制暫存器	[20h] ~ [29h]
3	工作視窗設定暫存器	[30h] ~ [3Fh]
4	游標設定暫存器	[40h] ~ [4Eh]
5	BTE 引擎控制暫存器	[50h] ~ [67h]
6	觸控面板設定暫存器	[70h] ~ [74h]
7	圖形游標設定暫存器	[80h] ~ [85h]
8	PLL 設定暫存器	[88h], [89h]
9	脈波寬度調變設定控制暫存器	[8Ah] ~ [8Eh]
10	繪圖控制暫存器	[90h] ~ [ACh]
11	DMA 控制暫存器	[B0h] ~ [BFh]
12	KEY & IO 控制暫存器	[C0h] ~ [C7h]
13	浮動視窗控制暫存器	[D0h] ~ [DBh]
14	串列 Flash 控制暫存器	[E0h] ~ [E2h]
15	中斷控制暫存器	[F0h] ~ [F1h]

暫存器功能說明如下，每個暫存器包含 8 bits 資料，暫存器的名稱、編號、初始值及存取屬性皆列在每個功能表中。

註：RO：代表此暫存器是唯讀 (Read only)。

WO：代表此暫存器為唯寫 (Write only)。

RW：代表此暫存器可以讀/寫 (Read-able and Write-able)。

5-1 狀態暫存器

狀態暫存器 Status Register (STSR)

Bit	說明	初始值	Access
7	記憶體讀取/寫入忙碌 (包含字體寫入忙碌) 0：閒置狀態 (No Memory Read/Write event)。 1：忙碌狀態 (Memory Read/Write busy)。	0	RO
6	BTE 忙碌 (BTE Busy) 0：BTE 處於非忙碌狀態。 1：BTE 處於忙碌狀態。	0	RO
5	觸控掃描偵測 (Touch Panel Event Detected) 0：觸控掃描沒有偵測到輸入訊號。 1：觸控面板偵測到輸入訊號 此位元直接來自觸控控制器 ADET 信號且無消除彈跳處理。建議利用輪詢模式時需多次確認觸控事件，以保其正確性。	0	RO
4	睡眠模式狀態 0：RA8875 處於工作模式。 1：RA8875 處於睡眠模式。	0	RO
3-1	N/A	0	RO
0	Serial Flash/ROM 忙碌 (Serial Flash/ROM Busy) Serial Flash/ROM 介面處於忙碌狀態。 0：閒置 1：忙碌	0	RO

5-2 系統與組態暫存器

REG[01h] Power and Display Control Register (PWRR)

Bit	說明	初始值	Access
7	LCD 顯示關閉訊號 (LCD Display Off) 0: LCD 畫面關閉。 1: LCD 畫面顯示。	0	RW
6-2	NA	0	RO
1	睡眠模式 0: 正常模式。 1: 睡眠模式。 註： 1. 睡眠模式可以由：觸控事件、鍵盤輸入、軟體程式三種方法來喚醒。 2. 當使用 IIC 介面時，不支援此功能。 3. 當使用 SPI 介面時，請參考 6-1-2-3 節的使用步驟。	0	RW
0	軟體重置 0: 不動作。 1: 軟體重置。 註：此位元必須先設定為 1，然後再設定為 0 後，才能完成正確的軟體重置動作。	0	WO

REG[02h] Memory Read/Write Command (MRWC)

Bit	說明	初始值	Access
7-0	寫入功能：Memory 寫入 Data 資料寫入記憶體對應到 MWCR1[3:2] 的設定。可利用連續性的資料讀取週期來進行大量的資料寫入。 讀取功能：Memory 讀取 Data 從記憶體讀取資料對應到 MWCR1[3:2] 的設定。可利用連續性的資料讀取週期來進行大量的資料讀取。第一筆資料讀取週期為多餘週期 (Dummy Read)，請忽略。	--	RW

REG[04h] Pixel Clock Setting Register (PCSR)

Bit	說明	初始值	Access
7	PCLK Inversion 0: PDAT 是在 PCLK 正緣上升 (Rising Edge) 時被取樣。 1: PDAT 是在 PCLK 負緣下降 (Falling Edge) 時被取樣。	0	RW
6-2	NA	0	RO
1-0	PCLK 時脈週期設定 Pixel Clock (PCLK) 時脈週期設定。 00b: PCLK 時脈週期= 系統時脈週期。 01b: PCLK 時脈週期= 2 倍的系統時脈週期。 10b: PCLK 時脈週期= 4 倍的系統時脈週期。 11b: PCLK 時脈週期= 8 倍的系統時脈週期。	0	RW

REG[05h] Serial Flash/ROM Configuration Register (SROC)

Bit	說明	初始值	Access
7	Serial Flash/ROM I/F # 選擇 0: 選擇 Serial Flash/ROM 0 介面。 1: 選擇 Serial Flash/ROM 1 介面。	0	RW
6	Serial Flash/ROM 定址模式 0: 24 位元定址模式。 此位元必須設為 0。	0	RW
5	Serial Flash/ROM 波形模式 0: 波形模式 0。 1: 波形模式 3。	0	RW
4-3	Serial Flash /ROM 讀取週期 (Read Cycle) 00b: 4 bus → 無空週期 (No Dummy Cycle)。 01b: 5 bus → 1 byte 空週期。 1Xb: 6 bus → 2 byte 空週期。	0	RW
2	Serial Flash /ROM 存取模式 (Access Mode) 0: 字型模式。 1: DMA 模式。	0	RW
1-0	Serial Flash /ROM I/F Data Latch 選擇模式 0Xb: 單一模式。 10b: 雙倍模式 0。 11b: 雙倍模式 1。	0	RW

REG[06h] Serial Flash/ROM CLK Setting Register(SFCLR)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	<p>Serial Flash/ROM 時脈頻率設定 0xb: SFCL 頻率 = 系統時脈頻率 (當 DMA 為致能狀態，並且色彩深度為 256 色，則 SFCL 頻率固定為=系統時脈頻率/ 2) 10b: SFCL 頻率 = 系統時脈頻率/ 2 11b: SFCL 頻率 = 系統時脈頻率/ 4</p>	0	RW

REG[10h] System Configuration Register (SYSR)

Bit	說明	初始值	Access
7-4	N/A	0	RO
3-2	<p>色彩深度設定 (Color Depth Setting) 00b : 8-bpp 的通用 TFT 介面， i.e. 256 色。 1xb : 16-bpp 的通用 TFT 介面， i.e. 65K 色。</p>	0	RW
1-0	<p>MCUIF 選擇 00b : 8-位元 MCU 介面。 1xb : 16-位元 MCU 介面。</p>	0	RW

REG[12h] GPI

Bit	說明	初始值	Access
7-5	NA	0	RO
4-0	<p>GPI[4:0] : 通用型輸入 (General Purpose Input) KEY_EN = 0: 通用型輸入的資料暫存區， 資料讀自信號 KIN[4:0]。</p>	NA	RO

Note : KEY_EN : REG[C0h] bit 7

REG[13h] GPO

Bit	說明	初始值	Access
7-4	NA	0	RO
3-0	<p>GPO[3:0] : 通用型輸出 (General Purpose Output) KEY_EN = 0: 通用型輸出的資料來源， 輸出到 KOUT[3:0]。 KEY_EN = 1: 無作用。</p>	0	RW

Note : KEY_EN : REG[C0h] bit 7

REG[14h] LCD Horizontal Display Width Register (HDWR)

Bit	說明	初始值	Access
7	NA	0	RO
6-0	水平顯示區域寬度設定位元[6:0] 此暫存器規範液晶面板水平顯示寬度，每單位 8-像素解析度。 水平顯示寬度(像素) = (HDWR + 1)*8	0	RW

Note : HDWR 設定必須小於 64h，因為最小水平顯示寬度為 800 像素。

REG[15h] Horizontal Non-Display Period Fine Tuning Option Register (HNDFTR)

Bit	說明	初始值	Access
7	DE 信號的極性 0 : High 動作。 1 : Low 動作。	0	RW
6-4	NA	0	RO
3-0	水平非顯示期間微調寬度設定位元 [3:0] (HNDFT) 這個暫存器規範液晶面板水平非顯示微調寬度 (支援 SYNC mode 面板)，每單位為 2-像素解析度。	0	RW

REG[16h] LCD Horizontal Non-Display Period Register (HNDR)

Bit	說明	初始值	Access
7-5	NA	0	RO
4-0	水平非顯示期間寬度設定位元 [4:0] (HNDRP) 這個暫存器規範液晶面板水平非顯示寬度。 水平非顯示寬度 (像素) = (HNDR + 1)x8+(HNDFTR/2+1)x2 + 2	0	RW

REG[17h] HSYNC Start Position Register (HSTR)

Bit	說明	初始值	Access
7-5	NA	0	RO
4-0	水平同步信號 (HSYNC) 起始位址寬度設定位元[4:0] 這個暫存器規範顯示區域結束到水平同步信號起始位址的寬度， 每一階的調變單位為 8-像素解析度。 水平同步信號起始位址寬度 (像素) = (HSTR + 1)x8	0	RW

REG[18h] HSYNC Pulse Width Register (HPWR)

Bit	說明	初始值	Access
7	HSYNC 動作準位 0 : Low 動作。 1 : High 動作。	0	RW
6-5	NA	0	RO
4-0	水平同步信號 (HSYNC) 脈波寬度設定位元[4:0] 水平同步信號脈波寬度 (像素) = (HPW + 1)x8	0	RW

REG[19h] LCD Vertical Display Height Register (VDHR0)

Bit	說明	初始值	Access
7-0	垂直顯示區域高度設定位元 [7:0] 垂直顯示區域高度 (Line) = VDHR + 1	0	RW

REG[1Ah] LCD Vertical Display Height Register0 (VDHR1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	垂直顯示區域高度設定位元 [8] 垂直顯示設定高度 (Line) = VDHR + 1	0	RW

Note : VDHR 設定必須小於 1E0h，因為最大的垂直顯示高度為 480。

REG[1Bh] LCD Vertical Non-Display Period Register (VNDR0)

Bit	說明	初始值	Access
7-0	垂直非顯示期間設定位元 [7:0] 垂直非顯示期間 (Line) = (VNDR + 1)	0	RW

REG[1Ch] LCD Vertical Non-Display Period Register (VNDR1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	垂直非顯示區域期間設定位元 [8] 垂直非顯示區域期間 (Line) = (VNDR + 1)	0	RW

REG[1Dh] VSYNC Start Position Register (VSTR0)

Bit	說明	初始值	Access
7-0	垂直同步信號 (VSYNC) 起始位址高度設定位元 [7:0] 此暫存器規範垂直顯示區域結束到垂直同步信號起始位置。 垂直同步信號起始位置(Line) = (VSTR + 1)	0	RW

REG[1Eh] VSYNC Start Position Register (VSTR1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	垂直同步信號 (VSYNC) 起始位址設定位元 [8] 此暫存器規範垂直顯示區域結束到垂直同步信號起始位置高度。 垂直同步信號起始位置 (Line) = (VSTR + 1)	0	RW

REG[1Fh] VSYNC Pulse Width Register (VPWR)

Bit	說明	初始值	Access
7	VSYNC 動作準位 0 : Low 動作。 1 : High 動作。	0	RW
6-0	VSYNC 脈波寬度 [6:0] VSYNC 脈波寬度 (Line) = (VPWR + 1)	0	RW

5-3 LCD 顯示模式

REG[20h] Display Configuration Register (DPCR)

Bit	說明	初始值	Access
7	圖層設定 (Layer Control) 0 : 單圖層。 1 : 雙圖層。	0	RW
6-4	NA	0	RO
3	HDIR 水平掃描方向設定 (n = SEG number) 0 : 由 SEG0 到 SEG(n-1)。 1 : 由 SEG(n-1) 到 SEG0。	0	RW
2	VDIR 垂直掃描方向設定 (n = COM number) 0 : 由 COM0 到 COM(n-1)。 1 : 由 COM(n-1) 到 COM0。	0	RW
1-0	NA	0	RO

REG[21h] Font Control Register 0 (FNCR0)

Bit	說明	初始值	Access
7	CGRAM/CGROM 文字選擇 0 : 選擇 CGROM font。 1 : 選擇 CGRAM font。 註： 此位元在文字模式時(REG[40h] bit 7 為 1)，用來選擇點陣圖來源，當 CGRAM 寫入時(REG[41h] bit 3-2 =01b)，此位元須設定為 0。當選擇 CGRAM 文字時，REG[21h] bit 5 必須被設為 1。	0	RW
6	NA	0	RO
5	外部/內部 CGROM 選擇 0 : 選擇內部 CGROM (REG[2Fh] 必須設為 00h) 1 : 選擇外部 CGROM (REG[2Eh] bit6 &bit7 必須設為 0)	0	RW
4-2	NA	0	RO
1-0	內部 CGROM 文字選擇 當 FNCR0 B7 = 0 且 B5 = 0，內部 CGROM 支援 ISO/IEC 8859-1~4 標準 8x16 文字，其支援英語及大部分歐洲國家語言文字。 00b : ISO/IEC 8859-1. 01b : ISO/IEC 8859-2. 10b : ISO/IEC 8859-3. 11b : ISO/IEC 8859-4.	0	RW

REG[22h] Font Control Register1 (FNCR1)

Bit	說明	初始值	Access
7	文字對齊功能設定 0 : 文字對齊功能關閉。 1 : 文字對齊功能開啟。	0	RW
6	文字通透模式 (Transparency) 0 : 文字具背景色模式。 1 : 文字背景通透模式，無背景色。	0	RW
5	NA	0	RO
4	文字旋轉 0 : 正常。 1 : 90 度。	0	RW
3-2	水平文字放大 00b : X1. 01b : X2. 10b : X3. 11b : X4.	0	RW

1-0	垂直文字放大 00b : X1. 01b : X2. 10b : X3. 11b : X4.	0	RW
-----	--	---	----

REG[23h] CGRAM Select Register (CGSR)

Bit	說明	初始值	Access
7-0	自造字型位置 CGRAM No. CGRAM 文字編號的設定，是用來寫入使用者自訂的文字點陣圖資料到 CGRAM 中。連續 16 筆資料寫入一個 8x16 文字點陣圖。注意 MWCR1 bit 3-2 先設定為 01b(CGRAM)，超過 16 筆資料寫入，會循環回到第一筆資料且覆蓋點陣圖。	0	RW

REG[24h] Horizontal Scroll Offset Register 0 (HOF0)

Bit	說明	初始值	Access
7-0	水平顯示捲動偏移[7:0] 設定水平捲動時每次移動的偏移量是多少像素。	0	RW

REG[25h] Horizontal Scroll Offset Register 1 (HOF1)

Bit	說明	初始值	Access
7-3	NA	0	RO
2-0	水平顯示捲動偏移[10:8] 設定水平捲動時每次移動的偏移量是多少像素。	0	RW

REG[26h] Vertical Scroll Offset Register 0 (VOF0)

Bit	說明	初始值	Access
7-0	垂直顯示捲動偏移 [7:0] 設定垂直捲動時每次移動的偏移量是多少像素。	0	RW

REG[27h] Vertical Scroll Offset Register 1 (VOF1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	垂直顯示捲動偏移[9:8] 設定垂直捲動時每次移動的偏移量是多少像素。	0	RW

REG[29h] Font Line Distance Setting Register (FLDR)

Bit	說明	初始值	Access
7-5	NA	0	RO
4-0	文字行距設定 在文字模式下，用來設定文字間的行距 (單位: 像素)。	0	RW

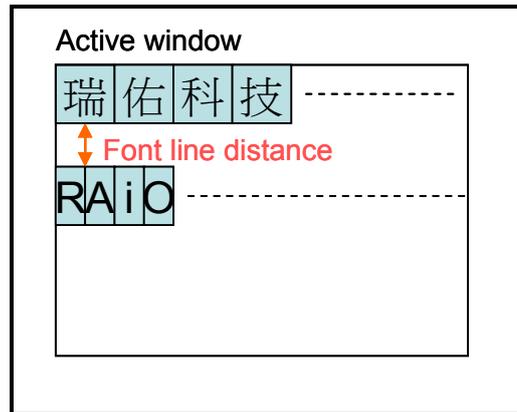


圖 5-1 : 文字行距

REG[2Ah] Font Write Cursor Horizontal Position Register 0 (F_CURXL)

Bit	說明	初始值	Access
7-0	文字寫入時的水平游標位置[7:0] 設定文字寫入的水平游標位置。	0	RW

REG[2Bh] Font Write Cursor Horizontal Position Register 1 (F_CURXH)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	文字寫入時的水平游標位置[9:8] 設定文字寫入的水平游標位置。	0	RW

REG[2Ch] Font Write Cursor Vertical Position Register 0 (F_CURYL)

Bit	說明	初始值	Access
7-0	文字寫入時的垂直游標位置[7:0] 設定文字寫入的垂直游標位置。	0	RW

REG[2Dh] Font Write Cursor Vertical Position Register 1 (F_CURYH)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	文字寫入時的垂直游標位置[8] 設定文字寫入的垂直游標位置。	0	RW

REG[2Eh] Font Write Type Setting Register

Bit	說明	初始值	Access																
7-6	文字大小設定 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th></th> <th>全型</th> <th>半型</th> <th>可變寬度</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>16x16</td> <td>8x16</td> <td>NX16</td> </tr> <tr> <td>01b</td> <td>24x24</td> <td>12x24</td> <td>NX24</td> </tr> <tr> <td>1Xb</td> <td>32x32</td> <td>16x32</td> <td>NX32</td> </tr> </tbody> </table> 註：文字寬度用“N”來表示，取決於字型的字碼。		全型	半型	可變寬度	00b	16x16	8x16	NX16	01b	24x24	12x24	NX24	1Xb	32x32	16x32	NX32	0	RW
	全型	半型	可變寬度																
00b	16x16	8x16	NX16																
01b	24x24	12x24	NX24																
1Xb	32x32	16x32	NX32																
5-0	字元水平間距設定 00h：字元無間距 01h：字元間距 = 1 像素 02h：字元間距 = 2 像素 . . 3Fh：字元間距 = 63 像素	0	RW																

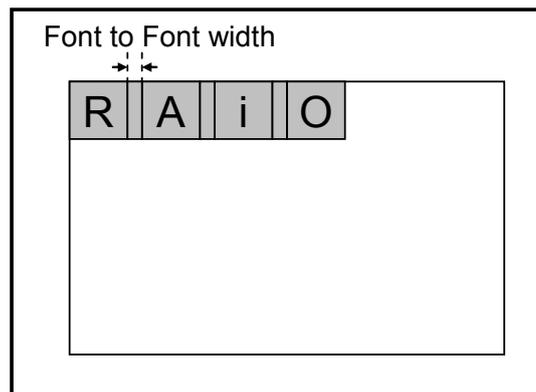


圖 5-2：字元間距

REG[2Fh] Serial Font ROM Setting

Bit	說明	初始值	Access																				
7-5	<p>選擇支援集通字庫的產品型號 (GT Serial Font ROM)</p> <p>000b: GT21L16TW / GT21H16T1W 001b: GT30L16U2W 010b: GT30L24T3Y / GT30H24T3Y 011b: GT30L24M1Z 100b: GT30L32S4W / GT30H32S4W</p>	0	RW																				
4-2	<p>設定 FONT ROM Coding</p> <p>對特定的集通字庫 (GT serial Font ROM) 而言，必須先設定編碼方式來分辨字碼的制定標準。</p> <p>000b: GB2312 001b: GB12345/GB18030 010b: BIG5 011b: UNICODE 100b: ASCII 101b: UNI-Japanese 110b: JIS0208 111b: Latin/Greek/ Cyrillic / Arabic</p>	0	RW																				
1-0	<p>ASCII / Latin/Greek/ Cyrillic / Arabic</p> <table border="1"> <thead> <tr> <th></th> <th>ASCII</th> <th>拉丁/希臘/西里爾文</th> <th>阿拉伯文</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Normal</td> <td>Normal</td> <td>NA</td> </tr> <tr> <td>01b</td> <td>Arial</td> <td>Variable Width</td> <td>Presentation Forms-A</td> </tr> <tr> <td>10b</td> <td>Roman</td> <td>NA</td> <td>Presentation Forms-B</td> </tr> <tr> <td>11b</td> <td>Bold</td> <td>NA</td> <td>NA</td> </tr> </tbody> </table>		ASCII	拉丁/希臘/西里爾文	阿拉伯文	00b	Normal	Normal	NA	01b	Arial	Variable Width	Presentation Forms-A	10b	Roman	NA	Presentation Forms-B	11b	Bold	NA	NA	0	RW
	ASCII	拉丁/希臘/西里爾文	阿拉伯文																				
00b	Normal	Normal	NA																				
01b	Arial	Variable Width	Presentation Forms-A																				
10b	Roman	NA	Presentation Forms-B																				
11b	Bold	NA	NA																				

5-4 工作視窗及捲動視窗設定

REG[30h] Horizontal Start Point 0 of Active Window (HSAW0)

Bit	說明	初始值	Access
7-0	工作視窗的水平起始點[7:0]	0	RW

REG[31h] Horizontal Start Point 1 of Active Window (HSAW1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	工作視窗的水平起始點[9:8]	0	RW

REG[32h] Vertical Start Point 0 of Active Window (VSAW0)

Bit	說明	初始值	Access
7-0	工作視窗的垂直起始點[7:0]	0	RW

REG[33h] Vertical Start Point 1 of Active Window (VSAW1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	工作視窗的垂直起始點[8]	0	RW

REG[34h] Horizontal End Point 0 of Active Window (HEAW0)

Bit	說明	初始值	Access
7-0	工作視窗的水平結束點[7:0]	0	RW

REG[35h] Horizontal End Point 1 of Active Window (HEAW1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	工作視窗的水平結束點[9:8]	0	RW

REG[36h] Vertical End Point of Active Window 0 (VEAW0)

Bit	說明	初始值	Access
7-0	工作視窗的垂直結束點[7:0]	0	RW

REG[37h] Vertical End Point of Active Window (VEAW1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	工作視窗的垂直結束點[8]	0	RW

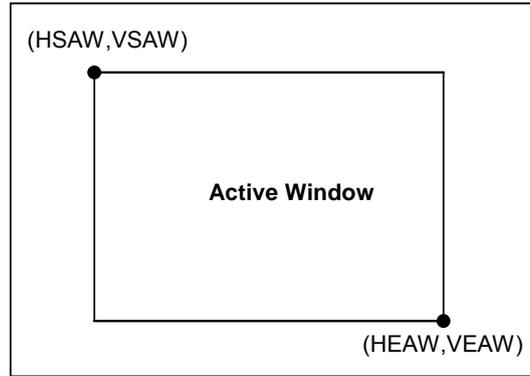


圖 5-3 : 工作視窗

REG[38h] Horizontal Start Point 0 of Scroll Window (HSSW0)

Bit	說明	初始值	Access
7-0	捲動視窗的水平起始點[7:0]	0	RW

REG[39h] Horizontal Start Point 1 of Scroll Window (HSSW1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	捲動視窗的水平起始點[9:8]	0	RW

REG[3Ah] Vertical Start Point 0 of Scroll Window (VSSW0)

Bit	說明	初始值	Access
7-0	捲動視窗的垂直起始點[7:0]	0	RW

REG[3Bh] Vertical Start Point 1 of Scroll Window (VSSW1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	捲動視窗的垂直起始點[8]	0	RW

REG[3Ch] Horizontal End Point 0 of Scroll Window (HESW0)

Bit	說明	初始值	Access
7-0	捲動視窗的水平結束點[7:0]	0	RW

REG[3Dh] Horizontal End Point 1 of Scroll Window (HESW1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	捲動視窗的水平結束點[9:8]	0	RW

REG[3Eh] Vertical End Point 0 of Scroll Window (VESW0)

Bit	說明	初始值	Access
7-0	捲動視窗的垂直結束點[7:0]	0	RW

REG[3Fh] Vertical End Point 1 of Scroll Window (VESW1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	捲動視窗的垂直結束點[8]	0	RW

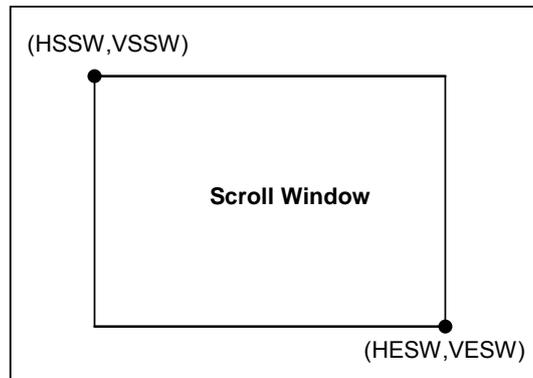


圖 5-4 : 捲動視窗

5-5 游標設定

REG[40h] Memory Write Control Register 0 (MWCR0)

Bit	說明	初始值	Access
7	顯示模式設定 0：繪圖模式。 1：文字模式。	0	RW
6	文字寫入游標/記憶體寫入游標設定 0：設定文字/記憶體寫入游標為不顯示。 1：設定文字/記憶體寫入游標為顯示。	0	RW
5	文字寫入游標/記憶體寫入游標閃爍設定 0：游標不閃爍。 1：游標閃爍。	0	RW
4	NA	0	RO
3-2	繪圖模式時的記憶體寫入方向 00b：左 → 右，然後上 → 下。 01b：右 → 左，然後上 → 下。 10b：上 → 下，然後左 → 右。 11b：下 → 上，然後左 → 右。	0	RW
1	記憶體寫入游標自動增加功能設定 0：當記憶體寫入時游標位置自動加一。 1：當記憶體寫入時游標位置不會自動加一。	0	RW
0	記憶體讀取游標自動增加功能設定 0：當記憶體讀取時游標位置自動加一。 1：當記憶體讀取時游標位置不會自動加一。	0	RW

REG[41h] Memory Write Control Register1 (MWCR1)

Bit	說明	初始值	Access
7	圖形游標設定 0：圖形游標關閉。 1：圖形游標開啟。	0	RW
6-4	圖形游標的選擇 從 8 款圖形游標中選擇一款。(000b to 111b) 000b：選擇圖形游標 1。 001b：選擇圖形游標 2。 010b：選擇圖形游標 3。 ： ： 111b：選擇圖形游標 8。	0	RW

3-2	寫入目的地選擇 00b : 圖層 1~2。 01b : CGRAM。 10b : 圖形游標。 11b : Pattern。 註：當選擇 CGRAM (01b), REG[21h] bit 7 設定為“0”。	0	RW
1	NA	0	RO
0	讀寫圖層選擇 當顯示圖層小於或等於 480x400 或色彩深度等於 8bpp 時: 0 : 圖層 1。 1 : 圖層 2。 當顯示圖層大於 480x400 且色彩深度大於 > 8bpp 時: 寫入圖層維持在圖層 1。	0	RW

REG[44h] Blink Time Control Register (BTCR)

Bit	說明	初始值	Access
7-0	文字閃爍時間設定 (Unit: Frame) 00h : 1 個 Frame 週期。 01h : 2 個 Frame 週期。 02h : 3 個 Frame 週期。 : : : FFh : 256 個 Frame 週期。	0	RW

REG[45h] Memory Read Cursor Direction (MRCD)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	繪圖模式時的記憶體讀取方向 00b : 左 → 右，然後上 → 下。 01b : 右 → 左，然後上 → 下。 10b : 上 → 下，然後左 → 右。 11b : 下 → 上，然後左 → 右。	0	RW

REG[46h] Memory Write Cursor Horizontal Position Register 0 (CURH0)

Bit	說明	初始值	Access
7-0	記憶體寫入游標水平位置[7:0]	0	RW

REG[47h] Memory Write Cursor Horizontal Position Register 1 (CURH1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	記憶體寫入游標水平位置[9:8]	0	RW

REG[48h] Memory Write Cursor Vertical Position Register 0 (CURV0)

Bit	說明	初始值	Access
7-0	記憶體寫入游標垂直位置[7:0]	0	RW

REG[49h] Memory Write Cursor Vertical Position Register 1 (CURV1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	記憶體寫入游標垂直位置[8]	0	RW

REG[4Ah] Memory Read Cursor Horizontal Position Register 0 (RCURH0)

Bit	說明	初始值	Access
7-0	記憶體讀取游標水平位置[7:0]	0	RW

REG[4Bh] Memory Read Cursor Horizontal Position Register 1 (RCURH01)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	記憶體讀取游標水平位置[9:8]	0	RW

REG[4Ch] Memory Read Cursor Vertical Position Register 0 (RCURV0)

Bit	說明	初始值	Access
7-0	記憶體讀取游標垂直位置[7:0]	0	RW

REG[4Dh] Memory Read Cursor Vertical Position Register 1 (RCURV1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	記憶體讀取游標垂直位置 [8]	0	RW

REG[4Eh] Font Write Cursor and Memory Write Cursor Horizontal Size Register (CURHS)

Bit	說明	初始值	Access
7-5	NA	0	RO
4-0	文字寫入游標水平大小設定 [4:0] 單位：像素 註：當文字放大時，游標設定會與文字放大倍數相同。	7h	RW

REG[4Fh] Font Write Cursor Vertical Size Register (CURVS)

Bit	說明	初始值	Access
7-5	NA	0	RO
4-0	文字寫入游標垂直大小設定 [4:0] 單位：像素 註：當文字放大時，游標設定會與文字放大倍數相同。	0	RW

5-6 BTE 引擎

REG[50h] BTE Function Control Register 0 (BECR0)

Bit	說明	初始值	Access
7	BTE 功能設定與狀態 Write 0：不動作。 1：BTE 功能開啟。 Read 0：BTE 處於閒置狀態。 1：BTE 處於忙碌狀態。	0	RW
6	BTE 做資料搬移時 ”讀取來源的資料選擇” 0：區塊模式 - 來源的資料是區塊資料讀出 (Rectangular Region)。 1：線性模式 - 來源的資料是連續資料讀出。	0	RW
5	BTE 做資料搬移時 ”寫入目的地的資料選擇” 0：區塊模式 - 目的地的資料是區塊資料寫入 (Rectangular Region)。 1：線性模式，目的地的資料是連續資料寫入。	0	RW
4-0	NA	0	RO

REG[51h] BTE Function Control Register1 (BECR1)

Bit	說明	初始值	Access
7-4	BTE 的光柵運算碼 (ROP Code) Bit[3:0] ROP是Raster Operation的縮寫。有些BTE操作碼要搭配光柵運算碼才能知道詳細的動作，請參考章節 7-6。	0	RW
3-0	BTE 的操作碼 (Operation Code) Bit[3:0] RA8875 包含一 2D的BTE引擎 (Block Transfer Engine)，可以執行 13 個BTE動作 (也就是操作碼 1100b ~ 0000b)，而 1111b ~ 1101b 不被使用。有些操作碼要搭配上光柵運算碼才能知道詳細的動作，請參考章節 7-6。	0	RW

REG[52h] Layer Transparency Register0 (LTPR0)

Bit	說明	初始值	Access
7-6	圖層捲動模式 00b : 圖層 1 與圖層 2 同時捲動。 01b : 只有圖層 1 捲動。 10b : 只有圖層 2 捲動。 11b : 捲動緩衝 (用圖層 2 當成捲動緩衝)。	0	RW
5	用 BGTR 設定浮動視窗通透顯示 0: 關閉。 1: 開啟。	0	RW
4-3	NA	0	RO
2-0	圖層顯示模式 000b : 只有圖層 1 顯示。 001b : 只有圖層 2 顯示。 010b : 顯示圖層 1 與圖層 2 的漸進/漸出模式。 011b : 顯示圖層 1 與圖層 2 的通透模式。 100b : Boolean OR。 101b : Boolean AND。 110b : 浮動視窗模式 (Floating window mode)。 111b : 保留。	0	RW

Note : 建議當使用緩衝捲動功能時，暫存器[40h] Bit 7 應設定為 1'b0 。

REG[53h] Layer Transparency Register1 (LTPR1)

Bit	說明	初始值	Access
7-4	圖層 2 的通透 (Transparency) 設定 0000b : Total 顯示。 0001b : 7/8 顯示。 0010b : 3/4 顯示。 0011b : 5/8 顯示。 0100b : 1/2 顯示。 0101b : 3/8 顯示。 0110b : 1/4 顯示。 0111b : 1/8 顯示。 1000b : 顯示關閉。	0	RW
3-0	圖層 1 的通透 (Transparency) 設定 0000b : Total 顯示。 0001b : 7/8 顯示。 0010b : 3/4 顯示。 0011b : 5/8 顯示。	0	RW

0100b : 1/2 顯示。		
0101b : 3/8 顯示。		
0110b : 1/4 顯示。		
0111b : 1/8 顯示。		
1000b : 顯示關閉。		

REG[54h] Horizontal Source Point 0 of BTE (HSBE0)

Bit	說明	初始值	Access
7-0	BTE 讀取資料來源的水平位置 BTE [7:0]	0	RW

REG[55h] Horizontal Source Point 1 of BTE (HSBE1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	BTE 讀取資料來源的水平位置 [9:8]	0	RW

REG[56h] Vertical Source Point 0 of BTE (VSBE0)

Bit	說明	初始值	Access
7-0	BTE 讀取資料來源的垂直位置 [7:0]	0	RW

REG[57h] Vertical Source Point 1 of BTE (VSBE1)

Bit	說明	初始值	Access
7	讀取資料來源的的圖層 0 : 圖層 1。 1 : 圖層 2。	0	RW
6-1	NA	0	RO
0	BTE 讀取資料來源的垂直位置[8]	0	RW

REG[58h] Horizontal Destination Point 0 of BTE (HDBE0)

Bit	說明	初始值	Access
7-0	BTE 寫入目標的水平位置 [7:0]	0	RW

REG[59h] Horizontal Destination Point 1 of BTE (HDBE1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	BTE 寫入目標的水平位置[9:8]	0	RW

REG[5Ah] Vertical Destination Point 0 of BTE (VDBE0)

Bit	說明	初始值	Access
7-0	BTE 寫入目標的垂直位置[7:0]	0	RW

REG[5Bh] Vertical Destination Point 1 of BTE (VDBE1)

Bit	說明	初始值	Access
7	BTE 寫入目標的圖層 0 : 圖層 1。 1 : 圖層 2。	0	RW
6-1	NA	0	RO
0	BTE 寫入目標的垂直位置[8]	0	RW

REG[5Ch] BTE Width Register 0 (BEWR0)

Bit	說明	初始值	Access
7-0	BTE 處理區塊的寬度[7:0]	0	RW

REG[5Dh] BTE Width Register 1 (BEWR1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	BTE 處理區塊的寬度[9:8]	0	RW

REG[5Eh] BTE Height Register 0 (BEHR0)

Bit	說明	初始值	Access
7-0	BTE 處理區塊的高度[7:0]	0	RW

REG[5Fh] BTE Height Register 1 (BEHR1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	BTE 處理區塊的高度[9:8]	0	RW

REG[60h] Background Color Register 0 (BGCR0)

Bit	說明	初始值	Access
7-5	NA	0	R0
4-0	紅色背景色 [4:0] 若 REG[10h] Bit[3:2] 設定為 256 色，此暫存器只用 Bit[2:0]。 若 REG[10h] Bit[3:2] 設定為 65K 色，此暫存器用到 Bit[4:0]。	0	RW

此暫存器用在設定 BTE 紅色部分背景顏色。

REG[61h] Background Color Register 1 (BGCR1)

Bit	說明	初始值	Access
7-6	NA	0	R0
5-0	綠色背景色 [5:0] 若 REG[10h] Bit[3:2] 設定為 256 色，此暫存器只用 Bit[2:0]。 若 REG[10h] Bit[3:2] 設定為 65K 色，此暫存器用到 Bit[5:0]。	0	RW

此暫存器用在設定 BTE 綠色部分背景顏色。

REG[62h] Background Color Register 2 (BGCR2)

Bit	說明	初始值	Access
7-5	NA	0	R0
4-0	藍色背景色 [4:0] 若 REG[10h] Bit[3:2] 設定為 256 色，此暫存器只用 Bit[1:0]。 若 REG[10h] Bit[3:2] 設定為 65K 色，此暫存器用到 Bit[4:0]。	0	RW

此暫存器用在設定 BTE 藍色部分背景顏色。

REG[63h] Foreground Color Register 0 (FGCR0)

Bit	說明	初始值	Access
7-5	NA	0	R0
4-0	紅色前景色 [4:0] 若 REG[10h] Bit[3:2] 設定為 256 色，此暫存器只用 Bit[2:0]。 若 REG[10h] Bit[3:2] 設定為 65K 色，此暫存器用到 Bit[4:0]。	1Fh	RW

此暫存器用在設定 BTE 紅色部分前景顏色。

REG[64h] Foreground Color Register 1 (FGCR1)

Bit	說明	初始值	Access
7-6	NA	0	R0
5-0	綠色前景色 [5:0] 若 REG[10h] Bit[3:2] 設定為 256 色，此暫存器只用 Bit[2:0]。 若 REG[10h] Bit[3:2] 設定為 65K 色，此暫存器用到 Bit[5:0]。	3Fh	RW

此暫存器用在設定 BTE 綠色部分前景顏色。

REG[65h] Foreground Color Register 2 (FGCR2)

Bit	說明	初始值	Access
7-5	NA	0	R0
4-0	藍色前景色 [4:0] 若 REG[10h] Bit[3:2] 設定為 256 色，此暫存器只用 Bit[1:0]。 若 REG[10h] Bit[3:2] 設定為 65K 色，此暫存器用到 Bit[4:0]。	1Fh	RW

REG[66h] Pattern Set No for BTE (PTNO)

Bit	說明	初始值	Access
7	Pattern 格式 (Pattern Format) 0: 8x8。 1: 16x16。	0	RW
6-4	NA	0	RO
3-0	Pattern Set No 若 pattern 格式為 8x8，Pattern 設定[3:0] 是有效的。 若 pattern 格式為 16x16，Pattern 設定[1:0] 是有效的。	0	RW

REG[67h] Background Color Register for Transparent 0 (BGTR0)

Bit	說明	初始值	Access
7-5	NA	0	R0
4-0	通透模式下的背景色之紅色部分[4:0] 若 REG[10h] Bit[3:2] 設定為 256 色，此暫存器只用 Bit[2:0]。 若 REG[10h] Bit[3:2] 設定為 65K 色，此暫存器用到 Bit[4:0]。	0	RW

REG[68h] Background Color Register for Transparent 1 (BGTR1)

Bit	說明	初始值	Access
7-6	NA	0	R0
5-0	<p>通透模式下的背景色之綠色部分[5:0]</p> <p>若 REG[10h] Bit[3:2] 設定為 256 色，此暫存器只用 Bit[2:0]。</p> <p>若 REG[10h] Bit[3:2] 設定為 65K 色，此暫存器用到 Bit[5:0]。</p>	0	RW

REG[69h] Background Color Register for Transparent 2 (BGTR2)

Bit	說明	初始值	Access
7-5	NA	0	R0
4-0	<p>通透模式下的背景色之藍色部分[4:0]</p> <p>若 REG[10h] Bit[3:2] 設定為 256 色，此暫存器只用 Bit[1:0]。</p> <p>若 REG[10h] Bit[3:2] 設定為 65K 色，此暫存器用到 Bit[4:0]。</p>	0	RW

5-7 觸控面板

REG[70h] Touch Panel Control Register 0 (TPCR0)

Bit	說明	初始值	Access
7	觸控面板功能設定 0：關閉。 1：開啟。	0	RW
6-4	觸控面板控制器取樣時間設定 000b：ADC 取樣時間為 512 個系統時脈週期。 001b：ADC 取樣時間為 1024 個系統時脈週期。 010b：ADC 取樣時間為 2048 個系統時脈週期。 011b：ADC 取樣時間為 4096 個系統時脈週期。 100b：ADC 取樣時間為 8192 個系統時脈週期。 101b：ADC 取樣時間為 16384 個系統時脈週期。 110b：ADC 取樣時間為 32768 個系統時脈週期。 111b：ADC 取樣時間為 65536 個系統時脈週期。	0	RW
3	觸控面板喚醒模式 0：關閉觸控事件喚醒模式。 1：觸控事件可喚醒睡眠模式。	0	RW
2-0	觸控面板控制器 ADC 時脈設定 000b：系統頻率。 001b：系統頻率 / 2。 010b：系統頻率 / 4。 011b：系統頻率 / 8。 100b：系統頻率 / 16。 101b：系統頻率 / 32。 110b：系統頻率 / 64。 111b：系統頻率 / 128。	0	RW

REG[71h] Touch Panel Control Register 1 (TPCR1)

Bit	說明	初始值	Access
7	N/A	1	RO
6	觸控面板模式設定 0：自動模式。 1：手動模式。	0	RW
5	觸控面板控制器 ADC 參考電壓(Vref)來源設定 0：內部產生參考電壓。 1：外部輸入參考電壓，ADC 參考電壓準位 = 1/2 VDD。	0	RW
4-3	NA	0	RO

2	<p>觸控中斷訊號的消除彈跳電路選擇</p> <p>0: 關閉消除彈跳電路。</p> <p>1: 開啟消除彈跳電路。</p>	0	R/W
1-0	<p>觸控面板手動模式之選擇位元</p> <p>00b : 閒置模式。觸控控制單元進入閒置模式。</p> <p>01b : 偵測觸摸事件發生。在此模式控制器會偵測觸摸事件的發生，事件發生可以引發中斷或是由暫存器得知(REG[F1h] Bit2)。</p> <p>10b : X 軸資料擷取模式。在此模式觸摸位置的 X 軸資料會被儲存至 REG[72h] 和 REG[74h]。</p> <p>11b : Y 軸資料擷取模式。在此模式觸摸位置的 Y 軸資料會被儲存至 REG[73h] and REG[74h]。</p>	0	RW

REG[72h] Touch Panel X High Byte Data Register (TPXH)

Bit	說明	初始值	Access
7-0	觸控面板 X 軸資料高位元組 Bit [9:2]	0	RW

REG[73h] Touch Panel Y High Byte Data Register (TPYH)

Bit	說明	初始值	Access
7-0	觸控面板 Y 軸資料高位元組[9:2]	0	RW

REG[74h] Touch Panel X/Y Low Byte Data Register (TPXYL)

Bit	說明	初始值	Access
7	<p>ADET</p> <p>觸摸事件偵測</p> <p>0 : 觸控面板被觸摸。</p> <p>1 : 觸控面板未被觸摸。</p>	1	RO
6-4	NA	0	RO
3-2	觸控面板 Y 軸資料低二位元 Bit[1:0]	0	RW
1-0	觸控面板 X 軸資料低二位元 Bit[1:0]	0	RW

5-8 圖形游標

REG[80h] Graphic Cursor Horizontal Position Register 0 (GCHP0)

Bit	說明	初始值	Access
7-0	圖形游標水平位置[7:0]	0	RW

REG[81h] Graphic Cursor Horizontal Position Register 1 (GCHP1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	圖形游標水平位置[9:8]	0	RW

REG[82h] Graphic Cursor Vertical Position Register 0 (GCVP0)

Bit	說明	初始值	Access
7-0	圖形游標垂直位置[7:0]	0	RW

REG[83h] Graphic Cursor Vertical Position Register 1 (GCVP1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	圖形游標垂直位置[8]	0	RW

REG[84h] Graphic Cursor Color 0 (GCC0)

Bit	說明	初始值	Access
7-0	256 色圖形游標顏色 0 設定 設定格式為 RRRGGGBB。	0	RW

REG[85h] Graphic Cursor Color 1 (GCC1)

Bit	說明	初始值	Access
7-0	256 色圖形游標顏色 1 設定 設定格式為 RRRGGGBB。	0	RW

5-9 PLL設定

REG[88h] PLL Control Register 1 (PLLC1)

Bit	說明	初始值	Access
7	PLLDIVM PLL 前置驅動電路之參數。 0：除以 1。 1：除以 2。	0	RW
6-5	NA	0	RO
4-0	PLLDIVN[4:0] PLL 輸入參數，輸入值必須是1~31。(注意“0”是禁止使用的!)	07h	RW

REG[89h] PLL Control Register 2 (PLLC2)

Bit	說明	初始值	Access
7-3	NA	0	RO
2-0	PLLDIVK[2:0] PLL 輸出除頻參數。 000b：除以 1。 001b：除以 2。 010b：除以 4。 011b：除以 8。 100b：除以 16。 101b：除以 32。 110b：除以 64。 111b：除以 128。	03h	RW

註：

1. 系統頻率(SYS_CLK) 預設值與外部晶體振盪器 Clock (FIN) 頻率相同。
2. 當 REG[88h]或 REG[89h]被設定後，為保證 PLL 輸出穩定，須等待一段「鎖頻時間」(> 100us)。
3. 晶體振盪器頻率 (FIN) 的輸入值必須介於 15MHz~30MHz 之間。

$F_{PLL} = FIN * (PLLDIVN [4:0] + 1)$ 必需等於或大於 110 MHz。

下表為外部晶體振盪 (FIN) 與 REG[88h] Bit[4:0] 的參考設定：

OSC Clock (FIN) X'tal (MHz)	PLLDIVN[4:0] REG[88h] Bit[4:0]
15	>= 7
16	>= 7
20	>= 5
25	>= 4
30	>= 3

4. RA8875 的內部系統頻率 (SYS_CLK) 是結合振盪電路及 PLL 電路所產生，頻率計算公式如下：

$$SYS_CLK = FIN * (PLLDIVN [4:0] + 1) / ((PLLDIVM + 1) * (2^{PLLDIVK [2:0]}))$$

5-10 脈波寬度調變 (PWM)

REG[8Ah] PWM1 Control Register (P1CR)

Bit	說明	初始值	Access																
7	脈波寬度調變 (PWM1) 設定 0: 關閉, 此狀態下, PWM1 輸出準位依照此暫存器 Bit6 決定。 1: 開啟。	0	RW																
6	PWM1 關閉時的準位 0: 當 PWM 關閉或於睡眠模式時, PWM1 輸出為"Low" 狀態。 1: 當 PWM 關閉或於睡眠模式時, PWM1 輸出為"High" 狀態。 此位元只有在暫存器 P1CR Bit 4 為 0 才有效。	0	RW																
5	保留	0	RO																
4	PWM1 功能選擇 0: PWM1 功能。 1: PWM1 固定輸出一頻率為外部晶體振盪器 Clock (Fin) 頻率 1/16 的 Clock。 PWM1 = F _{OSC} / 16 (F _{OSC} 為外部晶體振盪器的頻率)	0	RW																
3-0	PWM1 電路的時脈來源選擇 <table border="1" style="margin-left: 20px;"> <tbody> <tr><td>0000b : SYS_CLK / 1</td><td>1000b : SYS_CLK / 256</td></tr> <tr><td>0001b : SYS_CLK / 2</td><td>1001b : SYS_CLK / 512</td></tr> <tr><td>0010b : SYS_CLK / 4</td><td>1010b : SYS_CLK / 1024</td></tr> <tr><td>0011b : SYS_CLK / 8</td><td>1011b : SYS_CLK / 2048</td></tr> <tr><td>0100b : SYS_CLK / 16</td><td>1100b : SYS_CLK / 4096</td></tr> <tr><td>0101b : SYS_CLK / 32</td><td>1101b : SYS_CLK / 8192</td></tr> <tr><td>0110b : SYS_CLK / 64</td><td>1110b : SYS_CLK / 16384</td></tr> <tr><td>0111b : SYS_CLK / 128</td><td>1111b : SYS_CLK / 32768</td></tr> </tbody> </table> <p>“SYS_CLK”代表系統頻率, 例如 SYS_CLK 為 20MHz, 當 Bit[3:0] = 0001b 時, PWM1 時脈來源為 10MHz。</p>	0000b : SYS_CLK / 1	1000b : SYS_CLK / 256	0001b : SYS_CLK / 2	1001b : SYS_CLK / 512	0010b : SYS_CLK / 4	1010b : SYS_CLK / 1024	0011b : SYS_CLK / 8	1011b : SYS_CLK / 2048	0100b : SYS_CLK / 16	1100b : SYS_CLK / 4096	0101b : SYS_CLK / 32	1101b : SYS_CLK / 8192	0110b : SYS_CLK / 64	1110b : SYS_CLK / 16384	0111b : SYS_CLK / 128	1111b : SYS_CLK / 32768	0	RW
0000b : SYS_CLK / 1	1000b : SYS_CLK / 256																		
0001b : SYS_CLK / 2	1001b : SYS_CLK / 512																		
0010b : SYS_CLK / 4	1010b : SYS_CLK / 1024																		
0011b : SYS_CLK / 8	1011b : SYS_CLK / 2048																		
0100b : SYS_CLK / 16	1100b : SYS_CLK / 4096																		
0101b : SYS_CLK / 32	1101b : SYS_CLK / 8192																		
0110b : SYS_CLK / 64	1110b : SYS_CLK / 16384																		
0111b : SYS_CLK / 128	1111b : SYS_CLK / 32768																		

REG[8Bh] PWM1 Duty Cycle Register (P1DCR)

Bit	說明	初始值	Access
7-0	<p>PWM 的 Duty 設定</p> <p>00h → 1 / 256 高準位時間。 01h → 2 / 256 高準位時間。 02h → 3 / 256 高準位時間。 ⋮ FEh → 255 / 256 高準位時間。 FFh → 256 / 256 高準位時間。</p>	0	RW

REG[8Ch] PWM2 Control Register (P2CR)

Bit	說明	初始值	Access																
7	<p>脈波寬度調變 (PWM2) 設定</p> <p>0: 關閉, 此狀態下, PWM2 輸出準位依照此暫存器 Bit6 決定。 1: 開啟。</p>	0	RW																
6	<p>PWM2 關閉時的準位</p> <p>0: 當 PWM 關閉或於睡眠模式時, PWM2 輸出為"Low" 狀態。 1: 當 PWM 關閉或於睡眠模式時, PWM2 輸出為"High" 狀態。 此位元只有在暫存器 P2CR Bit 4 為 0 才有效。</p>	0	RW																
5	保留	0	RO																
4	<p>PWM2 功能選擇</p> <p>0: PWM2 功能。 1: PWM2 輸出一相同於系統時脈信號頻率的 Clock。 PWM2 = SYS_CLK / 16</p>	0	RW																
3-0	<p>PWM2 電路的時脈來源選擇</p> <table border="1" style="margin-left: 20px;"> <tbody> <tr> <td>0000b : SYS_CLK / 1</td> <td>1000b : SYS_CLK / 256</td> </tr> <tr> <td>0001b : SYS_CLK / 2</td> <td>1001b : SYS_CLK / 512</td> </tr> <tr> <td>0010b : SYS_CLK / 4</td> <td>1010b : SYS_CLK / 1024</td> </tr> <tr> <td>0011b : SYS_CLK / 8</td> <td>1011b : SYS_CLK / 2048</td> </tr> <tr> <td>0100b : SYS_CLK / 16</td> <td>1100b : SYS_CLK / 4096</td> </tr> <tr> <td>0101b : SYS_CLK / 32</td> <td>1101b : SYS_CLK / 8192</td> </tr> <tr> <td>0110b : SYS_CLK / 64</td> <td>1110b : SYS_CLK / 16384</td> </tr> <tr> <td>0111b : SYS_CLK / 128</td> <td>1111b : SYS_CLK / 32768</td> </tr> </tbody> </table> <p>例如 SYS_CLK 為 20MHz, 當 Bit[3:0] =0010b 時, PWM2 時脈來源為 5MHz。</p>	0000b : SYS_CLK / 1	1000b : SYS_CLK / 256	0001b : SYS_CLK / 2	1001b : SYS_CLK / 512	0010b : SYS_CLK / 4	1010b : SYS_CLK / 1024	0011b : SYS_CLK / 8	1011b : SYS_CLK / 2048	0100b : SYS_CLK / 16	1100b : SYS_CLK / 4096	0101b : SYS_CLK / 32	1101b : SYS_CLK / 8192	0110b : SYS_CLK / 64	1110b : SYS_CLK / 16384	0111b : SYS_CLK / 128	1111b : SYS_CLK / 32768	0	RW
0000b : SYS_CLK / 1	1000b : SYS_CLK / 256																		
0001b : SYS_CLK / 2	1001b : SYS_CLK / 512																		
0010b : SYS_CLK / 4	1010b : SYS_CLK / 1024																		
0011b : SYS_CLK / 8	1011b : SYS_CLK / 2048																		
0100b : SYS_CLK / 16	1100b : SYS_CLK / 4096																		
0101b : SYS_CLK / 32	1101b : SYS_CLK / 8192																		
0110b : SYS_CLK / 64	1110b : SYS_CLK / 16384																		
0111b : SYS_CLK / 128	1111b : SYS_CLK / 32768																		

REG[8Dh] PWM2 Control Register (P2DCR)

Bit	說明	初始值	Access
7-0	<p>PWM 的 Duty 設定</p> <p>00h → 1 / 256 高準位時間。 01h → 2 / 256 高準位時間。 02h → 3 / 256 高準位時間。 ⋮ FEh → 255 / 256 高準位時間。 FFh → 256 / 256 高準位時間。</p>	0	RW

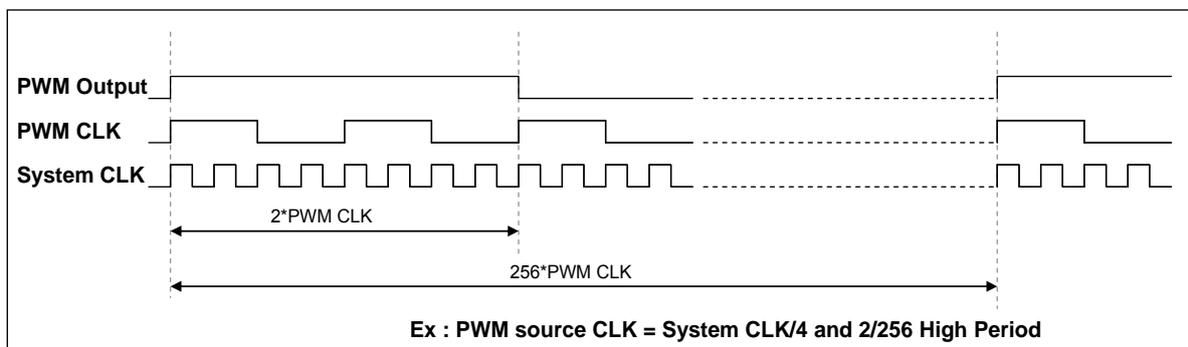


圖 5-5 : PWM 的 Duty

REG[8Eh] Memory Clear Control Register (MCLR)

Bit	說明	初始值	Access
7	<p>記憶體清除功能</p> <p>0 : 記憶體清除動作結束或停止。當此 Bit 被寫入為 0 時，RA8875 會停止記憶體清除動作。當讀回此 Bit = 0，則此代表記憶體清除動作已完成。</p> <p>1 : 記憶體清除動作開始。</p>	0	RW
6	<p>記憶體清除範圍設定</p> <p>0 : 記憶體清除範圍為顯示視窗，請參考 REG[14h],[19h], [1Ah]。</p> <p>1 : 記憶體清除範圍為工作視窗，請參考 REG[30h~37h] 的設定。清除區域請依照 REG[41h] Bit0 的設定。</p>	0	RW
5-0	NA	0	RO

5-11 繪圖控制暫存器

REG[90h] Draw Line/Circle/Square Control Register (DCR)

Bit	說明	初始值	Access
7	<p>畫直線/矩形/三角形的起始訊號</p> <p>寫入功能</p> <p>0: 停止畫直線/矩形/三角形的繪圖功能。</p> <p>1: 開始畫直線/矩形/三角形的繪圖功能。</p> <p>讀取功能</p> <p>0: 直線/矩形/三角形的繪圖完成。</p> <p>1: 直線/矩形/三角形的繪圖進行中。</p>	0	RW
6	<p>畫圓形的起始訊號</p> <p>寫入功能</p> <p>0: 停止圓形繪圖功能。</p> <p>1: 啟動圓形繪圖功能。</p> <p>讀取功能</p> <p>0: 圓形繪圖完成。</p> <p>1: 圓形繪圖進行中。</p>	0	RW
5	<p>填滿圓形/矩形/三角形訊號</p> <p>0: 不填滿。</p> <p>1: 填滿。</p>	0	RW
4	<p>畫直線或矩形選擇訊號</p> <p>0: 畫直線。</p> <p>1: 畫矩形。</p>	0	RW
3-1	NA	0	RO
0	<p>畫三角形或直線/矩形選擇訊號</p> <p>0: 畫直線或矩形。</p> <p>1: 畫三角形。</p>	0	RW

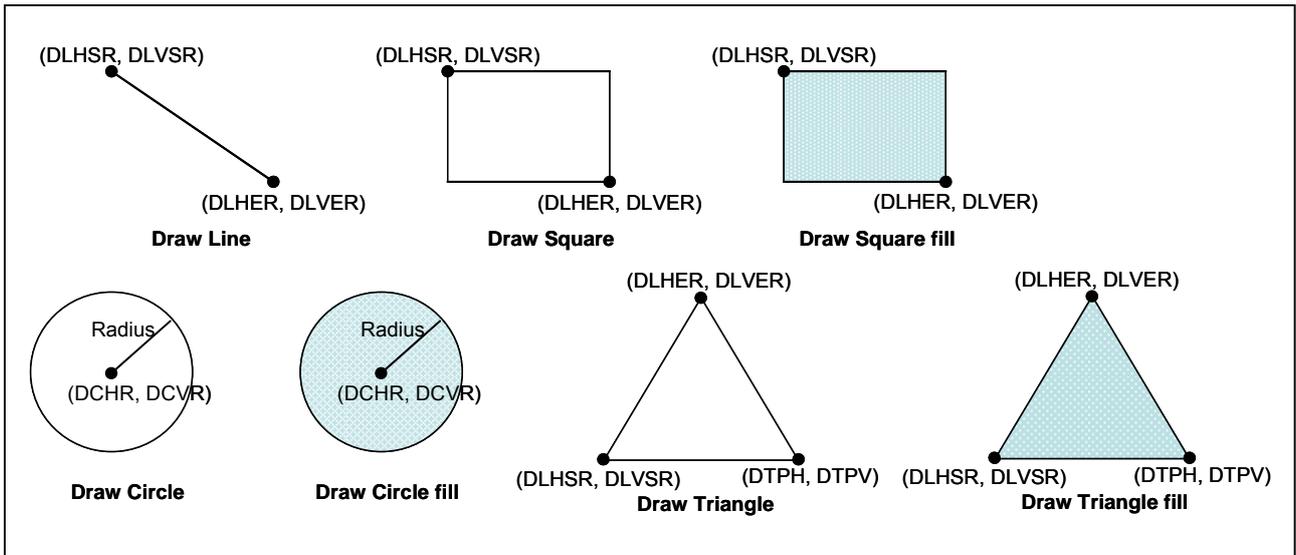


圖 5-6 : 繪圖功能參數

REG[91h] Draw Line/Square Horizontal Start Address Register0 (DLHSR0)

Bit	說明	初始值	Access
7-0	畫直線或矩形的水平起始位置[7:0]	0	RW

REG[92h] Draw Line/Square Horizontal Start Address Register1 (DLHSR1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	畫直線或矩形的水平起始位置[9:8]	0	RW

REG[93h] Draw Line/Square Vertical Start Address Register0 (DLVSR0)

Bit	說明	初始值	Access
7-0	畫直線或矩形的垂直起始位置[7:0]	0	RW

REG[94h] Draw Line/Square Vertical Start Address Register1 (DLVSR1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	畫直線或矩形的垂直起始位置[8]	0	RW

註：起始點與終點位置不相同

REG[95h] Draw Line/Square Horizontal End Address Register0 (DLHER0)

Bit	說明	初始值	Access
7-0	畫直線或矩形的水平結束位置[7:0]	0	RW

REG[96h] Draw Line/Square Horizontal End Address Register1 (DLHER1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	畫直線或矩形的水平結束位置[9:8]	0	RW

REG[97h] Draw Line/Square Vertical End Address Register0 (DLVER0)

Bit	說明	初始值	Access
7-0	畫直線或矩形的垂直結束位置[7:0]	0	RW

REG[98h] Draw Line/Square Vertical End Address Register1 (DLVER1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	畫直線或矩形的垂直結束位置[8]	0	RW

註：起始點與終點位置不相同

REG[99h] Draw Circle Center Horizontal Address Register0 (DCHR0)

Bit	說明	初始值	Access
7-0	畫圓形中心點的水平位置[7:0]	0	RW

REG[9Ah] Draw Circle Center Horizontal Address Register1 (DCHR1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	畫圓形中心點的水平位置[9:8]	0	RW

REG[9Bh] Draw Circle Center Vertical Address Register0 (DCVR0)

Bit	說明	初始值	Access
7-0	畫圓形中心點的垂直位置 [7:0]	0	RW

REG[9Ch] Draw Circle Center Vertical Address Register1 (DCVR1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	畫圓形中心點的垂直位置 [8]	0	RW

REG[9Dh] Draw Circle Radius Register (DCRR)

Bit	說明	初始值	Access
7-0	畫圓形的半徑 [7:0]	0	RW

REG[A0h] Draw Ellipse/Ellipse Curve/Circle Square Control Register

Bit	說明	初始值	Access
7	畫橢圓/圓形/矩形的起始訊號 寫入功能 0: 停止畫橢圓/圓形/矩形繪圖功能。 1: 啟動畫橢圓/圓形/矩形繪圖功能。 讀取功能 0: 橢圓/圓形/矩形繪圖完成。 1: 橢圓/圓形/矩形繪圖進行中。	0	RW
6	填滿橢圓/圓形/矩形訊號 0: 不填滿。 1: 填滿。	0	RW
5	畫橢圓/橢圓曲線或圓角方形選擇訊號 0: 畫橢圓/橢圓曲線 (依照 Bit4)。 1: 畫圓角方形。	0	RW
4	畫橢圓或橢圓曲線選擇訊號 0: 畫橢圓 1: 畫橢圓曲線	0	RW
3-2	NA	0	RO
1-0	畫部份橢圓曲線選擇訊號 (DECP)	0	RW

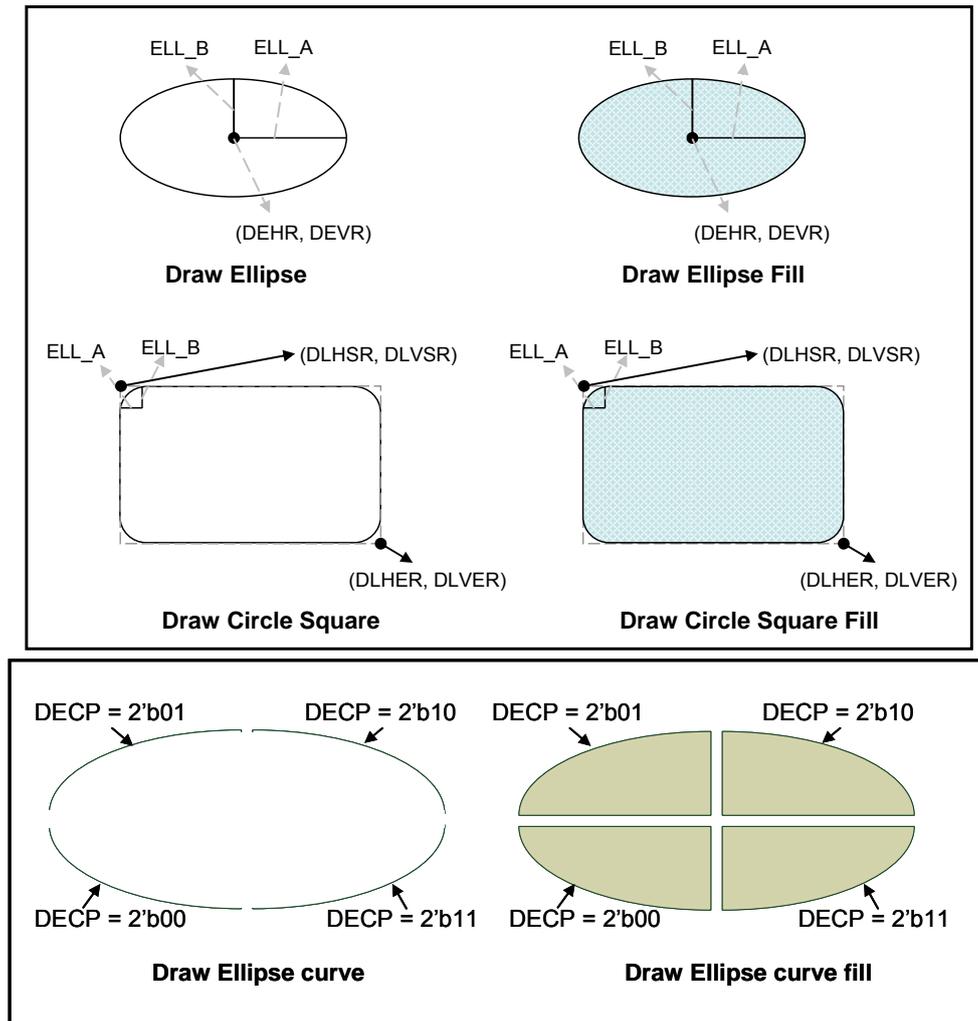


圖 5-7 : 繪圖功能

REG[A1h] Draw Ellipse/Circle Square Long axis Setting Register (ELL_A0)

Bit	說明	初始值	Access
7-0	畫橢圓/圓角方形長軸 [7:0]	0	RW

REG[A2h] Draw Ellipse/Circle Square Long axis Setting Register (ELL_A1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	畫橢圓/圓角方形長軸[9:8]	0	RW

REG[A3h] Draw Ellipse/Circle Square Short axis Setting Register (ELL_B0)

Bit	說明	初始值	Access
7-0	畫橢圓/圓角方形短軸[7:0]	0	RW

REG[A4h] Draw Ellipse/Circle Square Short axis Setting Register (ELL_B1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	畫橢圓/圓角方形短軸[8]	0	RW

REG[A5h] Draw Ellipse/Circle Square Center Horizontal Address Register0 (DEHR0)

Bit	說明	初始值	Access
7-0	畫橢圓/圓角方形中心點的水平位置 [7:0]	0	RW

REG[A6h] Draw Ellipse/Circle Square Center Horizontal Address Register1 (DEHR1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	畫橢圓/圓角方形中心點的水平位置	0	RW

REG[A7h] Draw Ellipse/Circle Square Center Vertical Address Register0 (DEVR0)

Bit	說明	初始值	Access
7-0	畫橢圓/圓角方形中心點的垂直位置[7:0]	0	RW

REG[A8h] Draw Ellipse/Circle Square Center Vertical Address Register1 (DEVR1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	畫橢圓/圓角方形中心點的垂直位置[8]	0	RW

REG[A9h] Draw Triangle Point 2 Horizontal Address Register0 (DTPH0)

Bit	說明	初始值	Access
7-0	畫三角形 2 點的水平位置 [7:0]	0	RW

REG[AAh] Draw Triangle Point 2 Horizontal Address Register1 (DTPH1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	畫三角形第 2 點的水平位置[9:8]	0	RW

REG[ABh] Draw Triangle Point 2 Vertical Address Register0 (DTPV0)

Bit	說明	初始值	Access
7-0	畫三角形第 2 點的垂直位置[7:0]	0	RW

REG[ACh] Draw Triangle Point 2 Vertical Address Register1 (DTPV1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	畫三角形第 2 點的垂直位置[8]	0	RW

5-12 直接記憶體存取 (DMA) 暫存器

REG[B0h] Source Starting Address REG0 (SSAR0)

Bit	說明	初始值	Access
7-0	DMA 來源開始位置[7:0]	0	RW

REG[B1h] Source Starting Address REG 1 (SSAR1)

Bit	說明	初始值	Access
7-0	DMA 來源開始位置[15:8]	0	RW

REG[B2h] Source Starting Address REG 2 (SSAR2)

Bit	說明	初始值	Access
7-0	DMA 來源開始位置[23:16]	0	RW

REG[B4h] Block Width REG 0(BWR0) / DMA Transfer Number REG 0 (DTNR0)

Bit	說明	初始值	Access
7-0	當暫存器 [BFh] Bit 1 為 0 (連續性模式) DMA 傳輸數量[7:0] 當暫存器[BFh] bit 1 為 1 (區塊模式) DMA 區塊寬度[7:0]	0	RW

REG[B5h] Block Width REG 1 (BWR1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	DMA 區塊寬度 [9:8]	0	RW

REG[B6h] Block Height REG 0(BHR0) /DMA Transfer Number REG 1 (DTNR1)

Bit	說明	初始值	Access
7-0	當暫存器[BFh] bit 1 為 0 (連續性模式) DMA 傳輸數量[15:8] 當暫存器[BFh] bit 1 為 1 (區塊模式) DMA 區塊寬度[7:0]	0	RW

REG[B7h] Block Height REG 1 (BHR1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	DMA 區塊高度[9:8]	0	RW

REG[B8h] Source Picture Width REG 0(SPWR0) / DMA Transfer Number REG 2(DTNR2)

Bit	說明	初始值	Access
7-3	DMA 來源圖片寬度 [7:3]	0	RW
2-0	當暫存器[BFh] bit 1 為 0 (連續性模式) DMA 傳輸數量[18:16] 當暫存器[BFh] bit 1 為 1 (區塊模式) DMA 來源圖片寬度[2:0]	0	RW

REG[B9h] Source Picture Width REG 1 (SPWR1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	DMA 來源圖片寬度[9:8]	0	RW

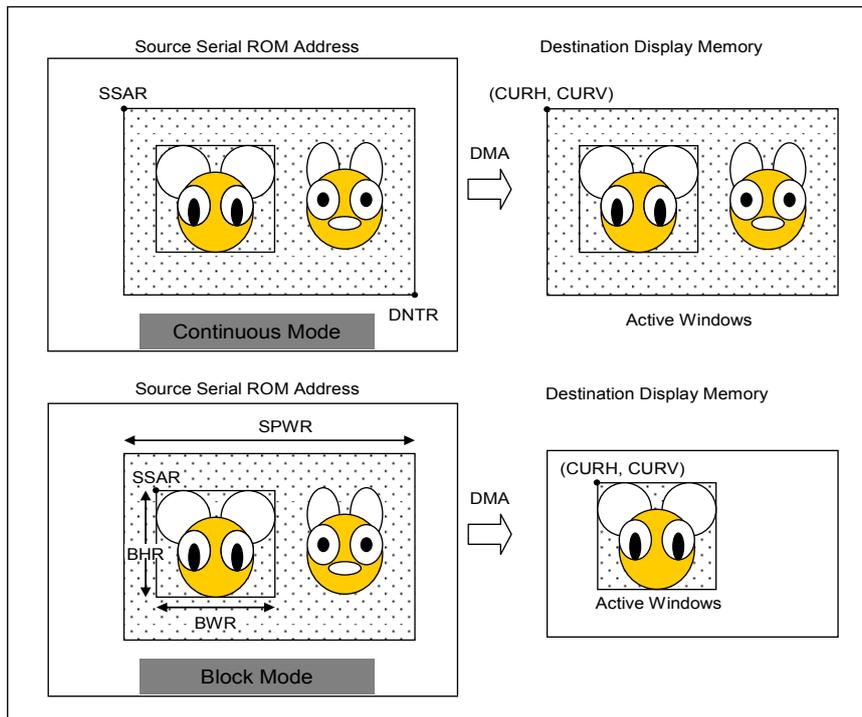


圖 5-8 : DMA 連續性模式與區塊模式

REG[BfH] DMA Configuration REG (DMACR)

Bit	說明	初始值	Access
7-2	NA	0	RO
1	選擇 DMA 連續性或區塊模式的讀取/寫入位元 0: 連續性模式 / 1: 區塊模式。	0	RW
0	寫入功能 → DMA 起始位元 自動地透過 MCU 設定 1 與重設 0。 讀取功能 → DMA 忙碌確認位元 0: 閒置狀態 / 1: 忙碌狀態。	0	RW

5-13 鍵盤掃描與 IO 控制暫存器

REG [C0h] Key-Scan Control Register 1 (KSCR1)

Bit	說明	初始值	Access																																																												
7	設定鍵盤掃描功能開啟位元 (KEY_EN) 1: 開啟。 0: 關閉。	0	R/W																																																												
6	設定長按鍵開啟位元 1: 開啟，長按鍵週期由 KSCR2 Bit4-2 設定。 0: 關閉。	0	RW																																																												
5-4	設定鍵盤掃描資料的取樣次數 鍵盤掃描機制的「消除機械彈跳」次數。 00b: 4 次。 01b: 8 次。 10b: 16 次。 11b: 32 次。	0	R/W																																																												
3	NA	0	RO																																																												
2-0	KF2-0: 鍵盤頻率 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>KF2</th> <th>KF1</th> <th>KF0</th> <th colspan="3">Key-Scan Cycle (4x5)</th> </tr> <tr> <th colspan="3">System Clock</th> <th>20MHz</th> <th>40MHz</th> <th>60MHz</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>128µs</td> <td>64µs</td> <td>42.67us</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>256µs</td> <td>128µs</td> <td>85.33µs</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>512µs</td> <td>256µs</td> <td>170.67µs</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1.024ms</td> <td>512µs</td> <td>341.33µs</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>2.048ms</td> <td>1.024ms</td> <td>682.67us</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>4.096ms</td> <td>2.048ms</td> <td>1.365ms</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>8.192ms</td> <td>4.096ms</td> <td>2.731ms</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>16.384ms</td> <td>8.192ms</td> <td>5.461ms</td> </tr> </tbody> </table>	KF2	KF1	KF0	Key-Scan Cycle (4x5)			System Clock			20MHz	40MHz	60MHz	0	0	0	128µs	64µs	42.67us	0	0	1	256µs	128µs	85.33µs	0	1	0	512µs	256µs	170.67µs	0	1	1	1.024ms	512µs	341.33µs	1	0	0	2.048ms	1.024ms	682.67us	1	0	1	4.096ms	2.048ms	1.365ms	1	1	0	8.192ms	4.096ms	2.731ms	1	1	1	16.384ms	8.192ms	5.461ms	0	R/W
KF2	KF1	KF0	Key-Scan Cycle (4x5)																																																												
System Clock			20MHz	40MHz	60MHz																																																										
0	0	0	128µs	64µs	42.67us																																																										
0	0	1	256µs	128µs	85.33µs																																																										
0	1	0	512µs	256µs	170.67µs																																																										
0	1	1	1.024ms	512µs	341.33µs																																																										
1	0	0	2.048ms	1.024ms	682.67us																																																										
1	0	1	4.096ms	2.048ms	1.365ms																																																										
1	1	0	8.192ms	4.096ms	2.731ms																																																										
1	1	1	16.384ms	8.192ms	5.461ms																																																										

REG [C1h] Key-Scan Controller Register 2 (KSCR2)

Bit	說明	初始值	Access																				
7	設定鍵盤掃描喚醒功能位元 0: 關閉鍵盤喚醒功能。 1: 開啟鍵盤喚醒功能。	0	R/W																				
6-4	NA	0	RO																				
3-2	長按鍵時間調整 <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>System Clock</th> <th>20MHz</th> <th>40MHz</th> <th>60MHz</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1.25 sec</td> <td>0.625 sec</td> <td>0.3125 sec</td> </tr> <tr> <td>01b</td> <td>2.5 sec</td> <td>1.25 sec</td> <td>0.625 sec</td> </tr> <tr> <td>10b</td> <td>3.75 sec</td> <td>1.875 sec</td> <td>0.9375 sec</td> </tr> <tr> <td>11b</td> <td>5 sec</td> <td>2.5 sec</td> <td>1.25 sec</td> </tr> </tbody> </table>	System Clock	20MHz	40MHz	60MHz	00b	1.25 sec	0.625 sec	0.3125 sec	01b	2.5 sec	1.25 sec	0.625 sec	10b	3.75 sec	1.875 sec	0.9375 sec	11b	5 sec	2.5 sec	1.25 sec	0	R/W
System Clock	20MHz	40MHz	60MHz																				
00b	1.25 sec	0.625 sec	0.3125 sec																				
01b	2.5 sec	1.25 sec	0.625 sec																				
10b	3.75 sec	1.875 sec	0.9375 sec																				
11b	5 sec	2.5 sec	1.25 sec																				

1-0	<p>被按的按鍵數目</p> <p>00b : 沒有鍵盤被按壓到。</p> <p>01b : 按到 1 個按鍵，讀取 REG[C2h] 來獲取按鍵值 (Key Code)。</p> <p>10b : 按到 2 個按鍵，讀取 REG[C2h ~ C3h] 來獲取按鍵值。</p> <p>11b : 按到 3 個按鍵，讀取 REG[C2h ~ C4h] 來獲取按鍵值。</p>	0	RO
-----	--	---	----

REG [C2h] Key-Scan Data Register (KSDR0)

Bit	說明	初始值	Access
7-0	<p>按鍵擷取資料#0 (Key Strobe Data0)</p> <p>請參考章節 7-9 的詳細說明。</p>	NA	RO

REG [C3h] Key-Scan Data Register (KSDR1)

Bit	說明	初始值	Access
7-0	<p>按鍵擷取資料#1</p> <p>請參考章節 7-9 的詳細說明。</p>	NA	RO

REG [C4h] Key-Scan Data Register (KSDR2)

Bit	說明	初始值	Access
7-0	<p>按鍵擷取資料#2</p> <p>請參考章節 7-9 的詳細說明。</p>	NA	RO

REG[C7h] Extra General Purpose IO Register (GPIOX)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	<p>GPIX/GPOX 資料位元</p> <p>讀取: 從 GPIX 腳位輸入資料。</p> <p>寫入: 輸出資料到 GPOX 腳位。</p>	NA	RW

5-14 浮動視窗控制暫存器

REG [D0h] Floating Windows Start Address XA 0 (FWSAXA0)

Bit	說明	初始值	Access
7-0	浮動視窗起始位置 XA [7:0]	0	RW

REG [D1h] Floating Windows Start Address XA 1 (FWSAXA1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	浮動視窗起始位置 XA [9:8]	0	RW

REG [D2h] Floating Windows Start Address YA 0 (FWSAYA0)

Bit	說明	初始值	Access
7-0	浮動視窗起始位置 YA [7:0]	0	RW

REG [D3h] Floating Windows Start Address YA 1 (FWSAYA1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	浮動視窗起始位置 YA [8]	0	RW

REG [D4h] Floating Windows Width 0 (FWW0)

Bit	說明	初始值	Access
7-0	浮動視窗寬度設定[7:0]	0	RW

REG [D5h] Floating Windows Width 1 (FWW1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	浮動視窗寬度設定[9:8]	0	RW

REG [D6h] Floating Windows Height 0 (FWH0)

Bit	說明	初始值	Access
7-0	浮動視窗高度設定[7:0]	0	RW

REG [D7h] Floating Windows Height 1 (FWH1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	浮動視窗高度設定[9:8]	0	RW

REG [D8h] Floating Windows Display X Address 0 (FWDXA0)

Bit	說明	初始值	Access
7-0	浮動視窗顯示 X 軸位置[7:0]	0	RW

REG [D9h] Floating Windows Display X Address 1 (FWDXA1)

Bit	說明	初始值	Access
7-2	NA	0	RO
1-0	浮動視窗顯示 X 軸位置[9:8]	0	RW

REG [DAh] Floating Windows Display Y Address 0 (FWDYA0)

Bit	說明	初始值	Access
7-0	浮動視窗顯示 X 軸位置[7:0]	0	RW

REG [DBh] Floating Windows Display Y Address 1 (FWDYA1)

Bit	說明	初始值	Access
7-1	NA	0	RO
0	浮動視窗顯示 Y 軸位置[8]	0	RW

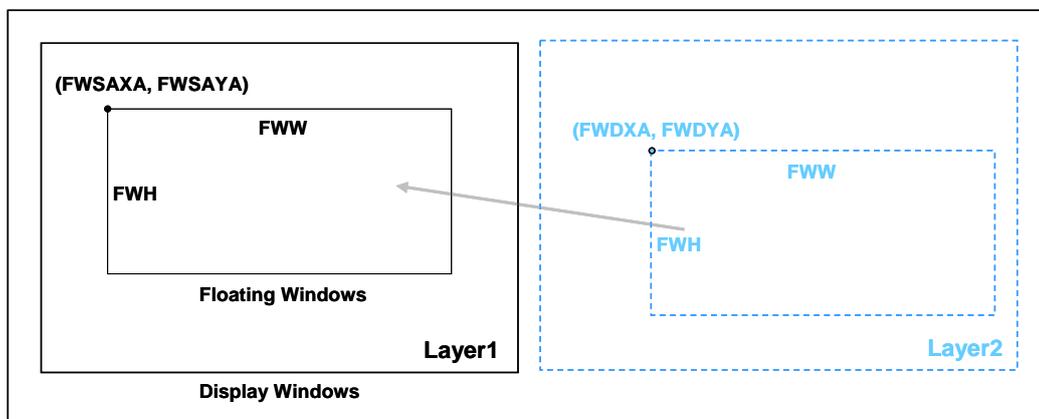


圖 5-9 : 浮動視窗

5-15 串列式Flash控制暫存器

SACS_MODE REG [E0h] Serial Flash/ROM Direct Access Mode

Bit	說明	初始值	Access
7-1	NA	0	RO
0	0: 關閉直接存取模式，此時使用者用 FONT/DMA 模式會關閉。 1: 開啟直接存取模式，此時 FONT/DMA 模式會被關閉。	0	RW

SACS_ADDR REG [E1h] Serial Flash/ROM Direct Access Mode Address

Bit	說明	初始值	Access
7-0	直接存取模式定址 串列式 Flash/ROM 是 24 位元的定址方式，因此使用者在定址時，必須將位址資料連續寫入 REG[E1h] 3 次。	0	WO

SACS_DATA [E2h] Serial Flash/ROM Direct Access Data Read

Bit	說明	初始值	Access
7-0	直接存取模式讀取資料緩衝區。	0	RO

5-16 中斷控制

REG[F0h] Interrupt Control Register1 (INTC1)

Bit	說明	初始值	Access
7-5	NA	0	RO
4	開啟鍵盤掃描中斷位元 0：關閉鍵盤中斷。 1：開啟鍵盤中斷。	0	RW
3	開啟 DMA 中斷位元 0：關閉 DMA 中斷。 1：開啟 DMA 中斷。	0	RW
2	開啟觸控面板中斷位元 0：關閉觸控中斷。 1：開啟觸控中斷。	0	RW
1	開啟 BTE 程序 (BTE Process) 完成的中斷位元 0：關閉 BTE 程序完成的中斷。 1：開啟 BTE 程序完成的中斷。	0	RW
0	當 BTE 選擇 MCU 相關的操作且 BTE 功能為開啟時 (REG[50h] Bit7 = 1)，此位元被用在開啟 MCU 讀取/寫入的 BTE 中斷功能： 0：關閉 MCU 讀取/寫入的 BTE 中斷。 1：開啟 MCU 讀取/寫入的 BTE 中斷。 當關閉 BTE 功能時，此位元被用在開啟文字寫入的中斷功能(*)： 0：關閉文字寫入的中斷。 1：開啟文字寫入的中斷。	0	RW

註：

- MCU 相關的 BTE 操作包含：「BTE 寫入搭配光柵運算」、「BTE 讀取」、「BTE 通透性寫入」、「顏色擴充」以及「通透性顏色擴充」功能。
- 文字寫入中斷代表已完成文字字體寫入 DDRAM 中。

REG[F1h] Interrupt Control Register2 (INTC2)

Bit	說明	初始值	Access
7-5	NA	0	RO
4	<p>寫入功能 → 鍵盤掃描中斷清除位元</p> <p>0: 未操作。</p> <p>1: 清除鍵盤掃描中斷。</p> <p>讀取功能 → 鍵盤掃描中斷狀態</p> <p>0: 未發生鍵盤掃描中斷。</p> <p>1: 發生鍵盤掃描中斷。</p>	0	RW
3	<p>寫入功能 → DMA 中斷清除位元</p> <p>0: 未操作。</p> <p>1: 清除 DMA 中斷功能。</p> <p>讀取功能 → DMA 中斷狀態</p> <p>0: 未發生 DMA 中斷。</p> <p>1: 發生 DMA 中斷。</p>	0	RW
2	<p>寫入功能 → 觸控面板中斷清除位元</p> <p>0: 未操作。</p> <p>1: 清除觸控面板中斷。</p> <p>讀取功能 → 觸控面板中斷狀態</p> <p>0: 未發生觸控面板中斷。</p> <p>1: 發生觸控面板中斷。</p>	0	RW
1	<p>寫入功能 → BTE 程序完成中斷清除位元</p> <p>0: 未操作。</p> <p>1: 清除 BTE 程序完成中斷。</p> <p>讀取功能 → BTE 中斷狀態</p> <p>0: 未發生 BTE 程序完成中斷。</p> <p>1: 發生 BTE 程序完成中斷。</p>	0	RW
0	<p>當 BTE 選擇 MCU 相關的操作且開啟 BTE 功能 (REG[50h] Bit7 = 1)</p> <p>寫入功能 → BTE 讀取/寫入中斷清除</p> <p>0: 未操作。</p> <p>1: 清除 MCU 寫入/讀取的 BTE 中斷。</p> <p>讀取功能 → BTE R/W 中斷狀態</p> <p>0: 未發生 BTE MCU 讀/寫中斷。</p> <p>1: 發生 BTE MCU 讀/寫中斷。</p> <p>當關閉 BTE 功能時，且開啟文字模式時：</p> <p>寫入功能 → 開啟文字寫入中斷(*)位元</p>	0	RW

	0: 未操作。 1: 清除文字寫入中斷。 讀取功能 → 文字寫入中斷狀態 0: 未發生文字寫入中斷。 1: 發生文字寫入中斷。		
--	--	--	--

註：

1. MCU 相關的 BTE 操作包含：「BTE 寫入搭配光柵運算」、「BTE 讀取」、「BTE 通透性寫入」、「顏色擴充」以及「通透性顏色擴充」功能。
2. 字體寫入中斷代表已完成文字字體寫入 DDRAM 中。

6. 硬體介面

6-1 MCU 介面

RA8875 支援 8080 和 6800 等兩種微處理機介面傳輸模式。介面的選擇決定於 IC 接腳 "C86" 的電位，當選擇 8080 來進行介面傳輸時，"C86" 須連接到低電位；若選擇 6800 來進行介面傳輸時，"C86" 必須連接到高電位，如下圖 6-1 與圖 6-2 所示。

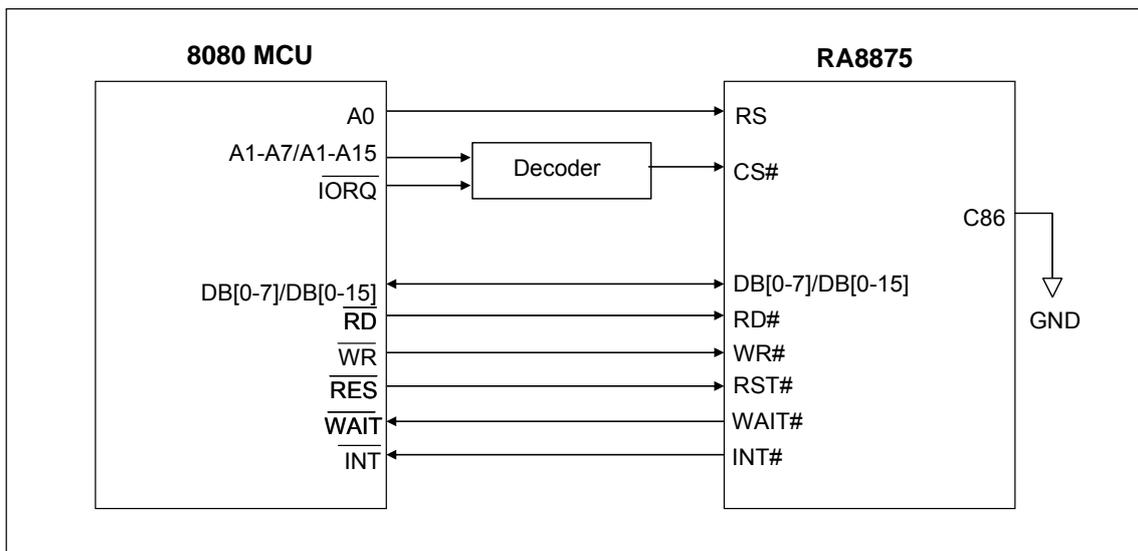


圖 6-1 : 8080 MCU 介面

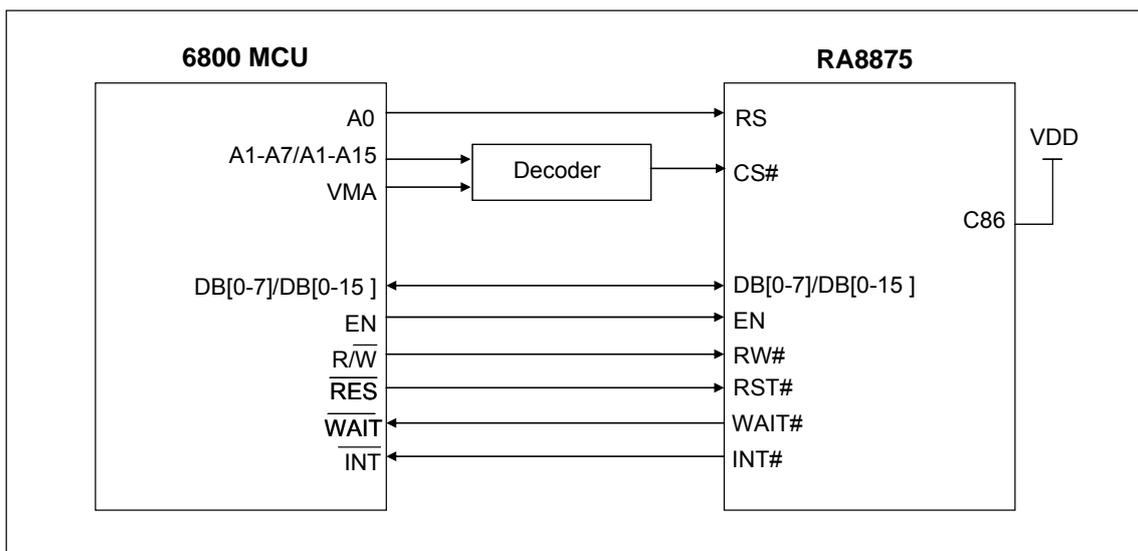


圖 6-2 : 6800 MCU 介面

6-1-1 MCU傳輸協定

6-1-1-1 並列式介面的傳輸協定

下面圖 6-3、表 6-1、圖 6-4、表 6-2 是RA8875 支援 8080 和 6800 兩種微處理機介面的傳輸協定與時序參數表。

6800 – 8/16-bit Interface

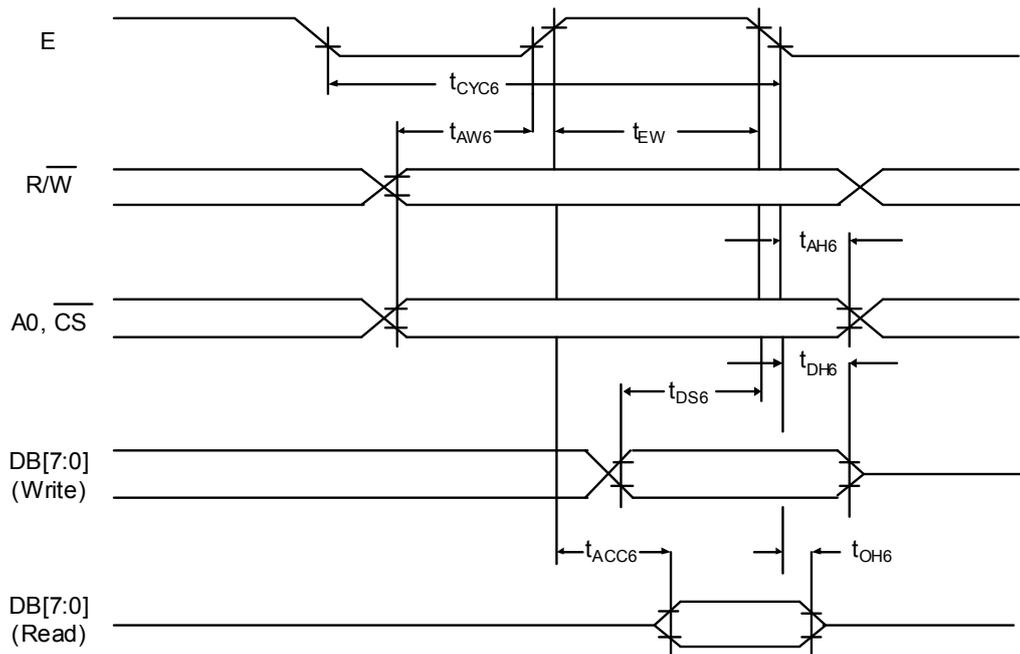


圖 6-3 : 6800 MCU 傳輸協定

表 6-1 : 6800 MCU 介面時序參數

Symbol	Parameter	Rating		Unit	Symbol
		Min.	Max.		
t_{CYC6}	Cycle time	50	--	ns	tc is one system clock period: tc = 1/SYS_CLK
t_{EW}	Strobe Pulse width	20	--	ns	
t_{AW6}	Address setup time	0	--	ns	
t_{AH6}	Address hold time	10	--	ns	
t_{DS6}	Data setup time	20	--	ns	
t_{DH6}	Data hold time	10	--	ns	
t_{ACC6}	Data output access time	0	20	ns	
t_{OH6}	Data output hold time	0	20	ns	

8080 – 8/16-bit Interface

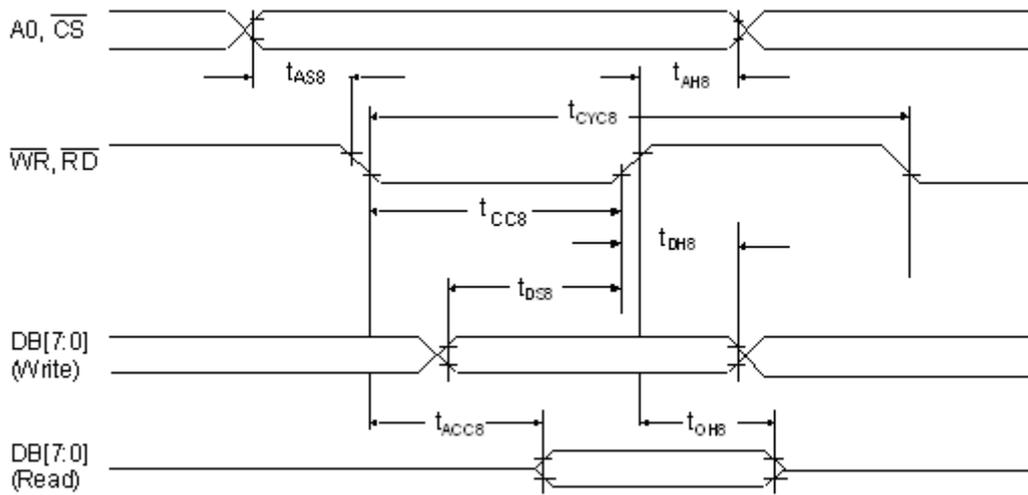


圖 6-4 : 8080 MCU 傳輸協定

表 6-2 : 8080 MCU 介面時序參數

Symbol	Parameter	Rating		Unit	Symbol
		Min.	Max.		
t_{CYC8}	Cycle time	50	--	ns	tc is one system clock period: tc = 1/SYS_CLK
t_{CC8}	Strobe Pulse width	20	--	ns	
t_{AS8}	Address setup time	0	--	ns	
t_{AH8}	Address hold time	10	--	ns	
t_{DS8}	Data setup time	20	--	ns	
t_{DH8}	Data hold time	10	--	ns	
t_{ACC8}	Data output access time	0	20	ns	
t_{OH8}	Data output hold time	0	20	ns	

RA8875 也可以透過暫存器 SYSR 的 Bit [1:0] 來決定 MCU 資料匯流排的寬度，當 Bit [1:0] = 00 時，資料匯流排的寬度為 8 位元，當 Bit [1:0] = 11 時，資料匯流排的寬度為 16 位元。無論在 8080 或 6800 的微處理機介面，資料匯流排的寬度設定皆相同。一旦選擇 8 位元來進行傳輸時，傳輸時間約將增加一倍，而且要注意所有暫存器的讀寫都是固定以 8 位元來進行傳輸。

連續性資料寫入速度決定了顯示更新速度。傳輸週期的間隔必須大於系統頻率週期的 5 倍。若超過規格可能會導致資料遺失或功能失效，請參考圖 6-5 及圖 6-6 的說明。

在許多工業控制的場合，各種電器的干擾源比較強，為了減輕這些干擾源對 MCU 與 RA8875 間的傳輸影響，可以在 RA8875 的 CS#、RD#、WR# 端加一小電容到 GND。如果 MCU 與 RA8875 間的傳輸是使用 Cable 線，則 Cable 線長度必須小於 20cm，如果超過建議 CS#、RD#、WR#、RS 等信號必須加上 1~10Kohm 的 pull-up 電阻。

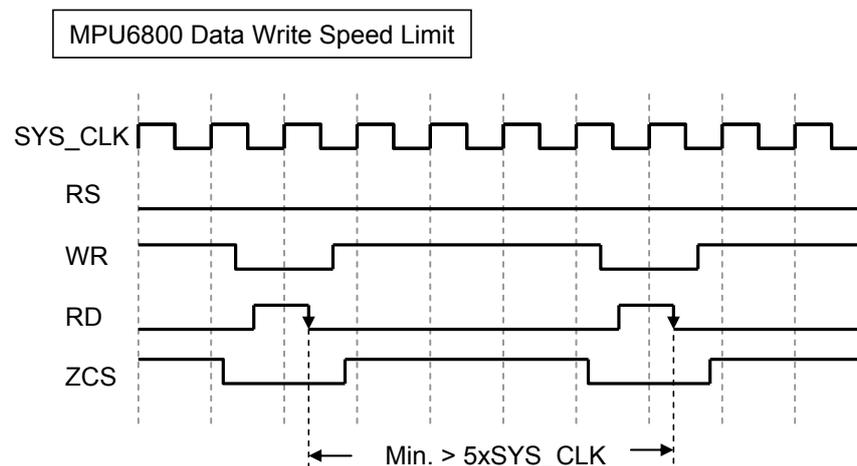


圖 6-5 : 6800 介面連續性資料寫入週期圖

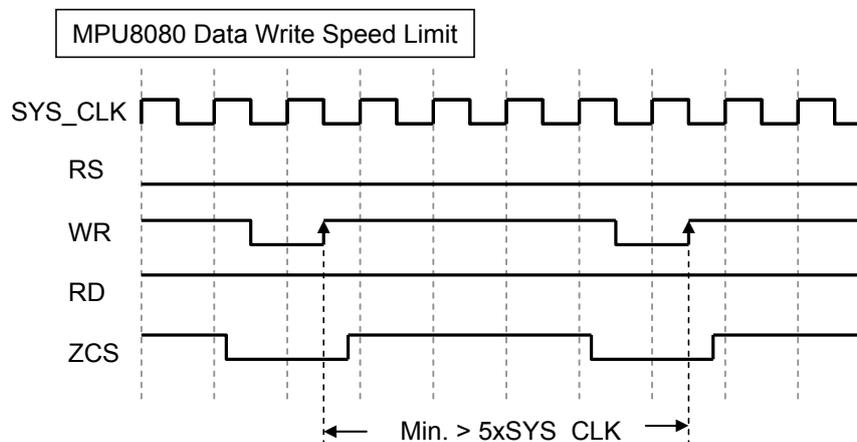


圖 6-6 : 8080 介面連續性資料寫入週期圖

6-1-2 串列式介面的協定

6-1-2-1 3-Wire SPI 介面

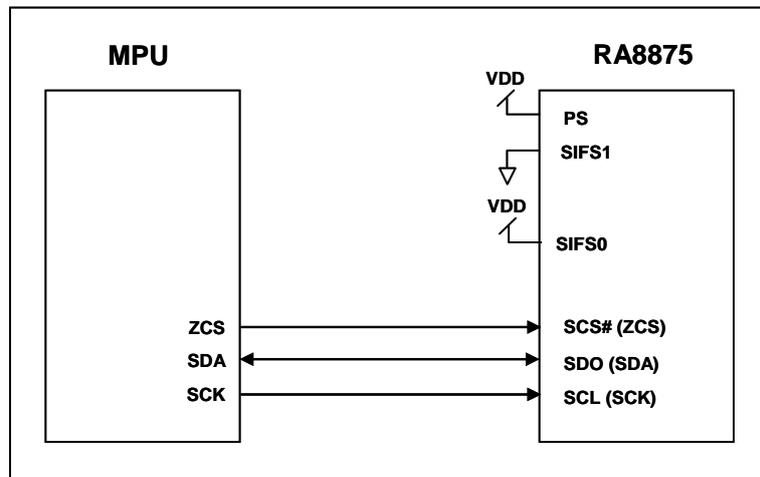


圖 6-7 : 3-Wire SPI 的 MCU 介面圖

RA8875 提供一個 SPI 從屬(Slave) 控制器，3-Wire SPI 寫的最大時脈速度是系統頻率的 1/3 (換言之，SPI 時脈高準位必須大於 1.5 個系統頻率)，而 3-Wire SPI 讀的最大時脈速度是系統頻率的 1/6。SPI 是由晶片選擇線 (ZCS)、串列傳輸時脈線 (SCK) 以及串列資料輸入/輸出線 (SDA) 所組成的。當 ZCS 是動作時，SCK 是由主要控制器(Master)所驅動的，用來閃鎖 SDA 的信號。使用 SPI 進行通訊時，通過對資料的第一個位元組的 MSB 2 Bits 可以設定目前的週期為指令/資料寫入模式，或是狀態位元/資料讀出的模式。在通訊的過程中，ZCS 必須要一直保持在低電位狀態，直到通訊結束。

當SPI 在指令/資料寫入模式時 (圖 6-8、圖 6-10)，此時傳輸的第 2 位元組為透過SPI的SDA腳位，由主要 (Master) 控制器端提供寫入資料。當SPI 在狀態位元/資料讀取模式時 (圖 6-9、圖 6-11)，第 2 位元組的資料讀取是由則RA8875 的SPI 從屬(Slave)控制器根據SCL的動作透過SDA傳送至主要 (Master) 控制器端。請參考 圖 6-8 ~ 圖 6-11 的說明。

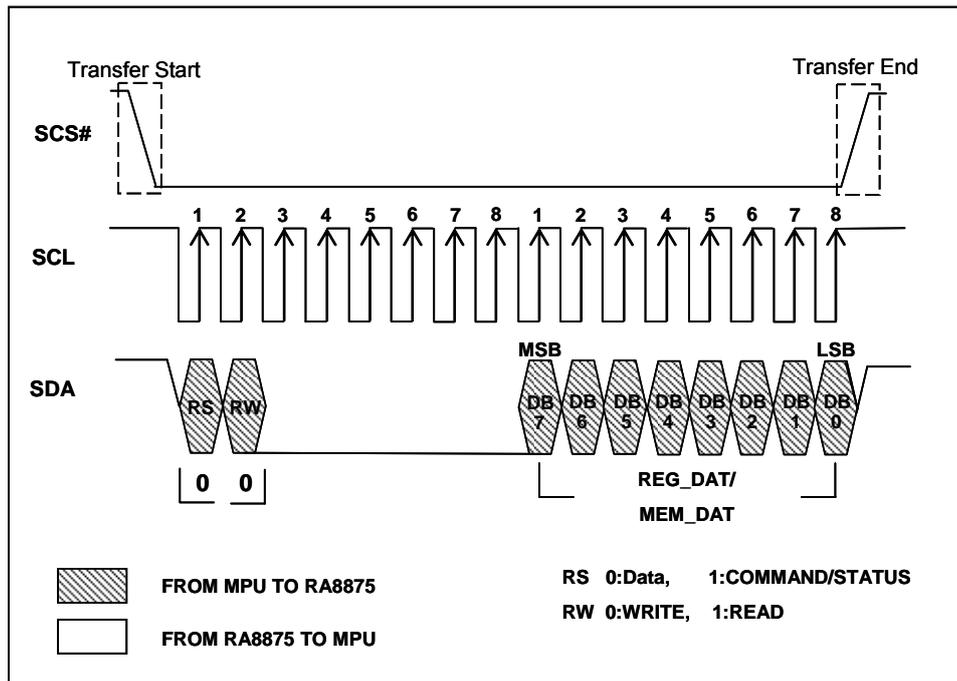


圖 6-8 : 3-Wire SPI 資料匯流排的資料寫入

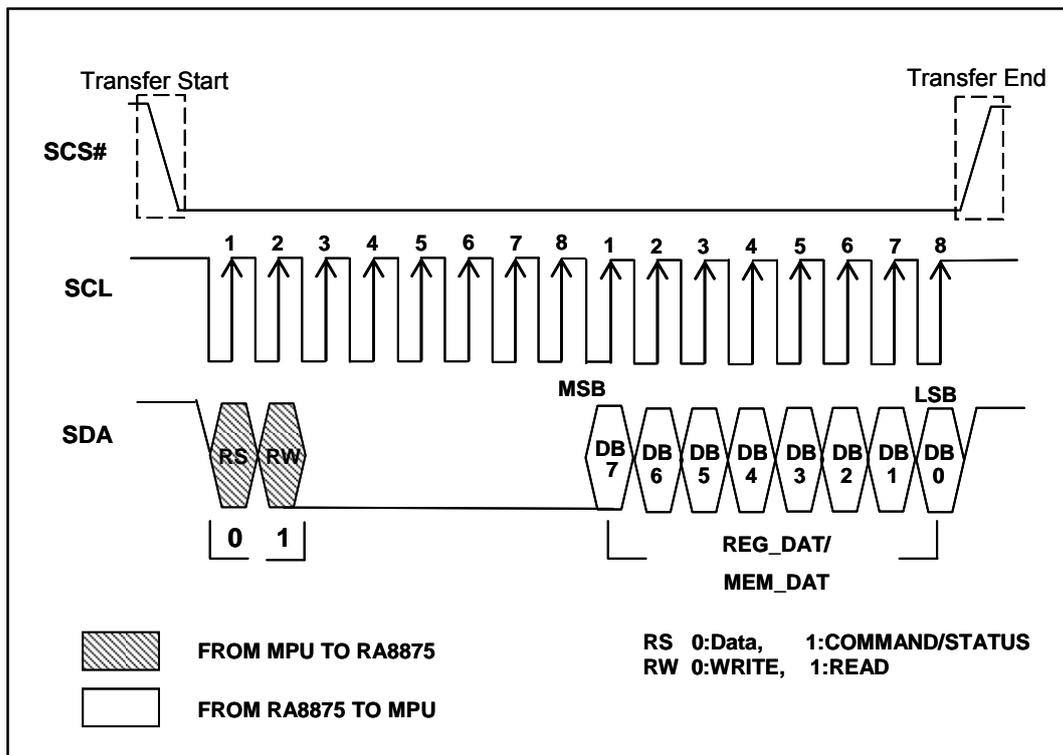


圖 6-9 : 3-Wire SPI 資料匯流排的資料讀取

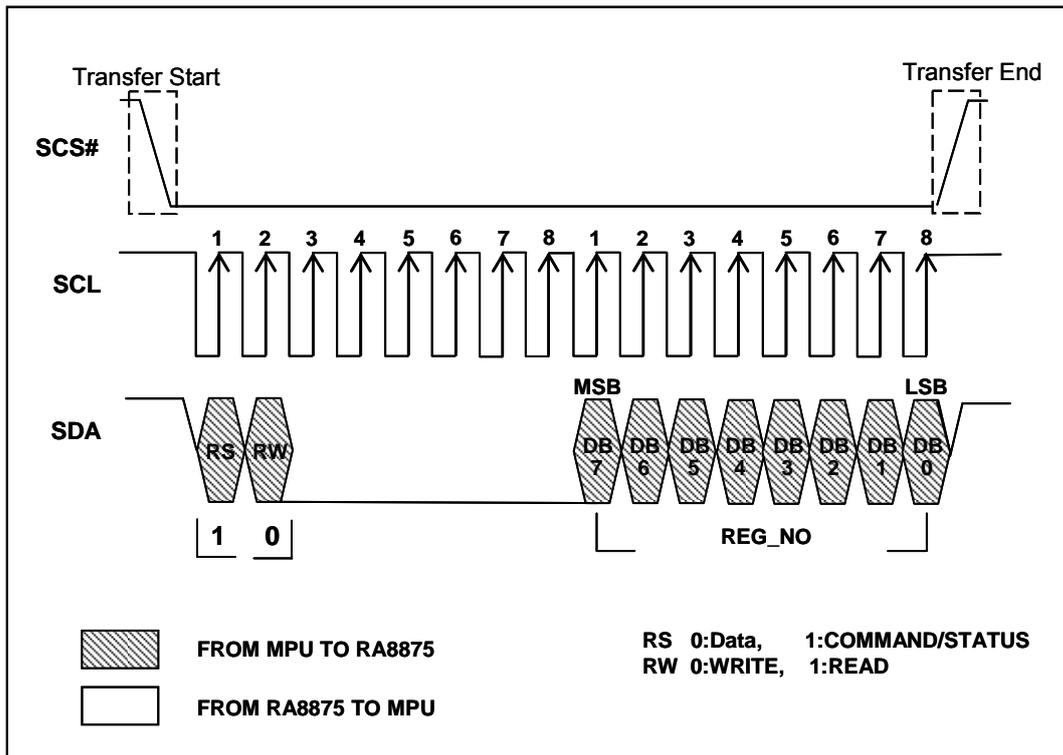


圖 6-10 : 3-Wire SPI 資料匯流排的指令寫入

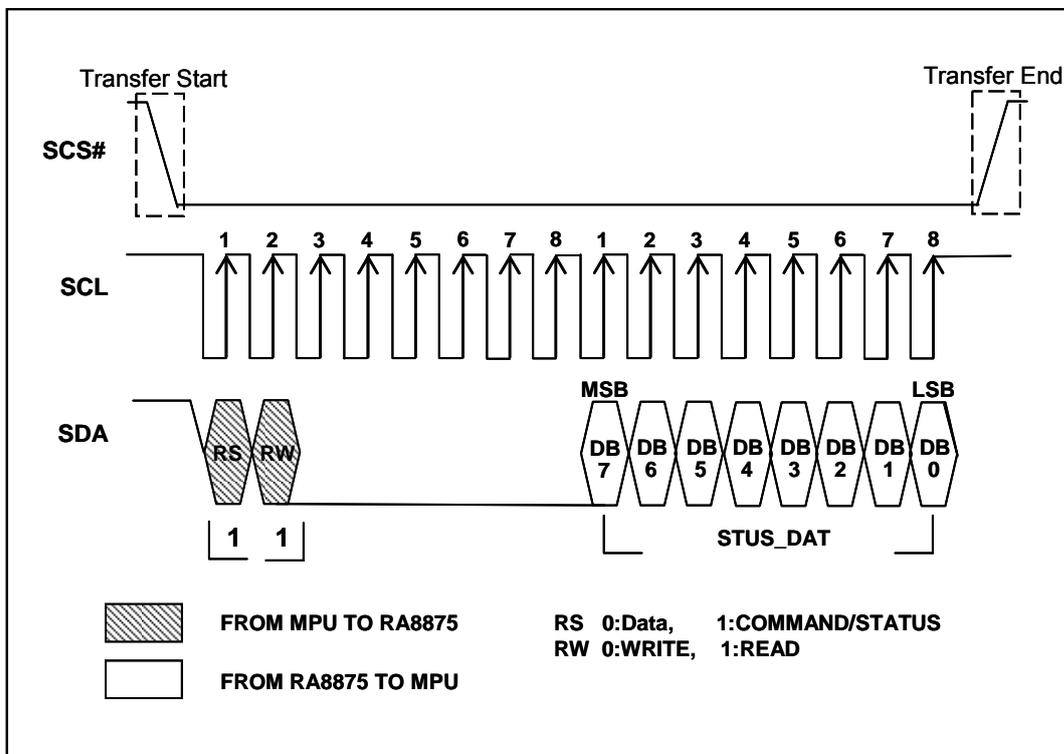


圖 6-11 : 3-Wire SPI 資料匯流排的狀態讀取

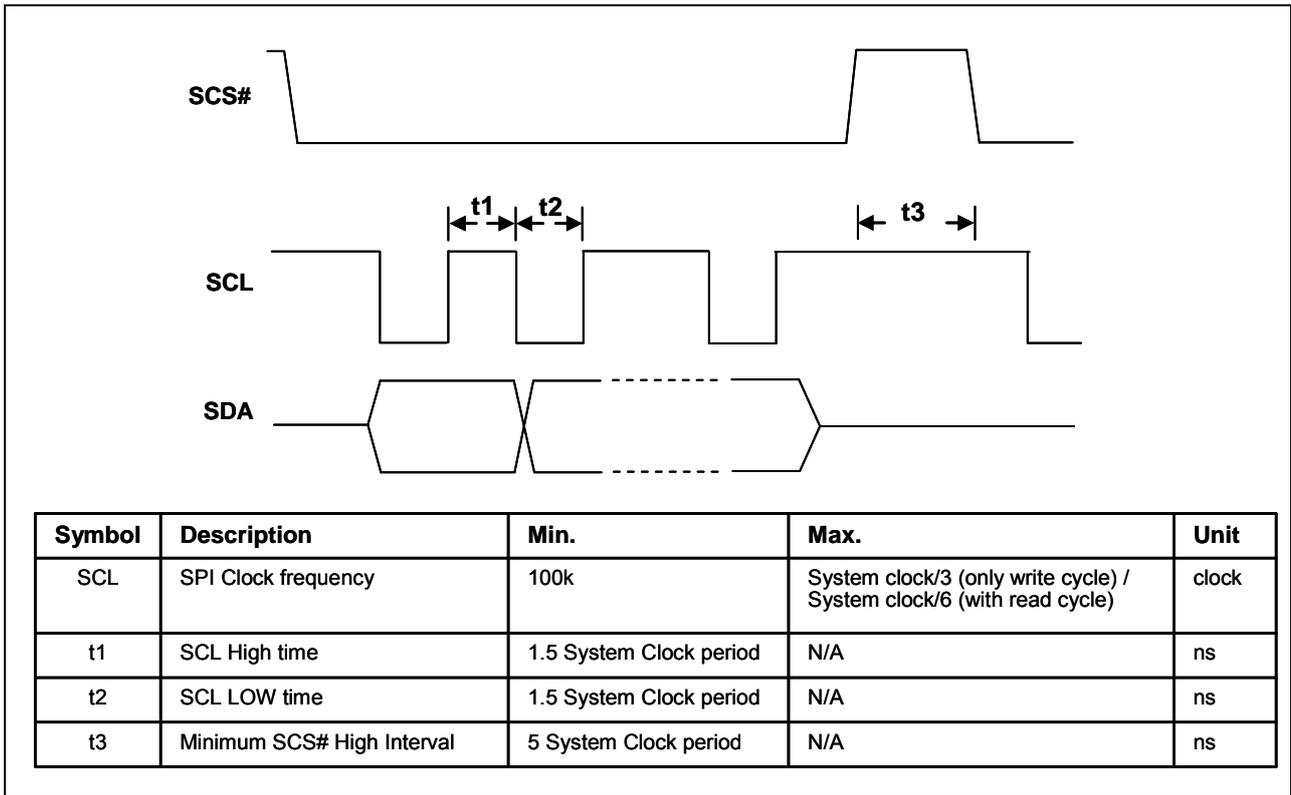


圖 6-12 : 3-Wire 的 SPI 時序

6-1-2-2 4-Wire SPI 介面

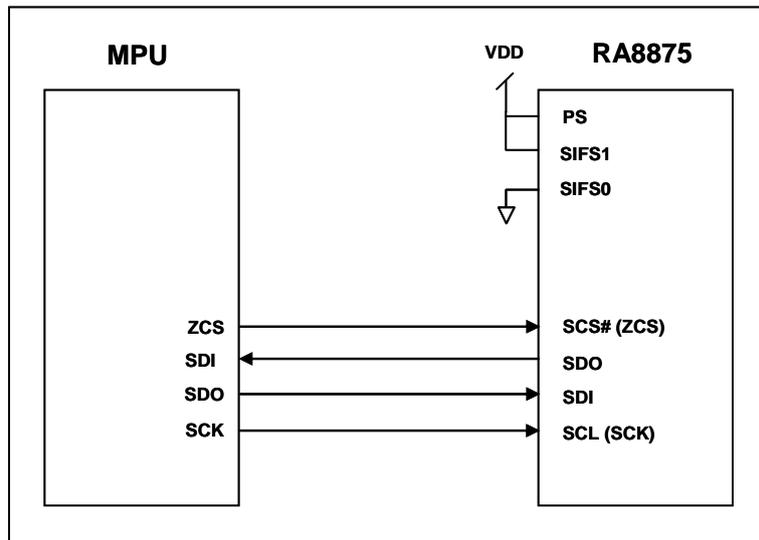


圖 6-13 : 4-Wire SPI MCU 介面圖

4-wire SPI介面與 3-wire SPI介面類似，唯一不同的是資料信號。在 3-wire SPI介面中，雙向的SDA信號用來當作資料信號且從屬(Slave)/ 主要(Master)皆可驅動。在 4-wire SPI介面中，SDA信號功能被區分為SDI與SDO信號。SDI是由SPI master驅動的資料腳位；SDO則是來自SPI從屬(Slave)端的資料輸出。4-Wire SPI寫的最大時脈速度是系統頻率的 1/3 (換言之，SPI時脈高準位必須大於 1.5 個系統頻率)，而 4-Wire SPI讀的最大時脈速度是系統頻率的 1/6。相關的說明請參考 圖 6-14 ~ 圖 6-17。

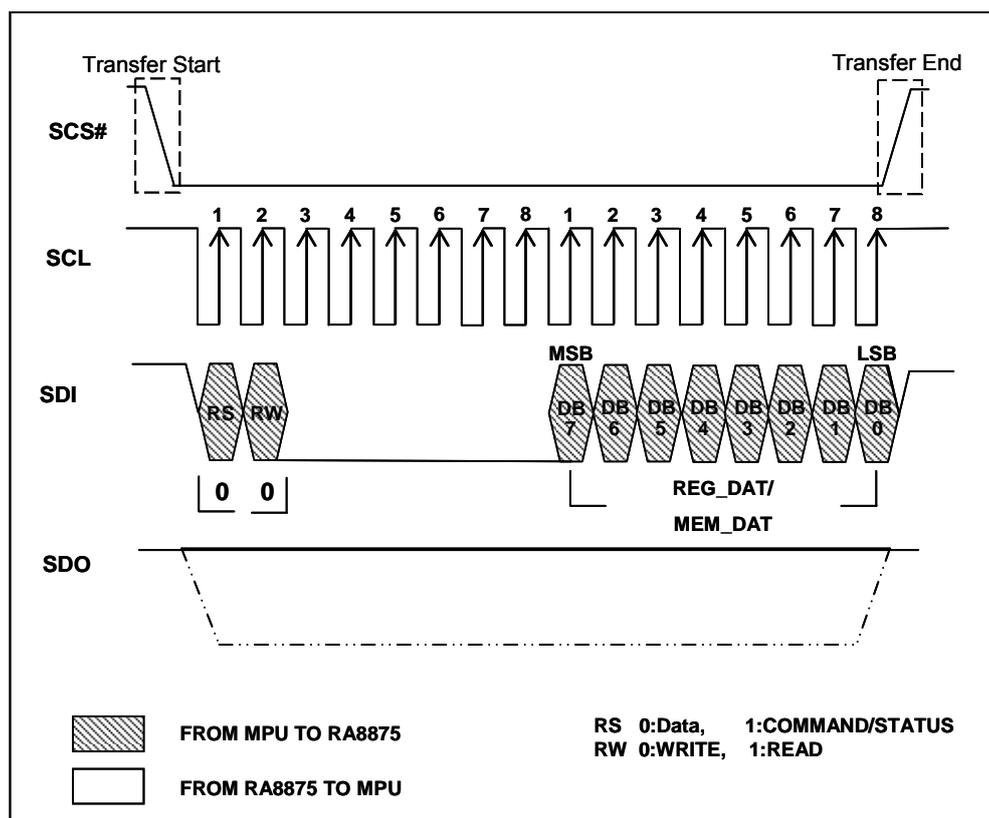


圖 6-14 : 4-Wire SPI 資料匯流排的資料寫入

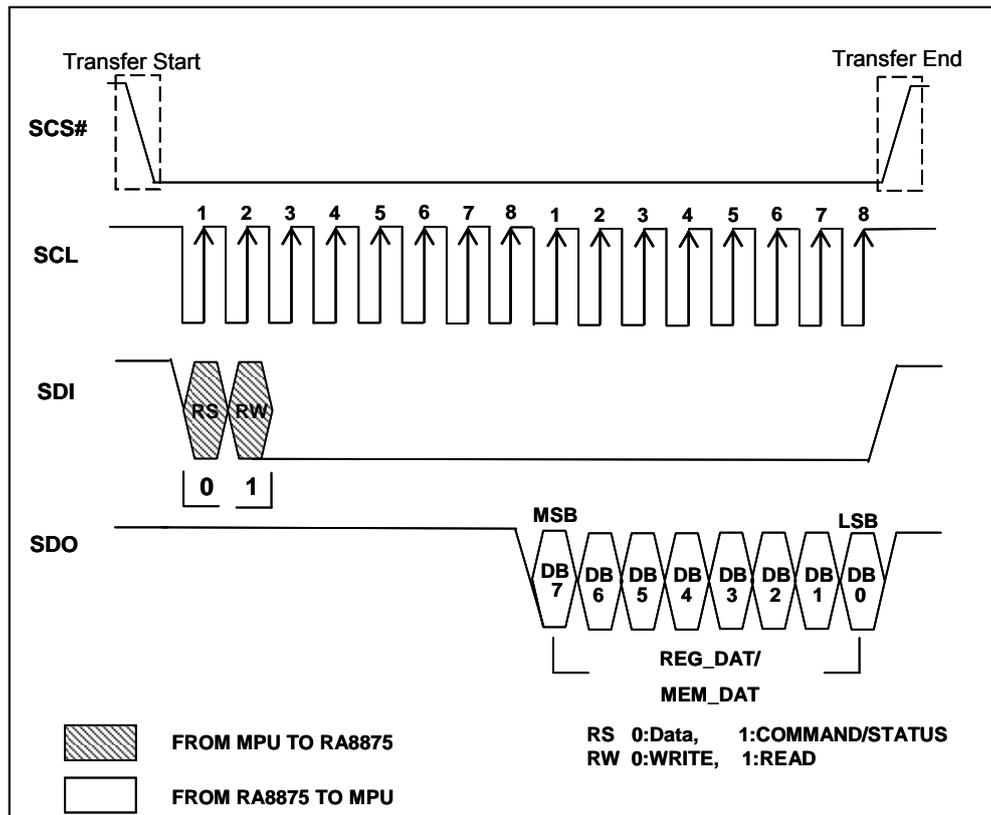


圖 6-15 : 4-Wire SPI 資料匯流排的資料讀取

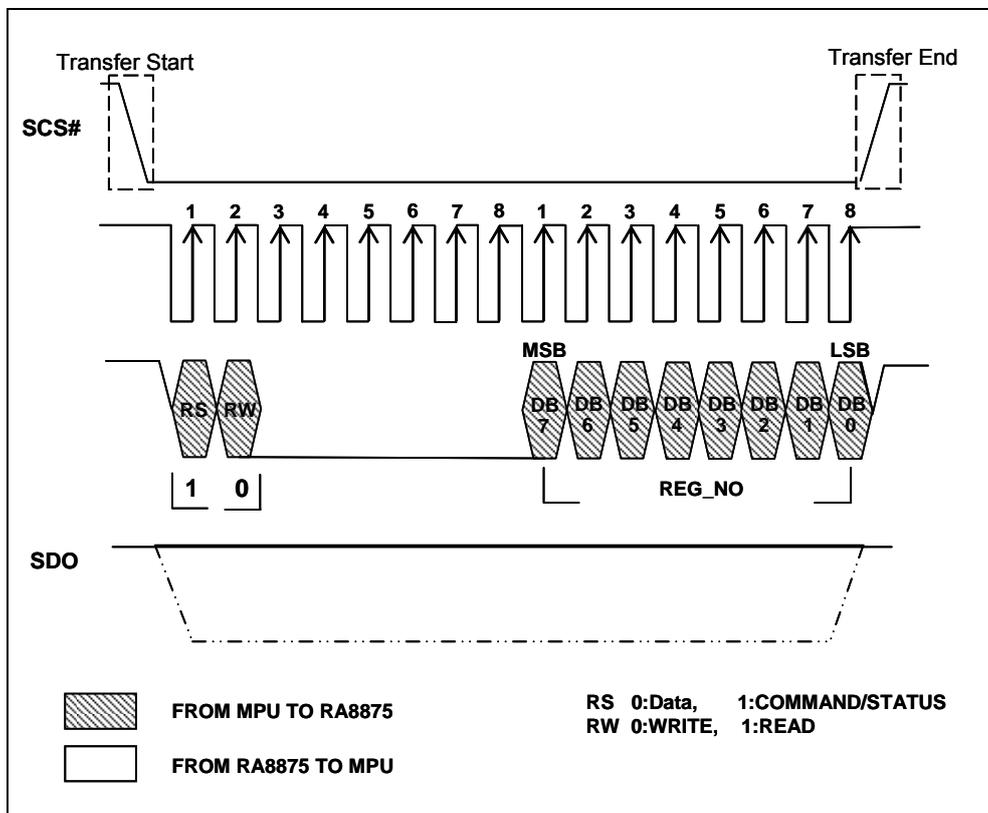


圖 6-16 : 4-Wire SPI 資料匯流排的指令寫入

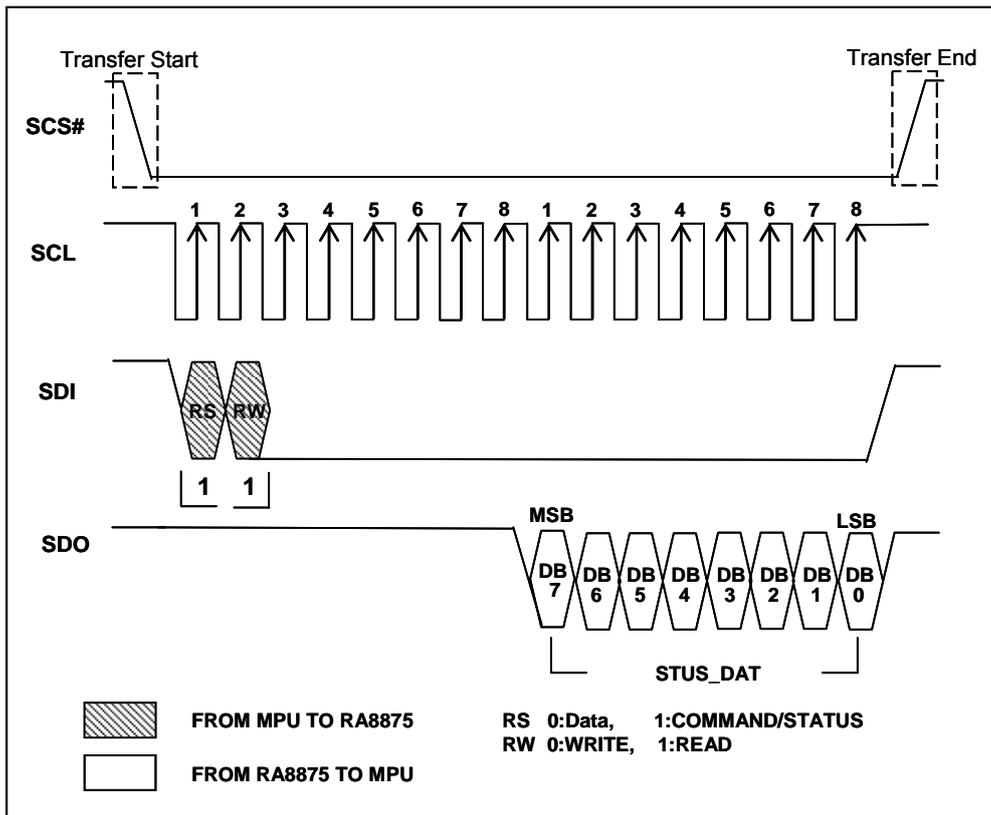


圖 6-17 : 4-Wire SPI 資料匯流排的狀態讀取

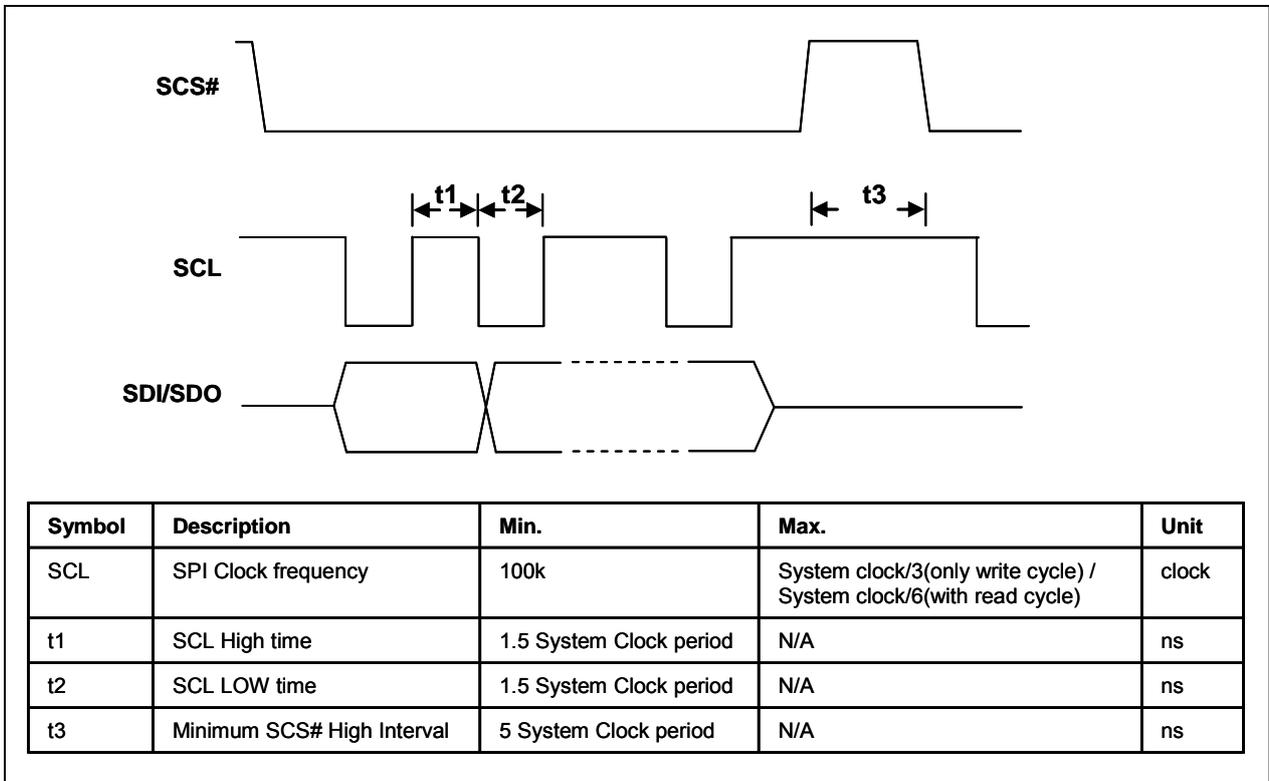


圖 6-18 : 4-Wire 的 SPI 時序

6-1-2-3 SPI 睡眠 / 喚醒

SPI 介面有些預設的環境限制，請參考以下流程來使用睡眠/喚醒功能。

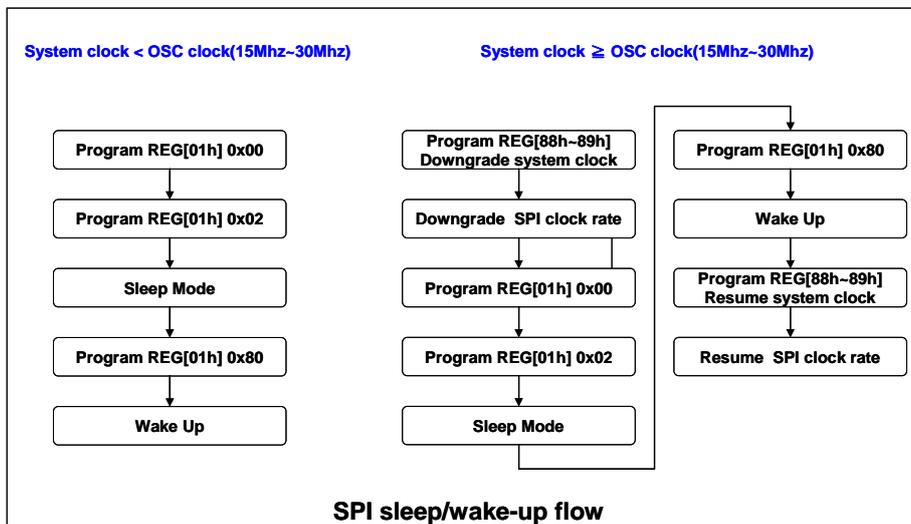


圖 6-19 : SPI 睡眠/喚醒流程

RA8875 SPI 有特定的睡眠/喚醒步驟，使用者請參考下圖步驟：

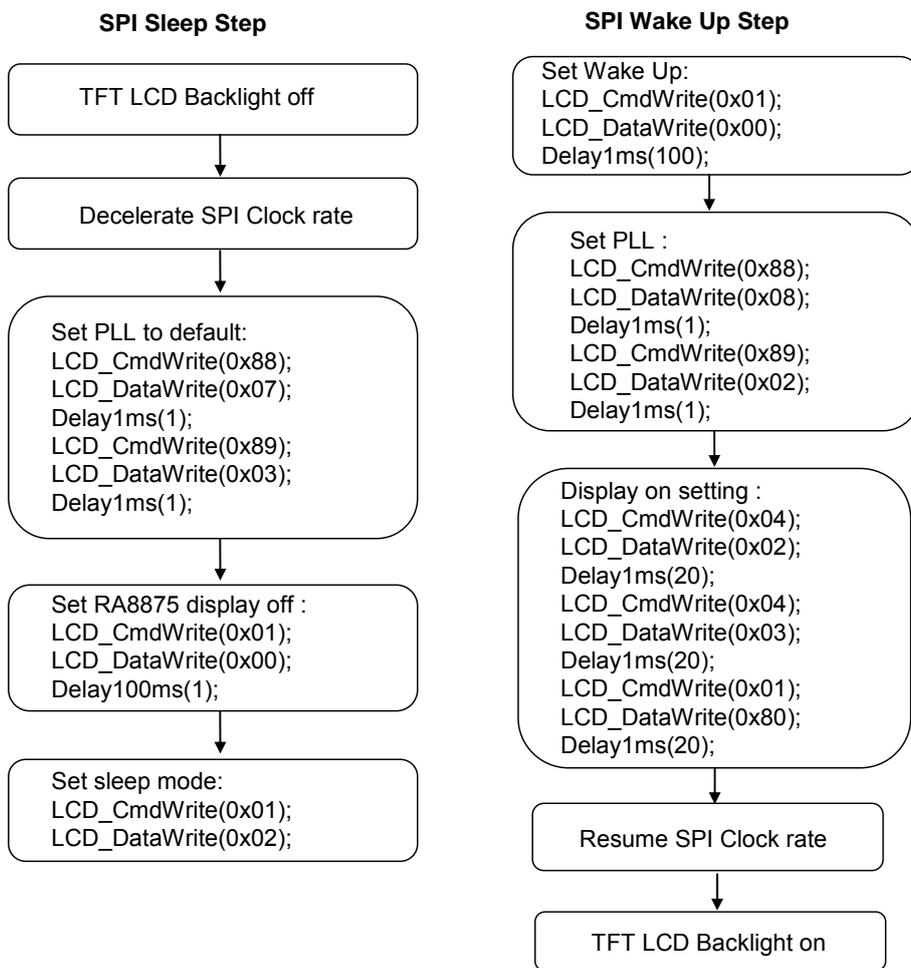


圖 6-20 : SPI 睡眠/喚醒

6-1-2-4 IIC 介面

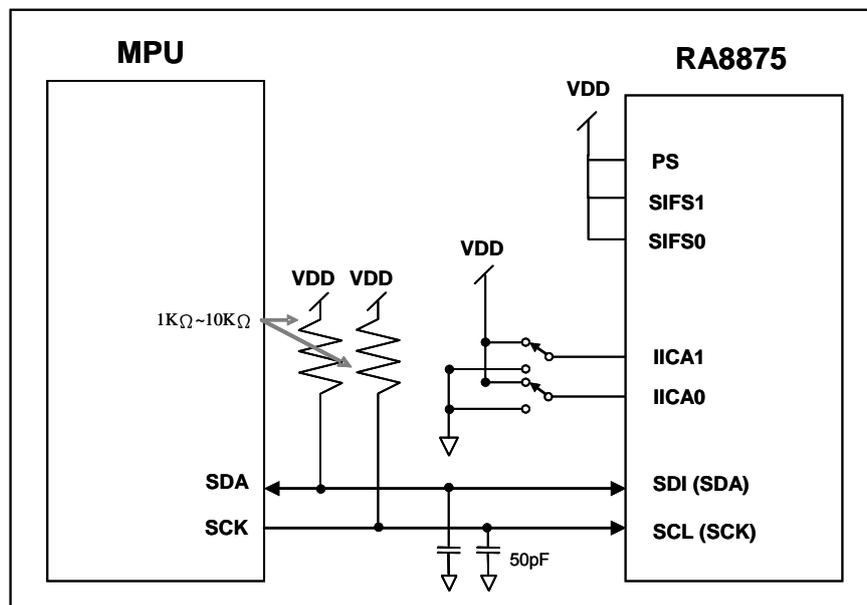


圖 6-21 : IIC 的 MCU 介面圖

IIC 介面由 SCK 與 SDA 兩條資料匯流排線所組成，相容於標準的 IIC 介面，只有直接支援 100K bps 以及 400K bps 兩種模式。IIC 傳輸的前 7 個位元，是指 IIC 的 Spec 中定義的從屬 (Slave) 端位址，在 RA8875 中被分為 2 個部份。前 6 個位元代表 RA8875 的 IIC device ID。接下來 1 個位元是 RS，代表週期類型。當 RS = 1，代表接下來的週期為指令週期；當 RS = 0，為資料週期。若 IIC 匯流排上的週期的 MSB 6 位元與 RA8875 的 device ID 相同，RA8875 的 IIC 從屬 (Slave) 就會動作。

RA8875 的配置位置(Device ID) 是可程式化的，但只限於LSB的 2 個位元，可以直接從IICA[1:0] 的腳位設定。其他MSB的 4 個位元都固定為 0，請參考 表 6-3。RA8875 有 4 種週期類型，分別為：「指令寫入」、「狀態讀取」、「資料寫入」與「資料讀取」週期。週期型態是由RS及RW位元所設定，詳細的說明，請參考 圖 6-22 ~ 圖 6-25。

表 6-3 : IIC 配置位置

IICA [5:0]					
BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0000b				IICA1	IICA0

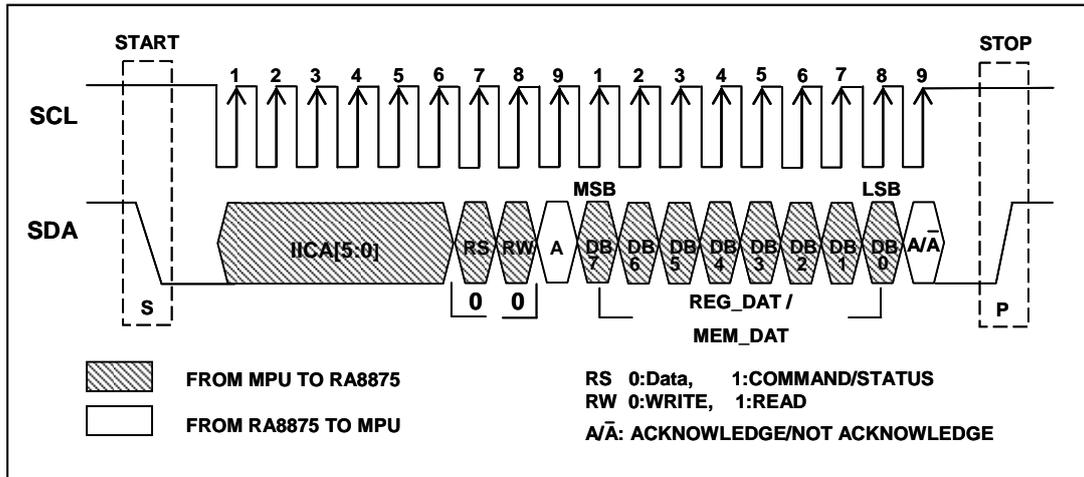


圖 6-22 : IIC 資料匯流排的資料寫入

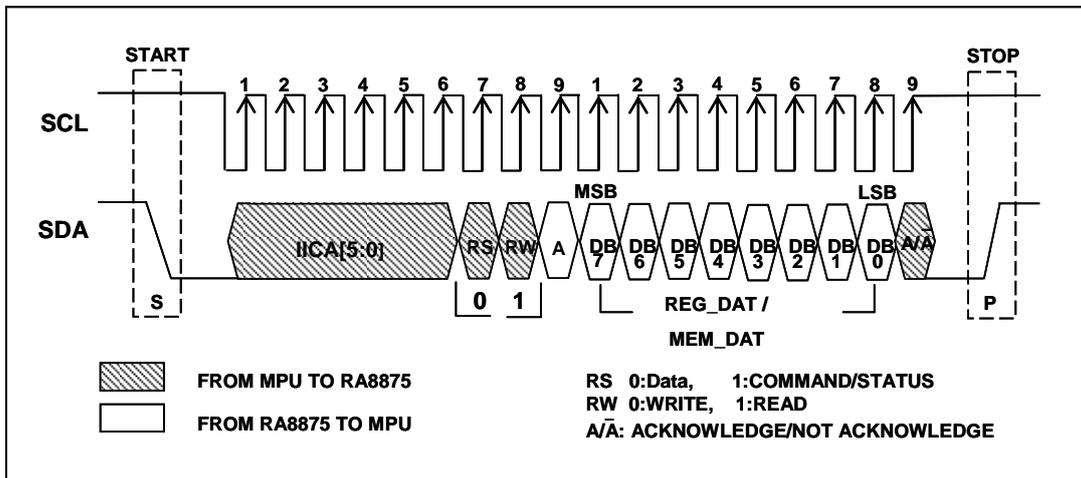


圖 6-23 : IIC 資料匯流排的資料讀取

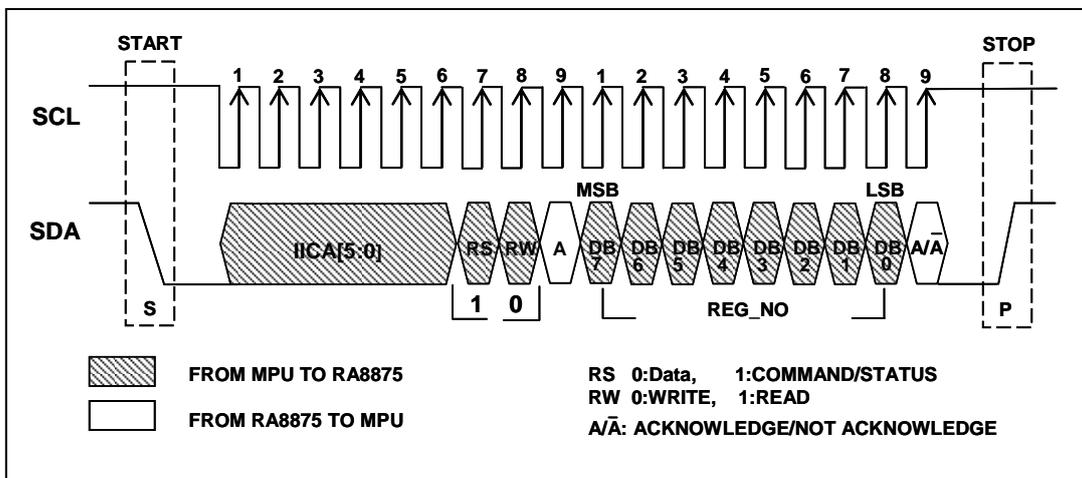


圖 6-24 : IIC 資料匯流排的指令寫入

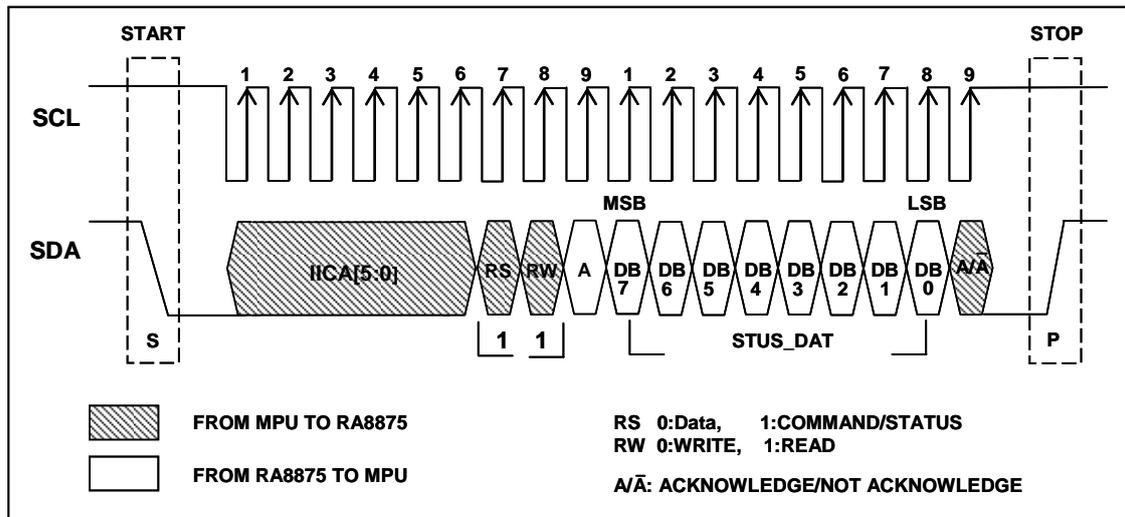


圖 6-25 : 資料匯流排的狀態讀取

6-1-3 狀態讀取暫存器

依據下 表 6-4， RA8875 可以接受四種資料傳輸週期，分別是「指令寫入」、「狀態讀取」、「資料寫入」、「資料讀取」週期。在第五章也提到狀態暫存器是一唯讀 (Read Only) 的暫存器，當“RS”為High時，MCU若對RA8875 進行存取週期，將會得到狀態暫存器的數據。請參考圖 6-26 的時序圖。

表 6-4 : RA8875 的存取週期

RS	WR#	Access Cycle
0	0	資料寫入 (Data Write)
0	1	資料讀取 (Data Read)
1	0	指令寫入 (CMD Write)
1	1	狀態讀取 (Status Read)

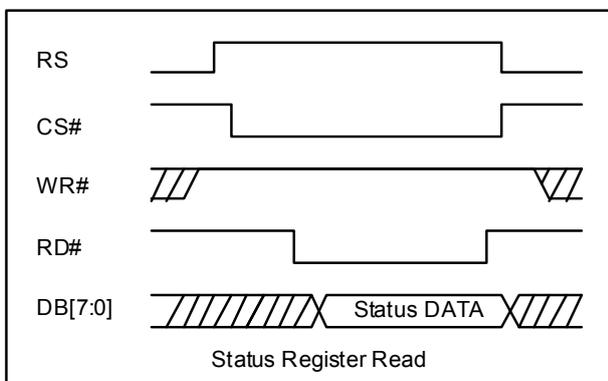


圖 6-26 : 8080 狀態讀取暫存器

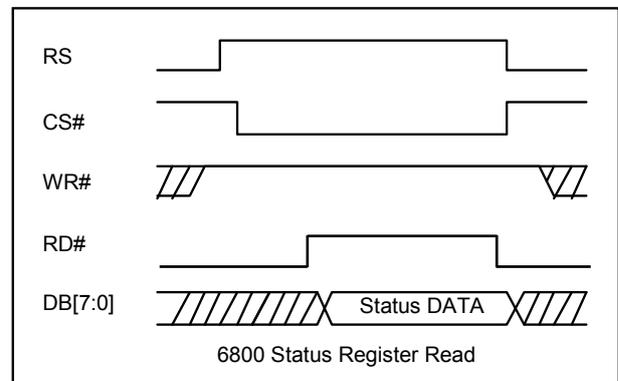


圖 6-27 : 6800 狀態讀取暫存器

6-1-4 指令寫入暫存器

RA8875 有許多個指令暫存器，當要針對某指令暫存器進行寫入指令時，首先必須執行「指令寫入週期」，包括先寫入暫存器位置，然後再以「資料寫入週期」將數值寫入該暫存器。因此，「指令寫入」意指「將資料寫到暫存器當中」，在前述兩個動作執行後，數值資料 (指令) 將被寫入到該暫存器，相關時序請參考圖 6-28 內的 (1)。

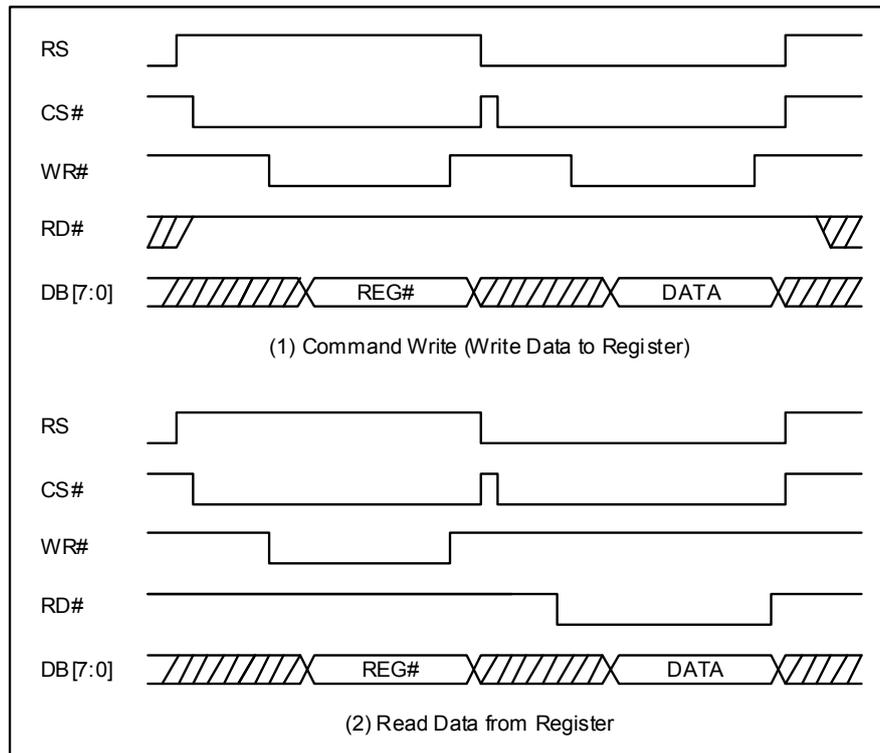


圖 6-28 : 8080 「指令寫入到暫存器」與「暫存器讀取」

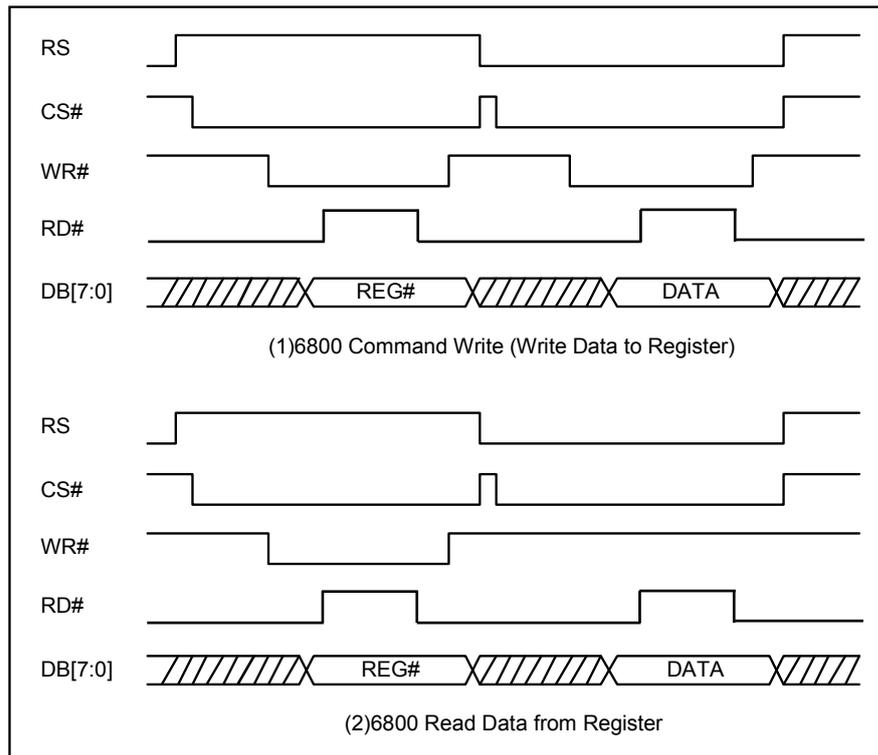


圖 6-29 : 6800 「指令寫入到暫存器」與「暫存器讀取」

若要讀取暫存器中的內容值，則第二個資料傳輸週期為「資料讀取週期」，請參考 圖 6-28 內的 (2)。需注意的是 圖 6-28 是以 8080 的傳輸介面來舉例。

6-1-5 記憶體讀取/寫入的操作

記憶體 (DDRAM或CGRAM) 的讀取/寫入操作是由 2 個週期所組成的。首先執行暫存器 [02h] 的「指令寫入週期」，然後再進行「資料讀取/寫入週期」。暫存器 [02h] 也稱為「記憶體讀寫指令」，用來設定RA8875 進入記憶體讀取/寫入模式。之後的資料寫入/讀取週期便會進行記憶體資料的寫入或資料讀取動作。當更多記憶體資料需要被讀取或寫入時，只要接著先前的週期再執行「資料讀取/寫入週期」，不需要重新再進行「記憶體讀寫指令」。「資料讀取/寫入週期」可持續進行直到完成資料的傳送。需注意在「記憶體讀取/寫入」的模式下不可交錯使用「資料讀取週期」與「資料寫入週期」。因為游標在「記憶體讀取」及「記憶體寫入」是使用不同的操作。詳細的說明請參考章節 7-3。注意「記憶體讀取」應在第一筆資料被讀取時先插入一個「空讀取週期」(Dummy Read Cycle)。「空讀取週期」(Dummy Read Cycle) 與「資料讀取週期」相同，但是裡面的資料是不使用的。「資料讀取週期」在「空讀取週期」後的資料才是正確的，請參考圖 6-30 的說明。

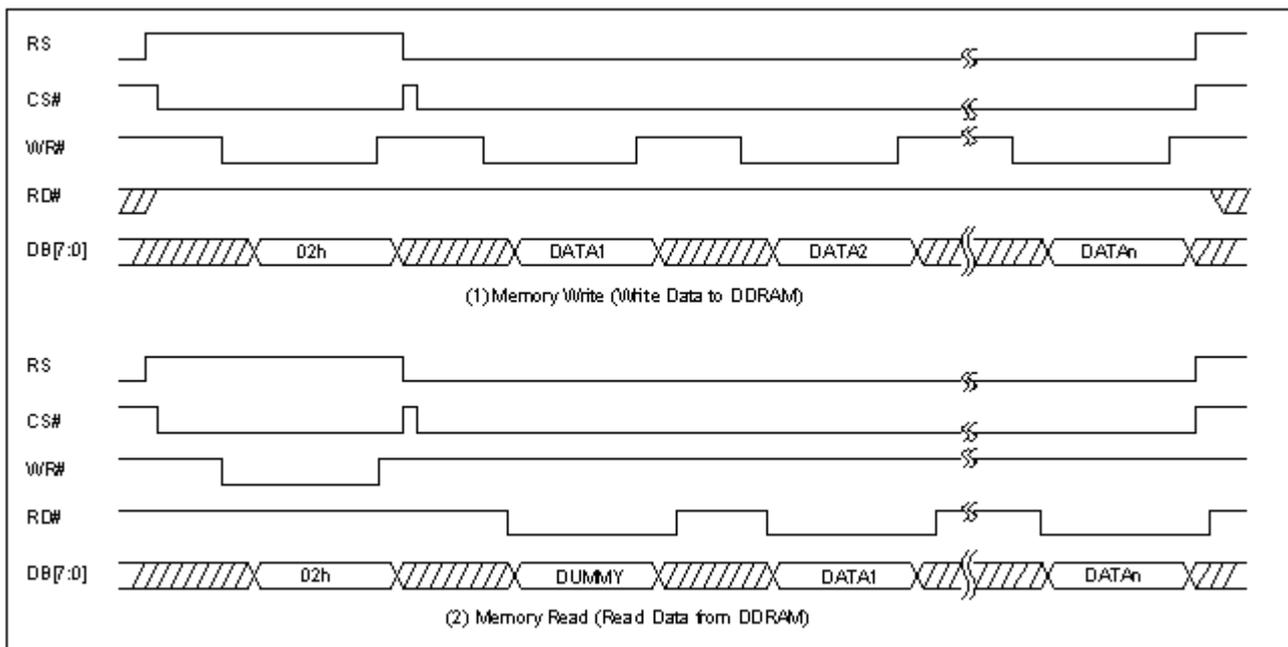


圖 6-30 : 8080 記憶體寫入與記憶體讀取

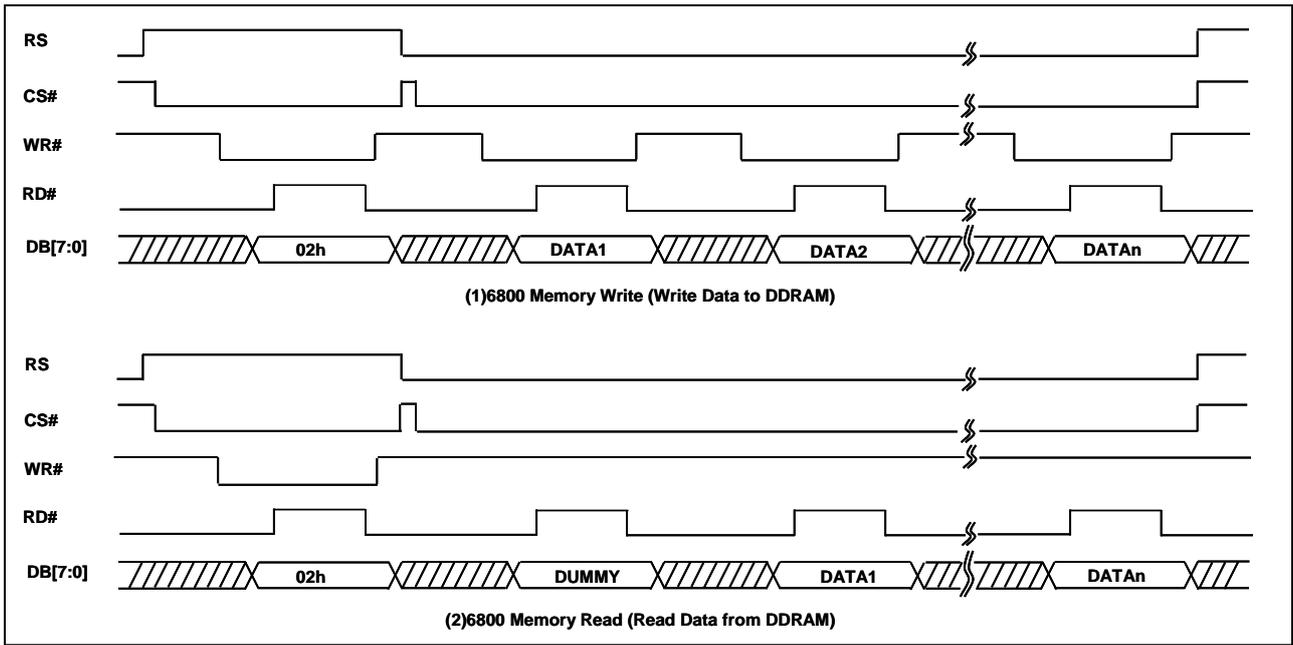


圖 6-31 : 6800 記憶體寫入與記憶體讀取

6-1-6 中斷與等待

RA8875 提供兩種方式的硬體狀態回報方式，分別為中斷 (Interrupt) 與輪詢 (Polling)。對中斷方式而言，RA8875 提供一個中斷信號輸出腳 (INT#) 給MCU去響應RA8875的中斷事件。對輪詢方式而言，也提供了一等待 (WAIT#) 信號輸出腳給MCU，去判斷RA8875 是否處於忙碌狀態。這兩個信號都是低電位觸發，請參考本章的圖 6-1 與圖 6-2。

6-1-6-1 中斷

RA8875 的中斷信號會在以下事件發生時產生，相對應的暫存器為 [F1h]。

- ◆ BTE 完成資料讀寫動作時，REG [F1h] Bit 0 被設定為 1。
- ◆ 文字 (Font) 寫入時，REG [F1h] Bit 0 被設定為 1。
- ◆ BTE 完成圖形移動或填圖時，REG [F1h] Bit 1 被設定為 1。
- ◆ 觸控面板發生被觸摸事件時，REG[F1h] Bit 2 被設定為 1。
- ◆ DMA 事件完成時。
- ◆ 鍵盤掃描 (KEYSCAN) 事件動作時。

這些中斷事件的開啟 (Enable)或關閉 (Disable) 可以透過暫存器 INTC1(REG[F0h]) 的設定來控制。另外，RA8875 還提供了軟體中斷功能，當使用者的系統不支援硬體中斷信號時，可以透過詢問的方式進行軟體中斷。要進行硬體中斷時，使用者必須要把中斷遮罩位元 (Interrupt Mask) 設為 1，其進行步驟如下：

- ◆ RA8875 發出中斷信號給 MCU。
- ◆ MCU 收到中斷信號後，其程式計數器 (PC)會跳到中斷服務程式 (ISR) 的起點。
- ◆ 同一時間 RA8875 的中斷事件相對的旗標位元會被設定為“1” (REG[F1h])。例如，當觸控面板控制器中斷產生，其觸控面板中斷標誌位元就會被設為“1”。
- ◆ 在 ISR 完成時，旗標位元必需被清除。也就是，寫入“1”到相對的狀態暫存器。

若使用軟體中斷方式時，使用者不需要任何外部設置，只要透過讀取暫存器 INTC2 的相關位元就可以檢測中斷是否發生。此外，中斷遮罩 (Interrupt Mask)設置只能應用在硬體中斷，不能遮罩暫存器 INTC2 的相關狀態。要注意的是，因為中斷旗標位元不會自動清除，所以使用者必須在進入中斷程式後手動清除為“0”，就是暫存器 INTC2(REG[F1h])的 Bit2 寫入 1，否則中斷會一直存在而使後續的中斷錯誤。

6-1-6-2 等待

RA8875 也提供一個等待信號(WAIT#)，當 WAIT#為"0"時就意味 RA8875 正處於忙碌狀態，而不能把數據寫入顯示記憶體 (DDRAM)，而其處於忙碌情況可分為以下四種：

1. 當 MCU 用文字模式寫入數據時，字體大小不同的字型需要不同的時間去寫入 DDRAM 裡，在這段時間裡 RA8875 是不能再往 DDRAM 裡寫數據的，此時正處於記憶體寫入忙碌狀態。
2. 當 MCU 發指令讓 RA8875 執行清除螢幕功能時，這段時間裡的 RA8875 在清理 DDRAM 同時也會引起記憶體寫入忙碌。
3. 當 RA8875 在執行 BTE 搬移功能時，此時的 RA8875 會自動進行 DDRAM 的寫入或讀取功能，此時 MCU 執行 DDRAM 的存取會造成顯示異常。
4. 當 MCU 執行指令寫入，RA8875 約需要一個時脈時間 (System Clock) 來寫入，若 MCU 速度比 RA8875 的時脈快出許多，有可能在一個時脈時間內執行兩個或更多的 RA8875 命令，此時建議要檢查 RA8875 是否處於忙碌狀態，當然大部分情況下是不需要特別確認的。

在記憶體寫入忙碌時，向 DDRAM 寫入資料會造成顯示資料的遺失。所以使用者在以上四種情況下寫入顯示資料時，一定要檢查等待狀態。正常情況下，會把等待信號" WAIT#" 接到 MCU 的輸入腳上，MCU 會在 RA8875 寫入數據前，對其忙碌狀態進行監控，其具體時序圖如下所示。

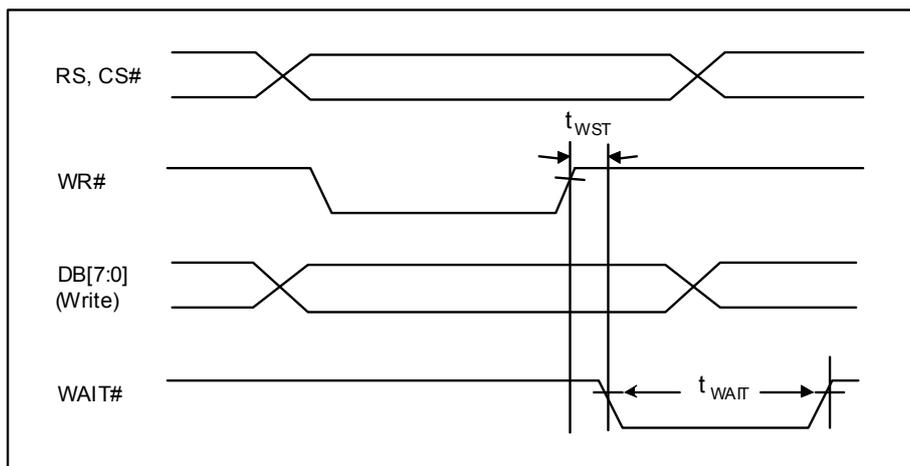


圖 6-32 : WAIT# 時序圖

6-1-7 資料匯流排與TFT的RGB資料格式

6-1-7-1 16位元資料匯流排

RA8875 支援 8-bit/16-bit顏色格式的TFT-LCD 面板，也就是所謂的 256 色與 65K 色TFT-LCD面板。RA8875 支援MCU介面中 8/16 位元的資料匯流排，當寫入顯示記憶體時，RGB 顏色位元排列(RGB color bit mapping) 所對應的資料格式請參考 圖 6-33。

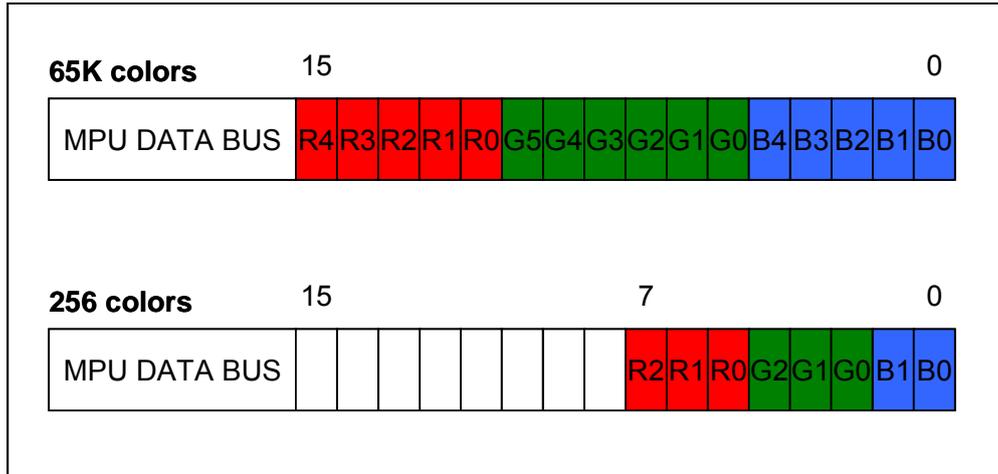


圖 6-33 : MCU 16-bit 時 Data Bus 所對應到的 RGB 數據

6-1-7-2 8位元資料匯流排

當 MCU 使用 8-bit 時，Data Bus 所對應到的 65K 色、256 色 TFT 面板的 RGB 數據如下。

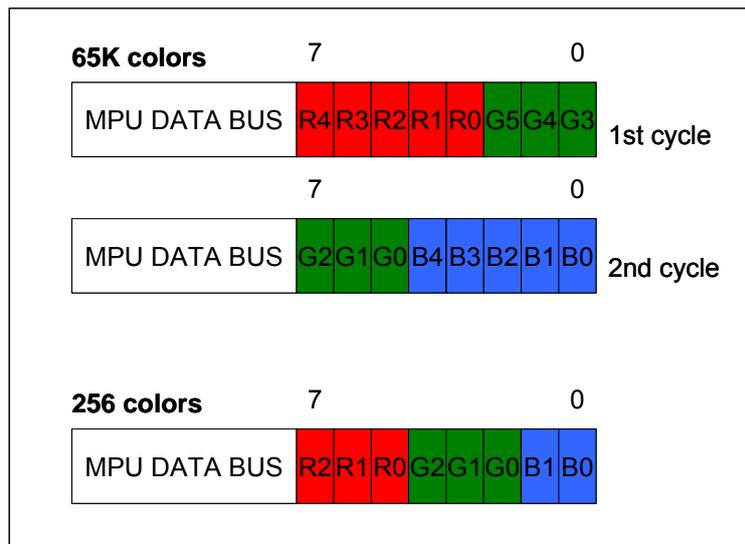


圖 6-34 : MCU 8-bit 時 Data Bus 所對應到的 RGB 數據

6-2 驅動介面的色彩設定模式

RA8875 的邏數位 TFT 驅動介面共有 16 條資料信號，最多可支援 65K 色的資料格式 (Data Format)。透過暫存器的設定，RA8875 可提供 256 色的 RGB 資料格式輸出，使 TFT 顯示達到相同的色彩深度效果。有關色彩設定的暫存器設定，請參考 REG[10h](SYSR) 的 Bit 3-2，其資料格式定義如下。

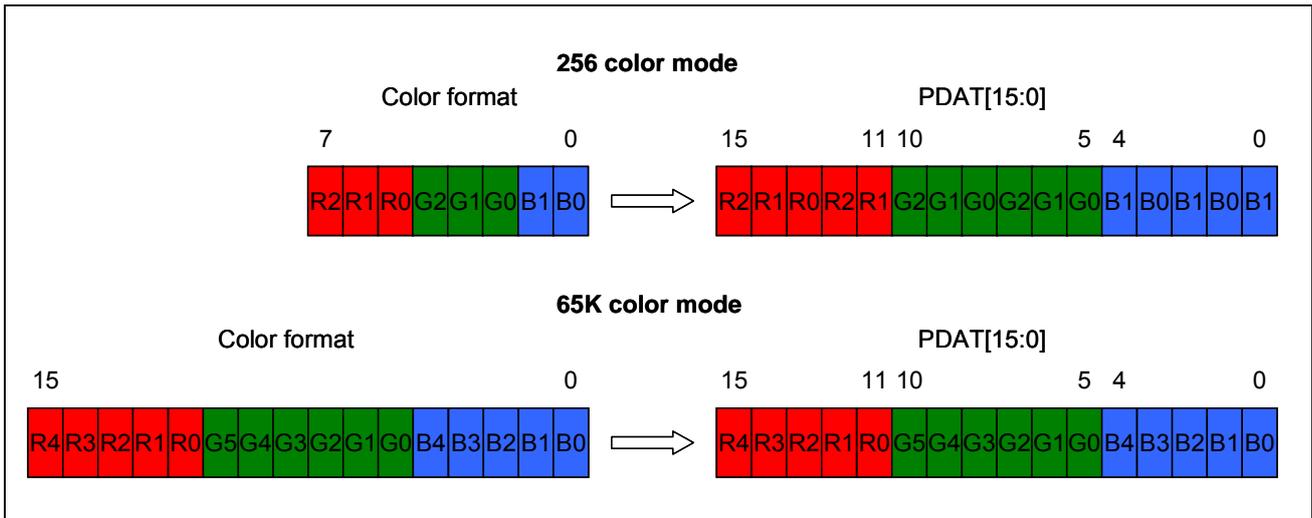


圖 6-35 : 色彩設定

6-3 LCD 介面

RA8875 支援 16-bit 顏色格式的面板尺寸，從 320x240 到 480x272 的雙圖層，640x480 到 800x480 的單圖層。8-bit 顏色格式從 320x240 到 800x480 雙圖層。

RA8875 支援數位TFT介面，透過安排資料匯流排 (Data Bus) 的連接，在大部分的模組皆可正常的使用。表 6-5 是有關RA8875 與數位TFT介面的信號連結說明。相關的時序說明請參考圖 6-36，應用電路請參考圖 6-37。另外，RA8875 的PWM輸出，可用來控制TFT面板的LCD背光，請參考章節 6-7 的說明。

表 6-5：數位 TFT 介面的信號說明

Pin Name	Type	Pin#	Digital TFT Panel			
			8-bit	16-bit	18-bit	24-bit
HSYNC	Output	47	HSYNC Pulse			
VSYNC	Output	48	VSYNC Pulse			
PCLK	Output	49	Pixel Clock			
DE	Output	50	Data Enable			
PDAT[15]	Output	69	R2	R4	R5, R0	R7, R2
PDAT[14]	Output	68	R1	R3	R4	R6, R1
PDAT[13]	Output	67	R0	R2	R3	R5, R0
PDAT[12]	Output	66		R1	R2	R4
PDAT[11]	Output	65		R0	R1	R3
PDAT[10]	Output	64	G2	G5	G5	G7, G1
PDAT[9]	Output	63	G1	G4	G4	G6, G0
PDAT[8]	Output	59	G0	G3	G3	G5
PDAT[7]	Output	58		G2	G2	G4
PDAT[6]	Output	57		G1	G1	G3
PDAT[5]	Output	56		G0	G0	G2
PDAT[4]	Output	55	B1	B4	B5, B0	B7, B2
PDAT[3]	Output	54	B0	B3	B4	B6, B1
PDAT[2]	Output	53		B2	B3	B5, B0
PDAT[1]	Output	52		B1	B2	B4
PDAT[0]	Output	51		B0	B1	B3

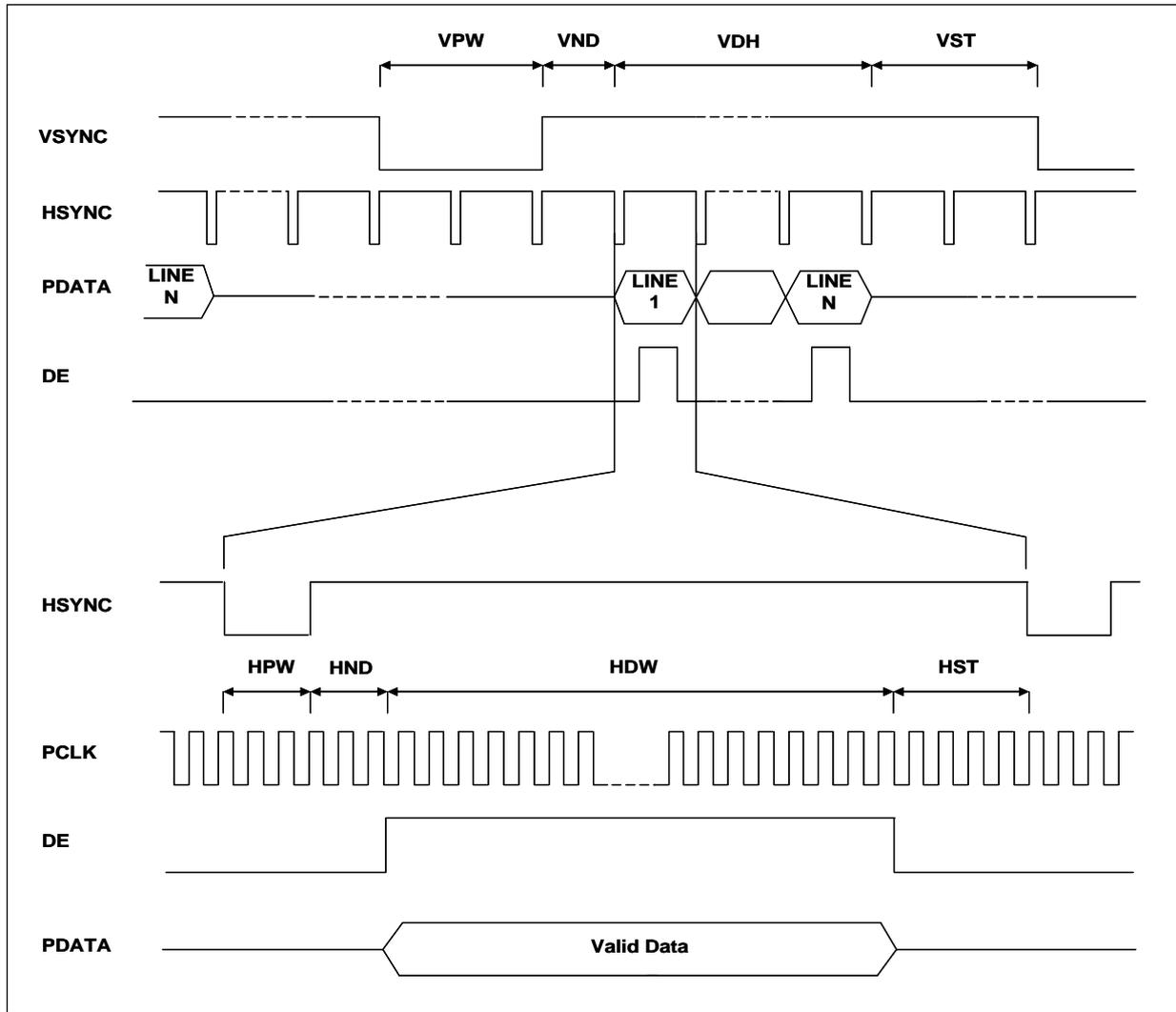


圖 6-36 : 數位 TFT 面板時序圖

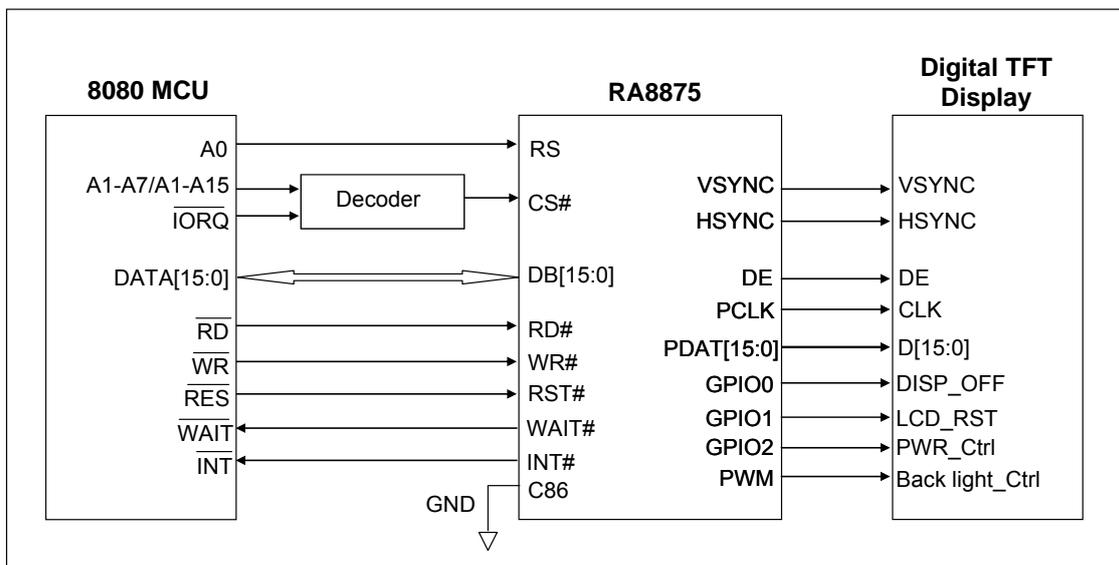


圖 6-37 : RA8875 與數位 TFT 介面

6-4 外部串列式Flash/ROM

RA8875 建立了串列式 Flash/ROM 的介面，來支援下列的傳輸模式：4-BUS 正常讀取 (Normal Read)、5-BUS 快速讀取 (FAST Read)、雙倍模式 0 (Dual mode 0)、雙倍模式 1 (Dual mode 1) 以及模式 0 (Mode 0) 與模式 3 (Mode 3)。

串列式Flash/ROM記憶體功能可用在文字模式 (FONT Mode)、DMA模式及直接存取模式。文字模式意指外部串列式Flash/ROM記憶體被當成字體點陣圖的來源。為了支援文字字體，RA8875 可與專業的字體供應商 — 上海集通公司的FONT ROM相容，相關的細節請參考章節 6-4-1。

DAM 模式意指串列式 Flash/ROM 可當作 DMA (Direct Memory Access) 的資料來源。使用者可以透過此模式，加快資料傳送到顯示記憶體(Display RAM) 的速度。串列式 Flash/ROM 可以直接被串列式介面存取。對不同的串列 Flash/ROM 的類型而言，RA8875 可以設定串列式 Flash/ROM 的時脈到暫存器 [06h]，也就是為 RA8875- SFCL 腳位。需注意當開啟暫存器[E0h] 的直接存取模式後，RA8875 將會忽略暫存器 [05h] 關於 FONT / DMA 的設定。

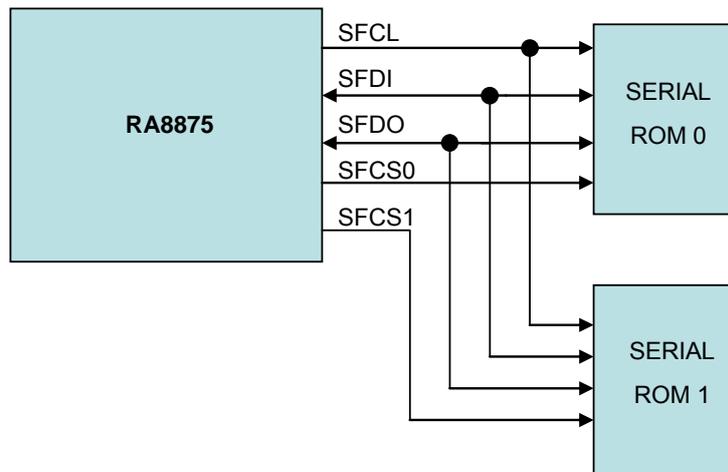


圖 6-38 : RA8875 串列式 Flash/ROM 系統

有關串列式Flash/ROM的傳輸設定，請參考 表 6-6。

表 6-6 : 串列式 Flash/ROM 傳輸相關暫存器參數設定

Protocol	REG [05h] BIT[3]	REG [05h] BIT [1:0]
4-BUS (Normal Read)	0h	0h
5- BUS (FAST Read)	1h	0h
Dual Mode 0	0h	2h
Dual Mode 1	0h	3h

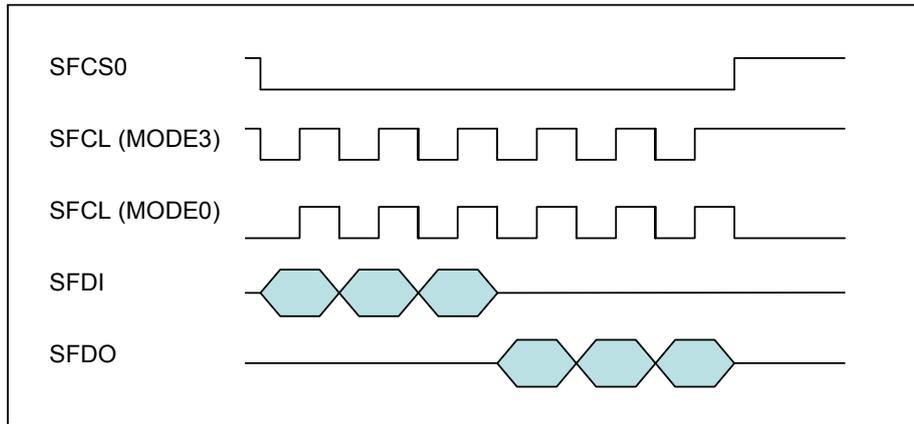


圖 6-39 : Mode 0 與 Mode 3 的傳輸方式

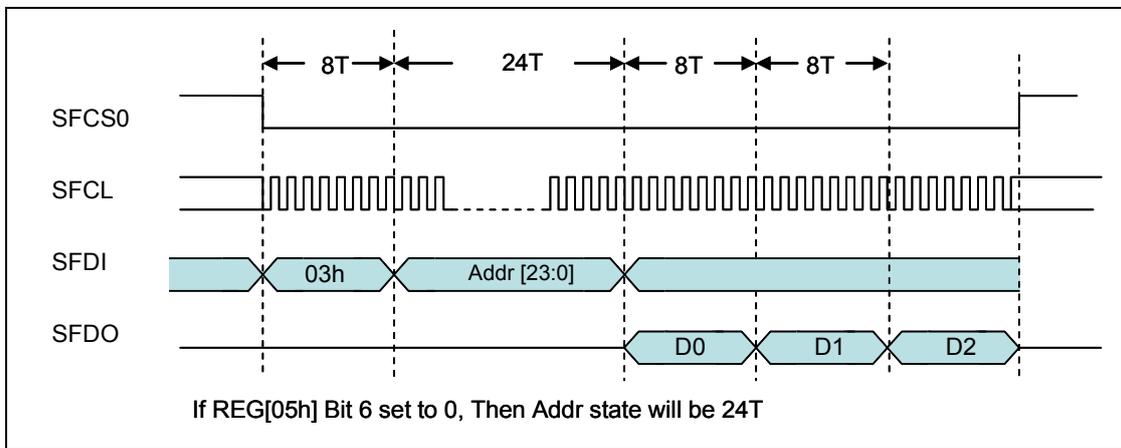


圖 6-40 : 4-BUS (正常) 讀取

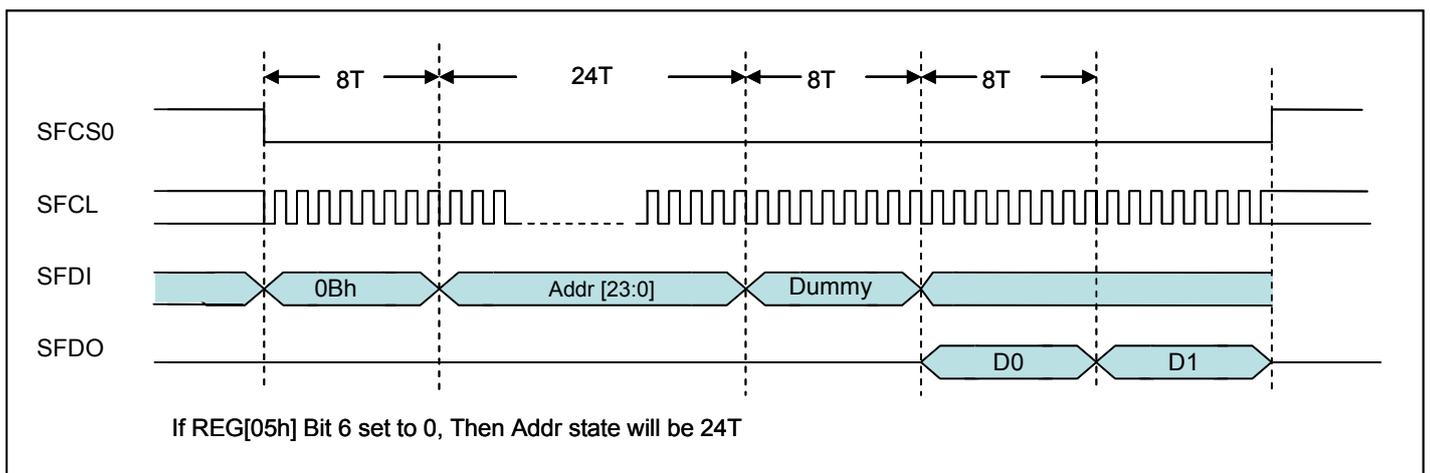


圖 6-41 : 5-BUS (快速) 讀取

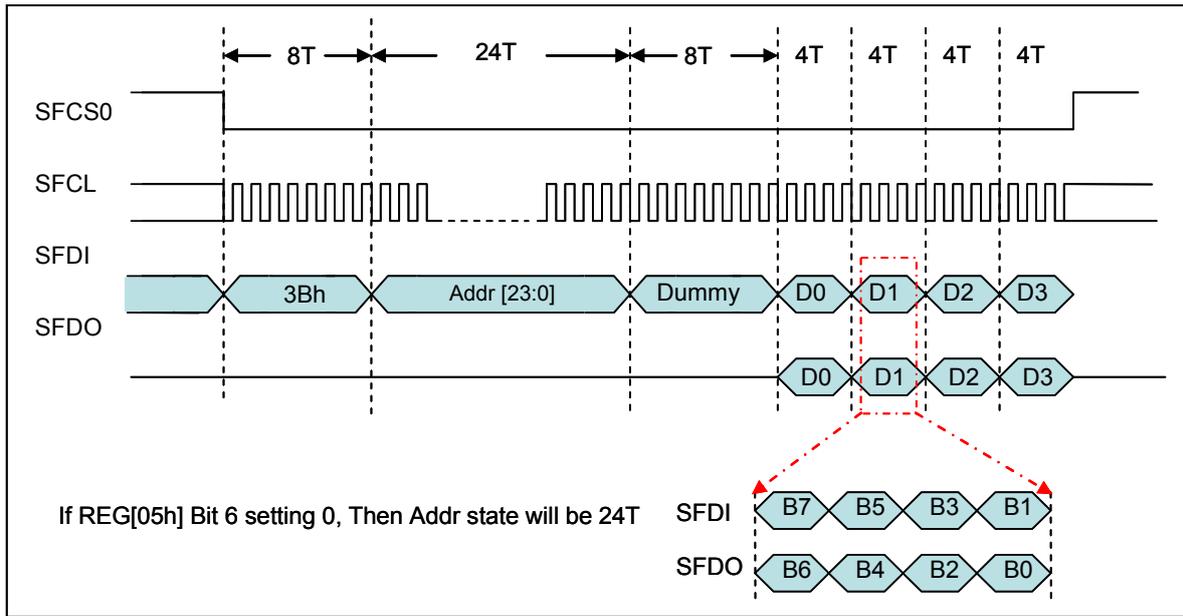


圖 6-42 : 雙倍模式 - 0 讀取

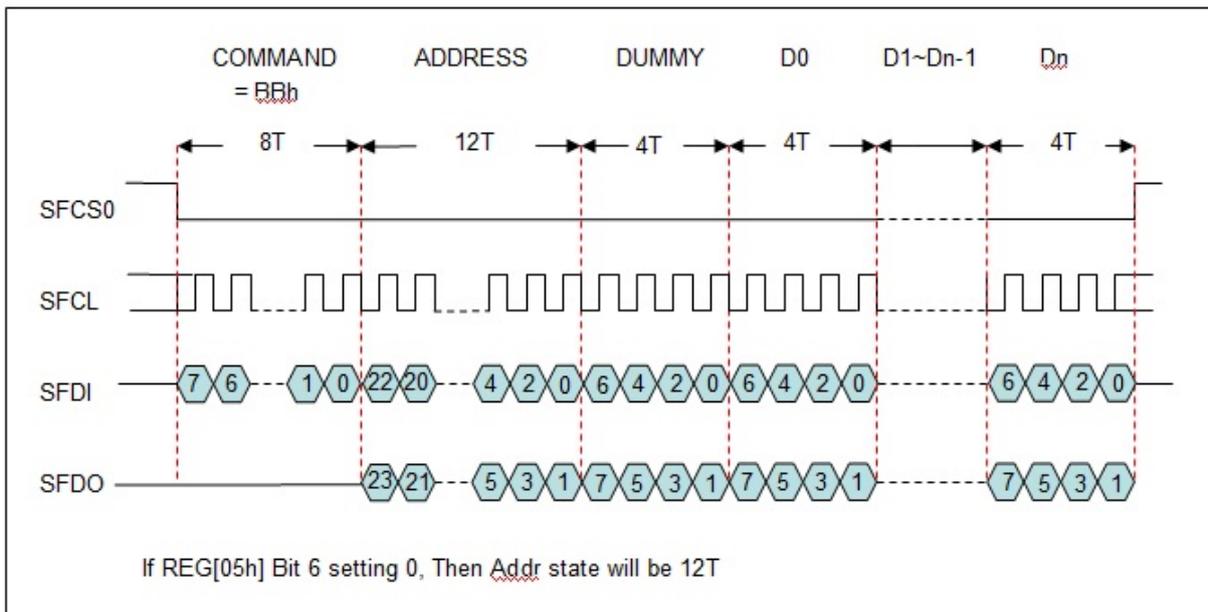


圖 6-43 : 雙倍模式 - 1 讀取

6-4-1 外部串列式字體記憶體

RA8875 透過使用 — 上海集通公司 (Genitop Inc) 外部串列式字體記憶體 (Font ROM)，可支援各樣的文字寫入到 DDRAM 中。RA8875 與上海集通公司 (Genitop Inc) 相容的產品包含：GT21L16TW/GT21H16T1W、GT30L16U2W、GT30L24T3Y/GT30H24T3Y、GT30L24M1Z、及 GT30L32S4W/GT30H32S4W。這些字體包含 16x16, 24x24, 32x32 點 (Dot) 與不同的字寬。

有三種文字編碼的格式：1 byte/2 bytes/4 bytes data，其說明如下：

1. 1 byte 文字編碼 – 所有的字型記憶體 (Font ROMs) 皆為 ASCII code。
2. 4/2 bytes GB 文字編碼 – 在 GT30L24M1Z 內的 GB18030 標準碼。
3. 2 bytes 文字編碼 + 2 bytes 索引碼 (Index Code) – 只用在 GT30L16U2W 的 UNI-CODE。
4. 其它文字編碼長度只有 2 bytes。

在採用特定的字體記憶體 (Font ROM) 前，建議使用者應該先了解編碼的規則，詳細的資訊請與上海集通公司 (Genitop Inc) 聯絡。

需注意在 GT30L16U2W 的規格書中，UNI-CODE 文字編碼需要額外參照 “ZFindex Table” 來決定真正的記憶體輸入位元位置 (Bitmap ROM Address)。若使用者在 00A1h~33D5h 或 E76Ch~FFE5h 的範圍內寫入一個 UNI-CODE，是一個特定的編碼區域，之後額外的 2bytes 文字編碼就需要參考 “ZFindex Table”。其他外部的 UNI-CODE 範圍只需要 2bytes 的文字編碼，詳細說明請參考 GT30L16U2W 的規格書。

舉例說明，若使用者用 GT30L16U2W 寫入 UNI-CODE (00A2)，位在 00A1h~33D5h 的範圍，之候 MCU 必須寫入額外的 2bytes 文字編碼 — 來自 ZFindex 到 RA8875 的索引。

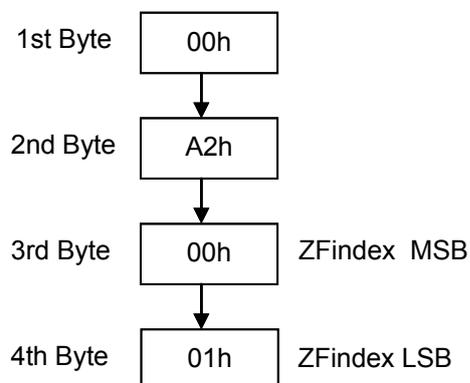


圖 6-44

註：其他資訊請參考章節 7-4-2。

6-4-2 外部串列式資料記憶體

外部串列 Flash/ROM 介面可以當作資料的來源，在 RA8875 中可用兩種模式所存取。

◆DMA (Direct Memory Access) 模式

串列Flash/ROM介面可以用來當作DMA功能的資料來源，Flash/ROM可以用來做大量資料的儲存，詳細說明情參考章節 7-10。

◆直接存取模式

串列Flash/ROM介面可以用RA8875 直接進行存取。首先透過內部暫存器的位置設定，之後設定位置的資料可以從特定的暫存器來進行讀取。請參考 圖 6-45 的流程圖。

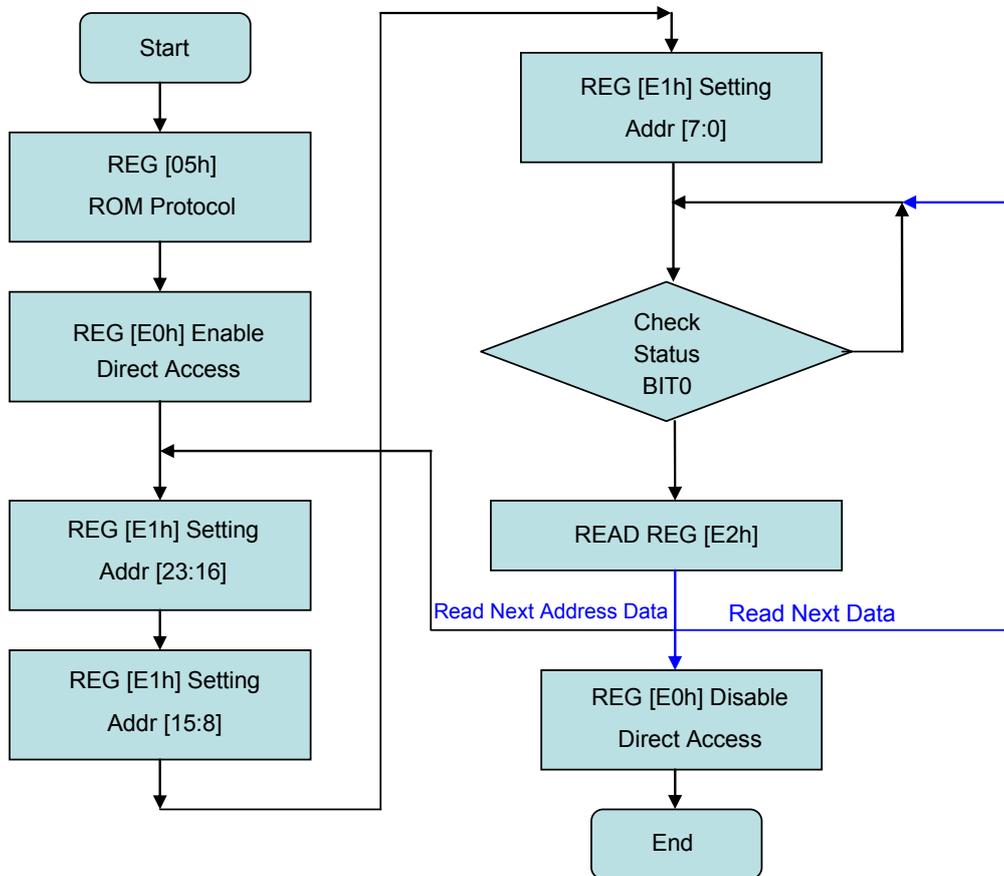


圖 6-45：直接存取模式流程圖

6-5 觸控介面

RA8875 內建一組 10 位元ADC和控制電路，可以連接 4 線電阻式的觸控面板。4 線電阻式的觸控面板是由兩層非常薄的電阻式面板所組成，如圖 6-46，在兩層面板中間有一小縫隙，當有外力施加在面板上某一點時，兩層電阻式面板將被觸碰 (Touch)，形成迴路而導通，由於兩層電阻式面板的端點含有電極 (XP,XN,YP,YN)，如圖 6-47，因此相對於觸碰的位置，系統將偵測到一個XY的座標值。

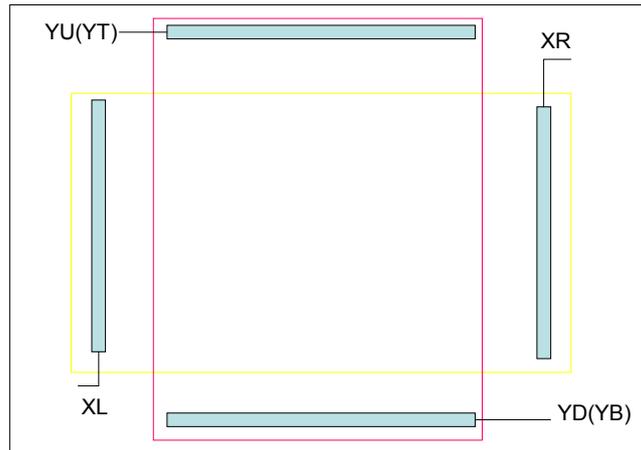


圖 6-46 : 4-wire 觸控面板架構

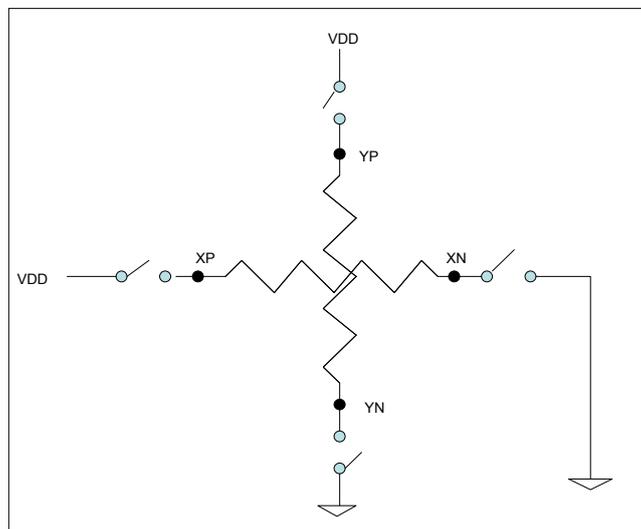


圖 6-47 : 4-wire 觸控面板原理

對使用者而言，應用RA8875 4 線電阻式觸控面板功能只需連接XR,XL,YD,YU等四條信號線到RA8875 即可，系統就能不斷監測，直到觸控的事件 (Touch Event) 發生為止。當觸控事件發生時，在觸控面板電阻上所產生的分壓將決定觸控所在位置。在XY座標值被傳回到RA8875，並個別儲存在特定的暫存器後，觸控面板控制器 (Touch Panel Controller) 將發出一中斷告知MCU。圖 6-48 為 4-wire觸控面板的應用電路。

腳位 ADC_VREF 是 ADC 的輸入參考電壓，可以由暫存器[71h] 的 Bit5 來決定使用內部產生參考電壓，或是外部輸入參考電壓，當使用外部參考電壓時，可用分壓電阻產生一 $1/2 VDD$ 的電壓給 ADC_VREF，正負誤差在 5%之內，同時必須接一個 $1\sim 10\mu F$ 的電容到地以提生 ADC 轉換的穩定度。

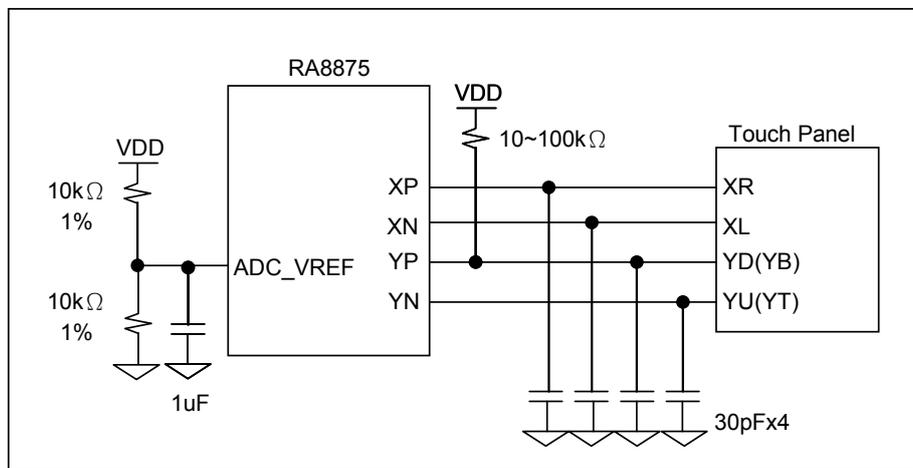


圖 6-48 : 4-wire 觸控面板應用電路

6-6 鍵盤掃描功能

RA8875 內建一鍵盤掃描電路，使系統具備鍵盤 (Keyboard) 功能，它有助於整合具有鍵盤功能的應用電路。如圖 6-49 為一 4x5 鍵盤的基本應用電路。

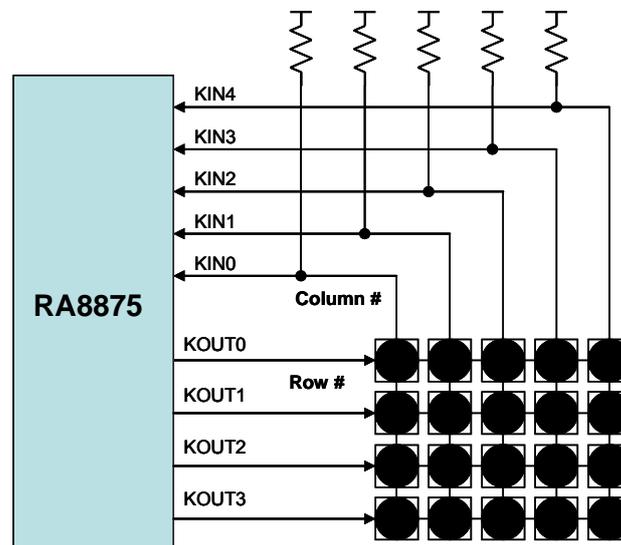


圖 6-49 : 4x5 鍵盤的基本應用電路

6-7 脈寬調變介面

RA8875 提供兩個可調節的脈寬調變 (PWM) 輸出，可供給 LCD 進行背光亮度調節 (Backlight) 或其他用途。其 PWM 的頻率和工作週期(Duty Cycle) 都可以透過相關暫存器的設定來調整。

圖 6-50 是應用 PWM 調節 TFT 面板背光的應用電路，此電路可用 PWM 的輸出從 0% ~ 100% 工作週期 (Duty Cycle) 來控制 LED 電流約 20mA ~ 0mA。

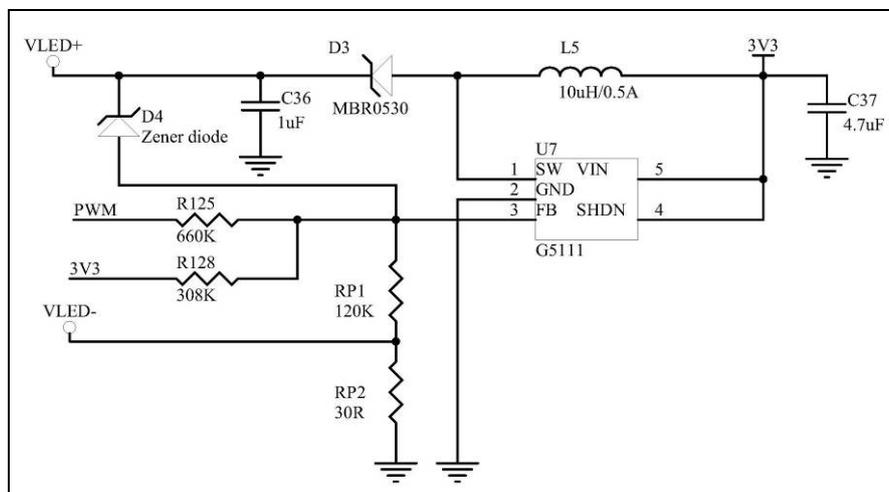


圖 6-50 : PWM 控制 TFT 面板背光的應用電路

6-8 時脈 (Clock) 與 PLL

RA8875 系統時脈 (System Clock) 是由外部的石英震盪器，配合內部晶體振盪電路及 PLL 電路所產生。內部的晶體振盪電路，它結合外部在 XI 和 XO 兩腳間的石英震盪器 (15MHz~30MHz)和電阻、電容產生基本頻率，再透過 PLL 電路及暫存器 (REG[88h]、[89h]) 設定，然後產生系統時脈供 RA8875 內部使用。相關的示意圖如下。

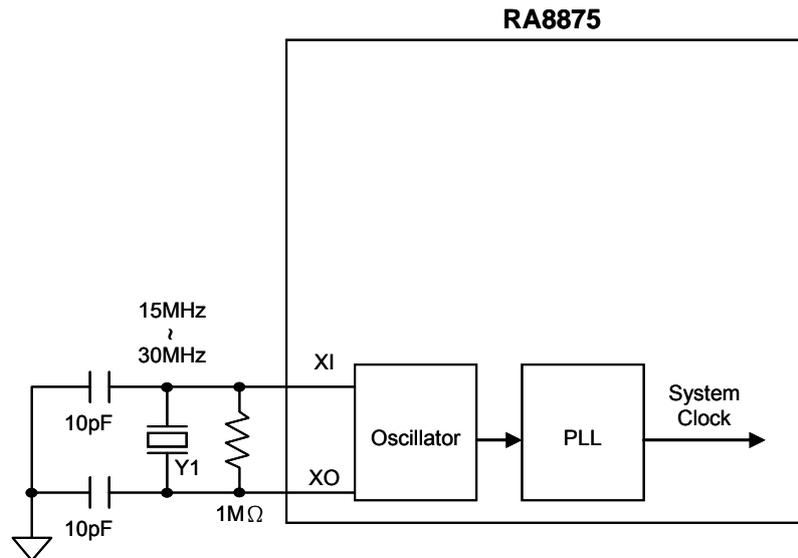


圖 6-51：石英震盪電路與系統時脈的產生

RA8875 系統時脈產生的公式如下：

$$\text{System Clock} = Y1 \times (\text{PLLDIVN}[4:0] + 1) / ((\text{PLLDIVM} + 1) \times (2^{\text{PLLDIVK}[2:0]}))$$

例如：

$$Y1 = 20\text{MHz}$$

$$\text{PLLDIVM} = 0, (\text{PLLDIVM 也就是 REG}[88\text{h}] \text{ 的 Bit}7)$$

$$\text{PLLDIVN}[4:0] = 01011\text{b}, (\text{PLLDIVN 也就是 REG}[88\text{h}] \text{ 的 Bit}[4:0])$$

$$\text{PLLDIVK}[2:0] = 010\text{b}, (\text{PLLDIVK 也就是 REG}[89\text{h}] \text{ 的 Bit}[2:0])$$

$$\begin{aligned} \text{System Clock} &= 20\text{MHz} \times (11+1) / ((0+1) \times (2^2)) \\ &= 20\text{MHz} \times 10 / 4 \\ &= 60\text{MHz} \end{aligned}$$

系統頻率 (SYS_CLK) 的預設值與外部晶體振盪器頻率相同。同時注意，當 REG[88h] 或 REG[89h] 被設定後，為了保證 PLL 輸出的穩定，必須等待一段「鎖頻時間」(約 >100us)以完成 PLL 頻率改變的程序。

RA8875 支援多種解析度的面板，然而較高解析度的面板需要更快的系統時脈，除了考慮配合MCU達到更快的畫面資料更新，也要讓面板的Frame Rate不致太低造成畫面閃爍，表 6-7 代表不同面板解析度與Clock 的需求，這些設定的前提以顯示換頁頻率為 60 Frame/sec和假設非顯示區和顯示區時間比例為 20%。

表 6-7 : Clock 設定與面板解析度的關係

Display Resolution	Layer No.	Color Depth (Bits)	Frame (Hz)	Pixel Clock (PCLK)
320x240	2	16	60	6.4MHz
320x480	2	16	60	12.8MHz
480x272	2	16	60	9MHz
640x480	2	8	60	25MHz
640x480	1	16	60	25MHz
800x480	2	8	60	30~33MHz
800x480	1	16	60	30~33MHz

6-9 重置功能

在RA8875程式化之前，建議先完成重置動作。RA8875的重置 (Reset) 動作必須在供電後提供不少於 $1024 \cdot t_c$ 的時間來進行，以25MHz 的系統頻率來說，其重置脈衝寬度就要不少於40.96 μ s。為了讓RA8875 正確的接受指令，我們建議RA8875供電後一定要進行重置動作。

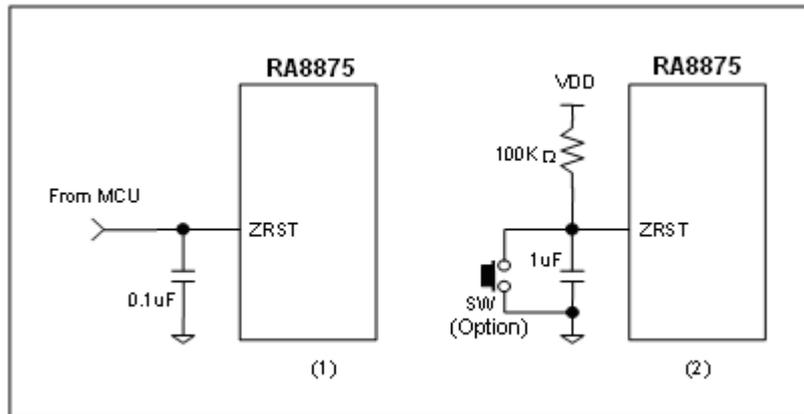


圖 6-52 : RST# 接腳的使用範例

圖 6-52 是一重置電路接法範例。其重置動作可以用MCU去控制如圖 6-52 內的 (1)，也可以由一RC電路來產生如圖 6-52內的 (2)。

RA8875 在重置過程中不能接受MCU的任何指令，所以應在重置後才可對內部暫存器進行初始化等設定，在VDD穩定後，重置腳RST# 在上升沿之後最少需延遲 1ms的時間才能進行其他操作，這樣可確保系統的穩定性，詳細參數要求可以參考圖 6-53。

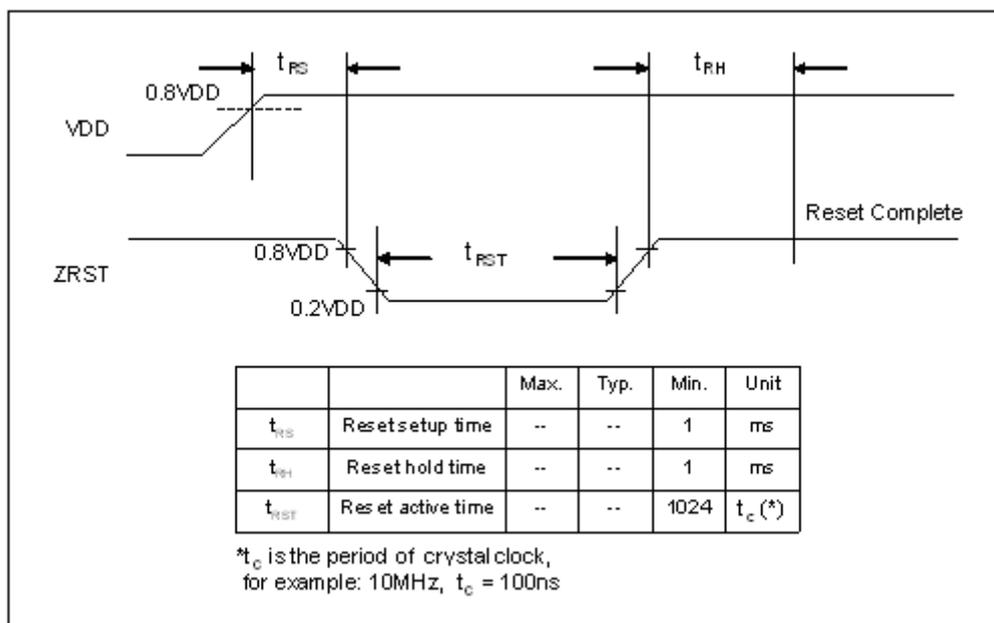


圖 6-53 : 重置時序參數

RA8875 在重置時 (RST# = Low)，相關輸出信號的狀態如 表 6-8 所示。

表 6-8 : 重置時相關輸出信號的狀態

Signal Name	Output Status
WAIT#, INT#	High
PWM1, PWM2	Low
PDAT[15:0]	Low
VSYNC, HSYNC	High
PCLK, DE	Low
KOUT[3:0]	Low
GPOX	Low

6-10 電源

6-10-1 電源腳位說明

RA8875 所需的工作電源是 3.3V 及 1.8V。3.3V 用於供應 IO、LDO、DAC 與 ADC 所需電源。而 1.8V 可以由內部 LDO 產生，用於供應核心電路所需電源。為了考慮 RA8875 的正常工作，請勿將 LDO 產生的 1.8V 電源供給其他元件使用，詳細的電源腳位說明請參考章節 4-8。

6-10-2 電源結構

RA8875 的電源結構如下圖 6-54 所示，在 PCB 板設計上，每個電源端加上濾波電容 (0.1uF)，同時儘可能接近 RA8875 的電源接腳。同時 LDO 輸出端 LDO_CAP 與 LDO_OUT 則務必各自接上 1uF 以上的濾波電容，已達到良好的穩壓效果。

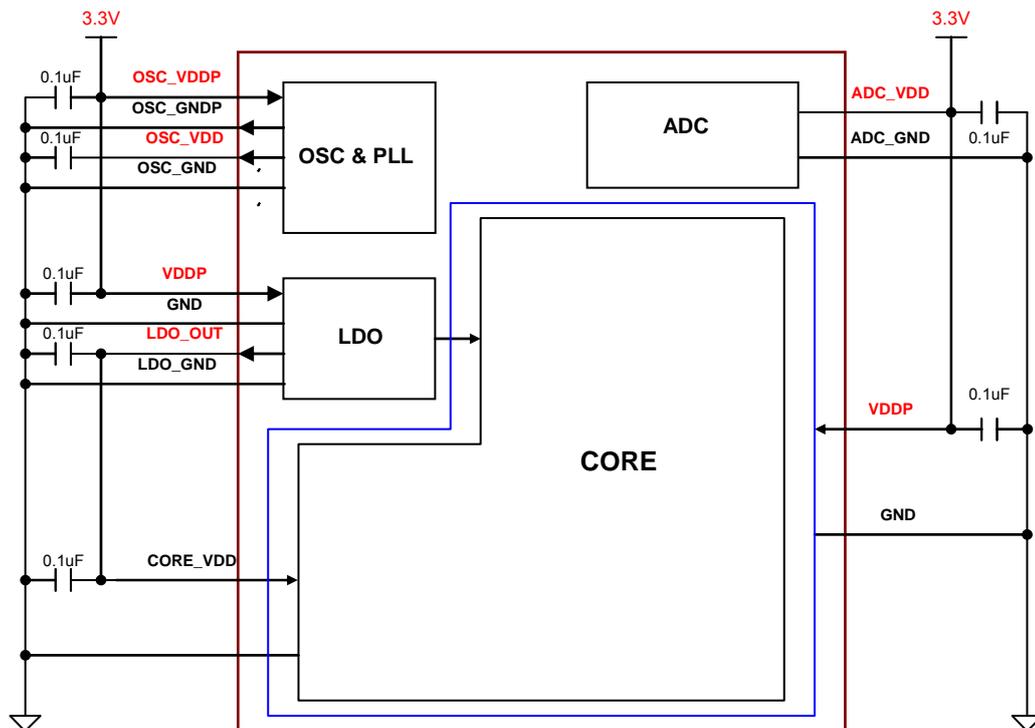


圖 6-54 : RA8875 的電源接法

7. 功能描述

7-1 捲動功能

RA8875 提供水平及垂直捲動兩種功能。藉由設定捲動視窗的偏移值，整個顯示區域可移動一個偏移值，而且偏移超過捲動視窗右邊界的區域，會從捲動視窗開頭的地方再開始顯示，就像是“捲動”的效果一般。

7-1-1 捲動視窗與捲動偏移值

捲動視窗的定義了捲動作用的範圍。捲動偏移值則為捲動視窗的捲動效果。在捲動範圍內，顯示效果會隨偏移值的單位設定（像素）而移動。透過暫存器的設定來增加或減少捲動偏移值，可看到捲動的效果。捲動範圍外的區域則不受捲動偏移值的影響。捲動視窗是由顯示區域的兩個點來設定的，例如：起始點及結束點。起始點及結束點是由對等的方法來表示。有關捲動視窗暫存器及偏移值的設定，請參考 表 7-1 及 表 7-2。請注意：HSSW 必須小於 HESW，且 VSSW 必須小於 VESW。

表 7-1：捲動視窗暫存器設定

Reg. NO.	Abbreviation	Description
38h, 39h	HSSW[9:0]	Horizontal Start Point of Scroll Window
3Ah, 3Bh	VSSW[8:0]	Vertical Start Point of Scroll Window
3Ch, 3Dh	HESW[9:0]	Horizontal End Point of Scroll Window
3Eh, 3Fh	VESW[8:0]	Vertical End Point of Scroll Window

表 7-2：捲動偏移值暫存器設定

Reg. NO.	Abbreviation	Description
24h, 25h	HOFS[10:0]	Horizontal Scroll Offset Register
26h, 27h	VOFS[9:0]	Vertical Scroll Offset Register

7-1-2 水平捲動與垂直捲動

RA8875 提供水平捲動的功能。使用者可以在顯示區域利用增加或減少解析度的值來移動，靈活地分配捲動範圍。使用者完成區塊捲動的效果，請參考 圖 7-1。

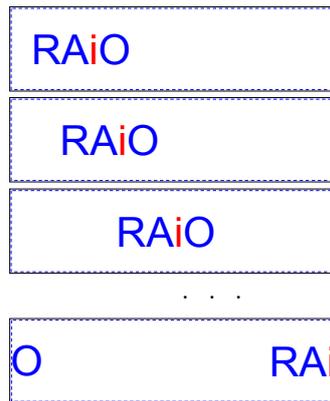


圖 7-1 : 水平捲動效果

註: 水平旋轉偏移值 HOFS 必須小於水平旋轉設定範圍 HESW – HSSW。

垂直捲動的功能與水平捲動的功能相似，不同地方的是偏移值的設定會造成垂直捲動效果。請參考圖 7-2 的範例。需注意水平偏移值與垂直偏移值可以同時設定。

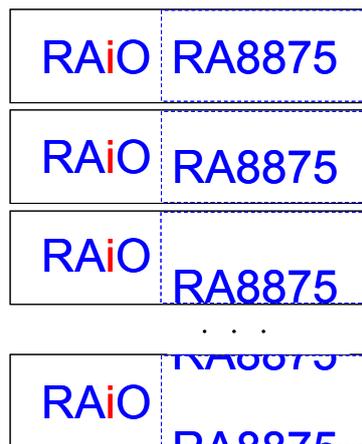


圖 7-2 : 垂直捲動效果

註: 垂直旋轉偏移值 VOFS 必須小於垂直旋轉設定範圍 VESW - VSSW。

7-1-3 圖層混合捲動

圖層混合捲動功能與先前描述的其他捲動功能相似。使用者的應用共有 4 種模式：圖層 1 的捲動、圖層 2 的捲動、雙圖層同時捲動、用圖層 2 作為緩衝的捲動，有關暫存器的設定請參考表 7-3。

表 7-3：捲動功能的暫存器設定

Reg. NO.	Abbreviation	Description
Layer Transparency Register 0		
52h	LTPR0	B[7:6]Layer 1/2 Scroll mode 00b:Layer 1/2 scroll simultaneously. 01b:Only Layer 1 scroll 10b:Only Layer 2 scroll 11b:Buffer scroll (use Layer 2 as buffer)

7-1-3-1 圖層1 與圖層2 同時捲動

當圖層 1 與圖層 2 捲動模式被設為 00b，使用者可以透過增加或減少偏移值 (單位: 像素)，更彈性的分配顯示區域的捲動範圍，圖層 1 與圖層 2 可以同時捲動。可透過的設定LTPR0[2:0] 來覆蓋圖層 1 與圖層 2。需注意若同時設定圖層 1 與圖層 2，且當LTPR0[2:0] 被設為只可顯示圖層 1 或只可顯示圖層 2 的情況下，只會顯示出一個圖層。顯示效果請參考 圖 7-3 的案例。

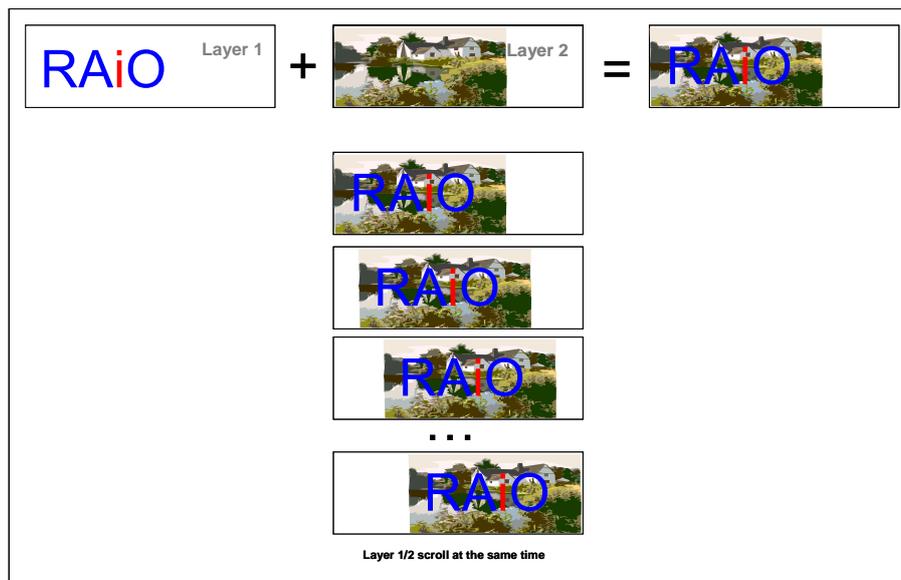


圖 7-3：圖層 1 與圖層 2 同時捲動的效果

7-1-3-2 只捲動圖層1

當LTPR0[7:6] 被設為 01b時，捲動視窗中只能顯示圖層 1。調整顯示效果案例請參考 圖 7-4。

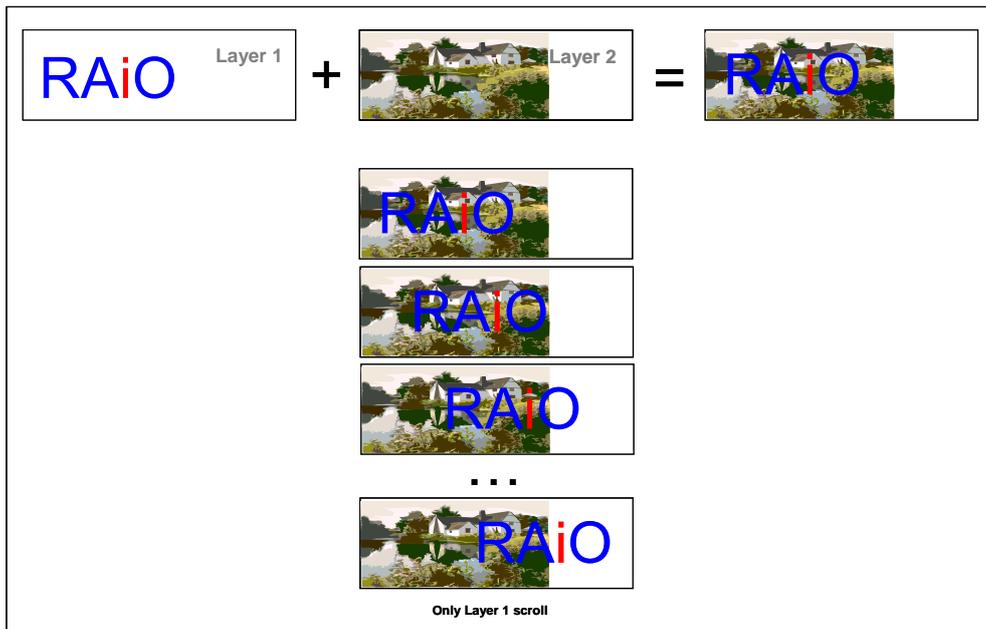


圖 7-4 : 只捲動圖層 1 的效果

7-1-3-3 只捲動圖層2

當LTPR0[7:6] 被設為 10b，捲動視窗中只能顯示圖層 2。顯示與圖層 1 相似，此功能在不同應用上更有彈性。相關的顯示效果請參考 圖 7-5。

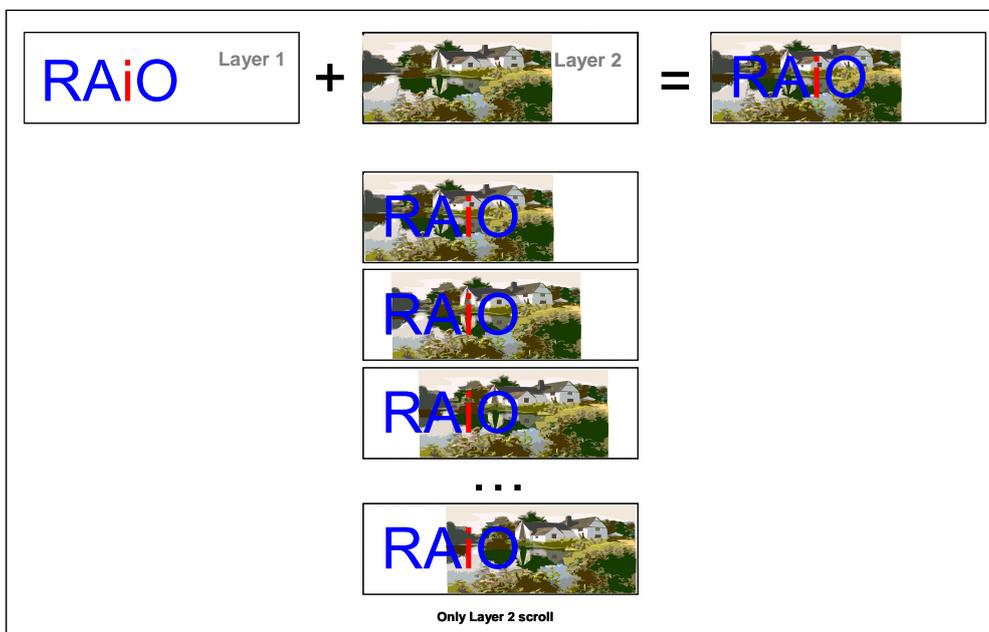


圖 7-5 : 只捲動圖層 2 的效果

註：HOFS(REG[24h-25h]) 的偏移值必須小於 HESW – HSSW，且 VOFS(REG[26h-27h]) 的偏移值必須小於 VESW – VSSW。

7-1-3-4 緩衝捲動

當LTPR0[7:6] 被設為 11b，可設定緩衝捲動模式。在圖層 1 與圖層 2 的捲動視窗記憶體區域，被視為捲動顯示的連續性記憶體。無論水平或垂直捲動的偏移值，都可以設為捲動視窗的 2 倍長度或寬度。在實際的應用上非常有用，因為有一捲動區域的區塊是看不見的。當此區塊為不可見的，使用者可以進行更新，捲動效果可以流暢的進行，請參考圖 7-6 例子。

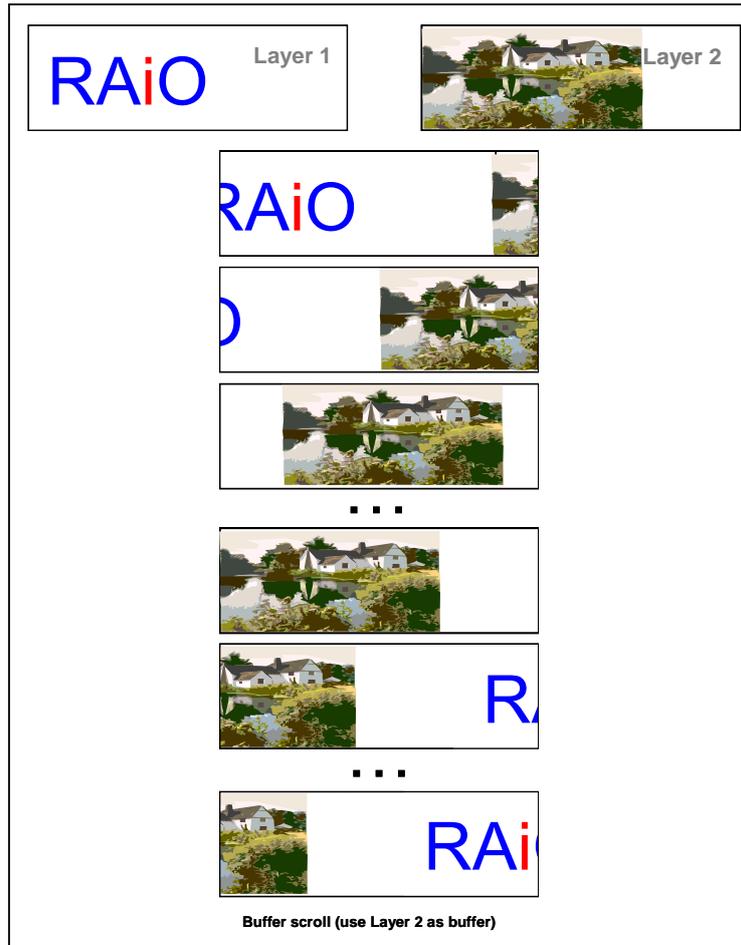


圖 7-6：緩衝捲動的效果

註：HOFS(REG[24h-25h]) 的偏移值，必須界於 $0 \leq \text{HOFS} \leq (2\{\text{HESW} - \text{HSSW}\} + 1)$ 間，而且 VOFS(REG[26h-27h]) 的偏移值，必須界於 $0 \leq \text{VOFS} \leq (2\{\text{VESW} - \text{VSSW}\} + 1)$ 。

7-2 工作視窗

7-2-1 工作視窗的文字寫入

當RA8875 執行文字寫入功能時，文字寫入的邊界線將會被限制在一個名為工作視窗(Active Window) 的區塊內。文字寫入方向的初始值設定是由左到右，然後由上到下。當文字向右水平寫入時，碰到右邊界線時文字游標會跳到下一行的左邊界處。若下一行的位置超過最底部的邊界線，游標會跳到視窗開始的位置，亦即最左上角的邊界位置。關於工作視窗的文字寫入效果可以參考 圖 7-7。需注意的是，若文字寫入游標被設在工作視窗的邊界之外，文字仍然會寫在文字游標的位置，直到碰到右邊的邊界線或顯示區域的邊界。當碰到邊界時，文字寫入游標會換行，然後依照工作視窗規則，請參考 圖 7-8 的例子。

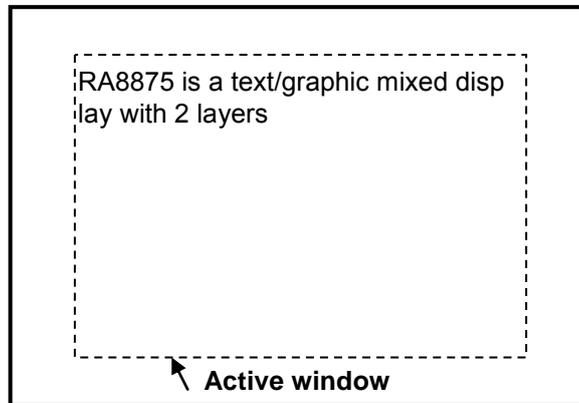


圖 7-7：工作視窗的文字寫入效果

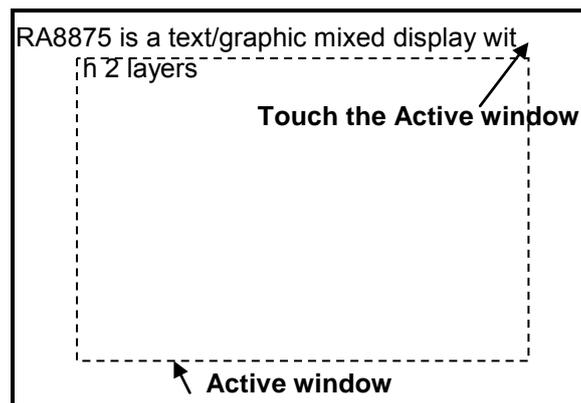


圖 7-8：當文字寫入游標在工作視窗外的文字寫入效果

7-2-2 工作視窗的幾何圖形顯示

工作視窗也控制幾何輸入的繪圖功能。只有工作視窗內的部份可以繪圖，請參考圖 7-9。

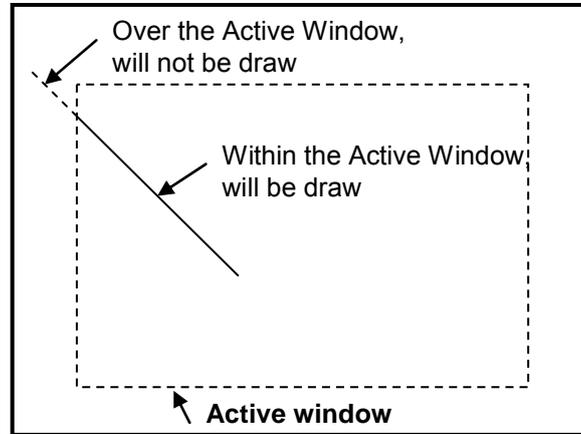


圖 7-9：在工作視窗中畫直線

註：工作視窗有關應用幾何功能應用上的例外條件如下：

1. 工作視窗不支援當畫橢圓。
2. 當畫圓的時候，假設圓心是 (X, Y) 且半徑為 R，在 $Y + R \geq 512$ 的情況下，工作視窗的限制沒有作用。

7-2-3 工作視窗中的DMA功能顯示

工作視窗也提供DMA的邊界線功能。DMA功能的目標是由工作視窗設定的。需注意DMA的來源為一區塊，若此區塊的定義大於工作視窗。超過工作視窗的部分會從工作視窗的起始點開始資料被覆蓋，有關詳細的內容請參考章節 7-10 的說明。

7-2-4 工作視窗的記憶體讀寫

當 RA8875 執行記憶體讀寫功能時，此功能的邊界線將會被工作視窗 (Active Window) 所控制。記憶體讀寫方向的初始值設定是由左到右，然後上到下。當記憶體讀寫時，碰到最右邊的邊界線時，讀寫位置游標會跳到下一行的左邊界線。若下一行的位置超過最底部的邊界線，游標會跳到視窗開始的位置。需注意，若記憶體讀寫游標設在工作視窗外的區塊，資料仍然會依游標的設定位置讀寫記憶體，直到碰到右邊的邊界線或顯示的邊界線。當碰到邊界時，記憶體讀寫游標會換行，然後依照工作視窗規則。

7-3 游標與圖形樣板

根據使用者不同的應用，RA8875 提供彈性且強大的游標與圖形樣板功能。RA8875 定義了四種不同的游標 — 圖形游標、記憶體讀取游標、記憶體寫入游標、文字寫入游標。圖形游標是一個 32x32 像素的圖形游標功能，可顯示在使用者所設定的位置上。當位置改變的時候，圖形游標就會移動。記憶體讀取游標與記憶體寫入游標是用在記憶體的讀取/寫入上，記憶體讀取/寫入游標會在記憶體資料寫入/讀出週期後自動移動，記憶體寫入游標定義了記憶體寫入資料的位置，記憶體讀取游標則定義記憶體讀取操作被讀取的位置。

記憶體寫入游標是一個資料被寫入的位置，記憶體讀取游標與記憶體寫入游標可被設定為自動移動或不被分開，移動方向也可單獨地設定，初始的的設定值為由左到右、由上到下自動的增加。需注意只有記憶體寫入游標是可見的，記憶體讀取游標在面板上不會顯示出來。文字寫入游標提供文字寫入功能有關的游標，此區塊的形狀、寬度與高度皆可透過程式設定。文字寫入游標的顯示位置，指的是文字正在寫入的位置。

另外 RA8875 也支援圖形樣板的機能，圖形樣板是個 8x8/16x16 像素的大小，顏色深度最多為 16bpp 的圖案，圖形樣板的顏色深度由 REG[10h] 的 Bit 3-2 所設定。經由與 BTE 引擎功能搭配，可以用於重複複製填滿某一個指定的區域，可以加速使用者對於重複動作的需求並降低 MCU 的負擔。

7-3-1 游標種類

7-3-1-1 圖形游標

圖形游標大小為 32x32 像素，每一像素由 2 個位元共 4 種顏色來設定，此 4 種顏色分別為 0 號顏色 (Color 0)、1 號顏色 (Color 1)、背景色與背景的反向色 (The inversion of background color)。每個圖形游標共需 256 bytes (32x32x2/8)。RA8875 內建記憶體提供使用者 8 款自訂圖形游標，可由暫存器來選擇或設定。圖形游標的顯示位置可以由暫存器 GCHP0 (REG[80h])、GCHP1(REG[81h])、GCVP0(REG[82h]) 和 GCVP1(REG[83h]) 設定。圖形游標的顏色可以由暫存器GCC0(REG[84h])、GCC1(REG[85h])、背景色、背景的反向色，依照圖形游標裡面的資料設定。請參考圖 7-10 的說明。

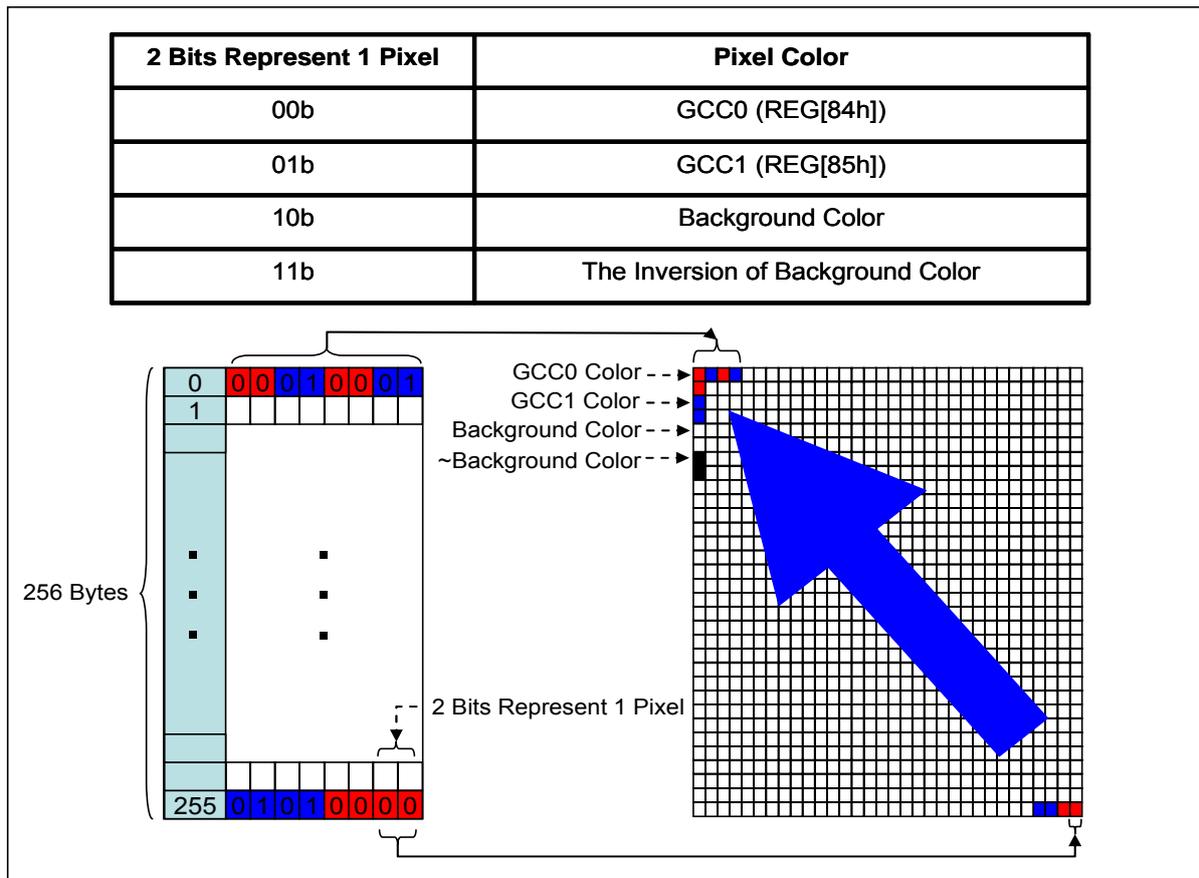


圖 7-10：記憶體對應和圖形游標的關聯

使用步驟：

1. 透過暫存器 GCC0[REG[84h] 與 GCC0[REG[85h].設定 GCC0 與 GCC1 顏色。
2. 透過 MWCR1(REG[41h]) 來設定圖形游標的編號並選擇寫入目標為圖形游標。
3. 使用繪圖模式來寫入圖形游標資料到圖形游標的儲存空間。
4. 開啟圖形游標功能(REG[41h] Bit7)。
5. 寫入GCHP0(REG[80h])、GCHP1(REG[81h])、GCVP0(REG[82h])、GCVP1(REG[83h]) 來改變圖形游標位置，請參考 圖 7-11 的顯示範例。

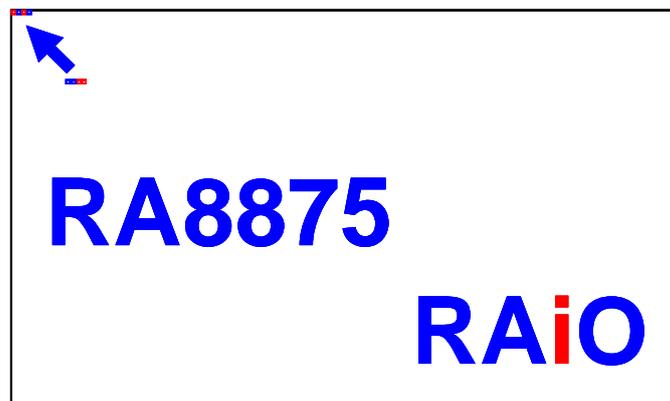


圖 7-11：圖形游標的顯示

7-3-1-2 記憶體讀取游標

記憶體讀取游標是記憶體讀取操作時的記憶體位置，記憶體讀取游標是不可見的。此游標的位置與記憶體寫入游標、文字寫入游標是獨立開來的。記憶體讀取游標可以被設為自動增加或非自動增加，並且游標可設定四種移動方向。需注意記憶體寫入游標在圖形模式或文字模式中，是可以使用的，請參考表 7-4 的說明。

表 7-4：記憶體讀取游標相關暫存器

Register Name	Bit Num	Function Description	Address
MWCR0	7	Text Mode Enable 0 : Graphic mode. 1 : Text mode.	[40h]
	0	Memory Read Cursor Auto-Increase Disable	
MRCR0	1-0	Memory Read Direction 00b : Left → Right then Top → Down. 01b : Right → Left then Top → Down. 10b : Top → Down then Left → Right. 11b : Down → Top then Left → Right.	[45h]
RCURH0/1	9-0	Memory Read Cursor Horizontal Location	[4Ah]、[4Bh]
RCURV0/1	8-0	Memory Read Cursor Vertical Location	[4Ch]、[4Dh]

7-3-1-3 記憶體寫入游標

記憶體寫入游標位在在圖形模式中記憶體寫入操作中的記憶體中，記憶體寫入游標是可見的。此游標的位置與記憶體寫入游標、文字寫入游標是獨立的。記憶體寫入游標可以被設為自動增加或非自動增加、閃爍或不閃爍。游標移動可設成四種方向，請參考表 7-5 的說明。

表 7-5：記憶體寫入游標相關的暫存器

Register Name	Bit Num	Function Description	Address
MWCR0	7	Text Mode Enable 0 : Graphic mode. 1 : Text mode.	[40h]
	6	Font Write Cursor/ Text Write Cursor Enable 0 : Font write cursor/ Text Write Cursor is not visible. 1 : Font write cursor/ Text Write Cursor is visible.	
	5	Font Write Cursor/ Text Write Cursor Blink Enable 0 : Normal display. 1 : Blink display.	
	3-2	Memory Write Direction (Only for Graphic Mode) 00b : Left → Right then Top → Down. 01b : Right → Left then Top → Down. 10b : Top → Down then Left → Right. 11b : Down → Top then Left → Right.	
	1	Memory Write Cursor Auto-Increase Disable 0 : Cursor auto-increases when memory write. 1 : Cursor doesn't auto-increases when memory write.	
CURH0/1	9-0	Memory Write Cursor Horizontal Location	[46h]、[47h]
CURV0/1	8-0	Memory Write Cursor Vertical Location	[48h]、[49h]

7-3-1-4 文字寫入游標

文字寫入游標是用在文字模式，是可見的。此游標的位置可以與記憶體讀取游標分開設定，與記憶體寫入游標類似，文字寫入游標可以被設為自動增加或非自動增加、閃爍或不閃爍。游標可以在工作視窗內自動移動。當在寫入文字時，游標會自動移動到下一個文字寫入的位置。依據文字的大小與文字方向，當碰到工作視窗的邊界線時，游標會自動換下一列。兩列之間的距離可以由像素 (Pixel)來設定。表 7-6 列出相關暫存器的描述。

表 7-6 : 文字寫入游標相關的暫存器

Register Name	Bit Num	Function Description	Address
FLDR	4-0	Font Line Distance Setting Register(FLDR)	[29h]
CURH0/1	9-0	Font Write Cursor Horizontal Location	[2Ah]、[2Bh]
CURV0/1	8-0	Font Write Cursor Vertical Location	[2Ch]、[2Dh]
MWCR0	7	Text Mode Enable 0 : Graphic mode. 1 : Text mode.	[40h]
	6	Font Write Cursor/Memory Write Cursor Enable 0 : Font write cursor/Memory Write Cursor is not visible. 1 : Font write cursor/Memory Write Cursor is visible.	
	5	Font Write Cursor/Memory Write Cursor Blink Enable 0 : Normal display. 1 : Blink display.	

7-3-2 游標屬性

7-3-2-1 游標閃爍

記憶體寫入游標與文字寫入游標可設為開啟、關閉或固定頻率的閃爍，皆由相同的暫存器來設定，此控制暫存器為 MWCR0(REG[40h])。閃爍的效果是重複游標的開啟(可見)、關閉(不可見)。閃爍的時間可以透過程式化設定，計算的公式如下，單位是秒 (Second)：

$$\text{Blink Time (sec)} = \text{BTCR}[44h] \times (1/\text{Frame_Rate}).$$

圖 7-12是游標閃爍的例子，游標閃爍的位置隨著最新的資料或文字寫入而移動。



圖 7-12 : 游標閃爍

7-3-2-2 游標的高度與寬度

除了圖形游標與記憶體讀取游標，另外兩種形式的游標是可以透過設定來設定高度與寬度。文字寫入游標的可設定寬度與高度組成一個區塊，控制的暫存器為CURHS(REG[4Eh])、CURVS(REG[4Fh])。記憶體寫入游標的形狀是一條線可以設定寬度，高度則固定為 1 像素。寬度的控制的暫存器與文字寫入游標相同，例如CURHS(REG[4Eh])，請參考圖 7-13 與圖 7-14。文字寫入游標的高度與寬度也與另外一個係數相關，那就是文字放大的設定暫存器(REG[2Eh] Bit3~0)。若放大的係數為 1，寬度就只透過CURHS/CURVS的設定為 1~32 像素。若放大的係數不是 1，則為實際的游標的寬度與高度必須再乘上這個放大係數。圖 7-13 為文字水平/垂直放大，係數為 1 的範例。需注意文字寫入游標不會被文字旋轉影響，若文字旋轉 90 度，文字寫入游標仍然會正常的情況相同。相關的顯示請參考圖 7-15 與圖 7-16。

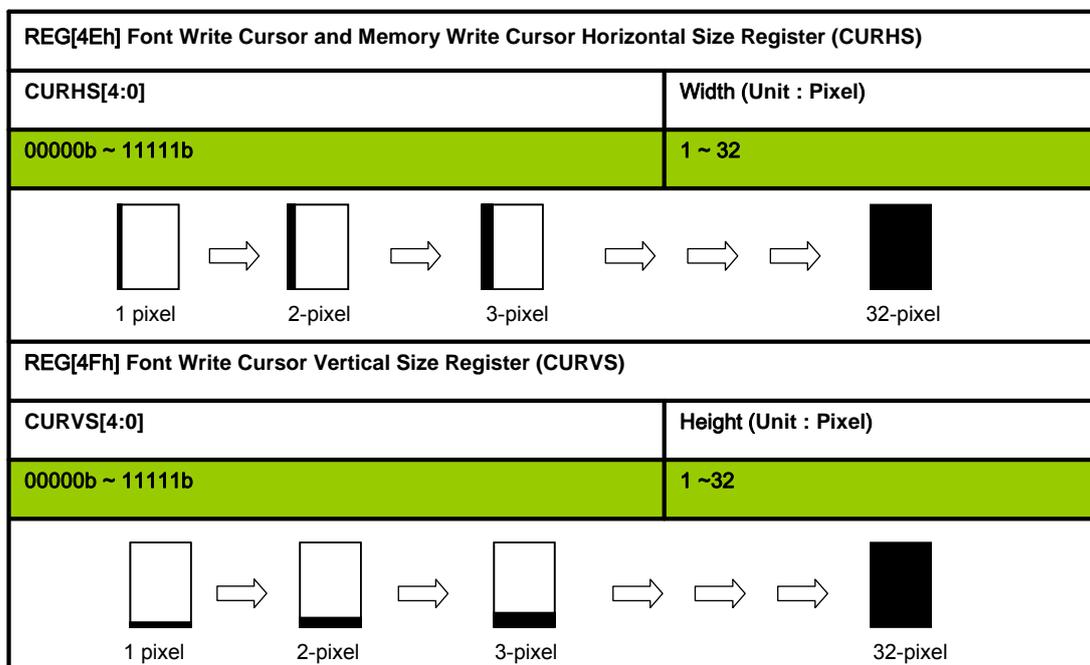


圖 7-13 : 文字寫入時游標高度與寬度的設定

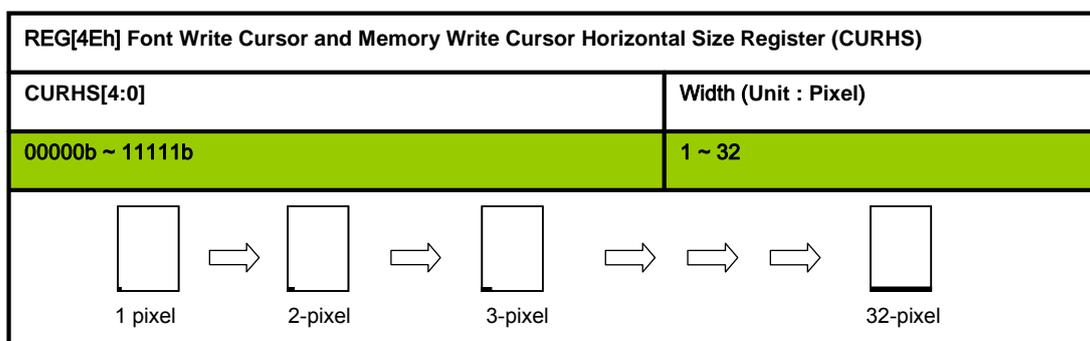


圖 7-14 : 計憶體寫入時游標寬度設定



圖 7-15 : 文字水平寫入時游標移動

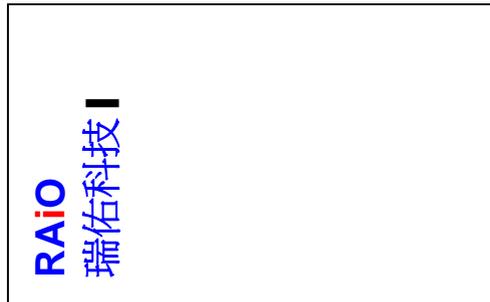


圖 7-16 : 文字垂直寫入時游標移動

7-3-3 圖形樣板

RA8875 內建樣板記憶體 (Pattern Memory) 可以寫入圖形樣板資料，記憶體的資料定義為圖形樣板資料 (Pattern Data)，是一個點陣圖的圖表。當 2D 相關的圖形樣板功能啟動時，指定的圖形樣板記憶體資料會填入指定的區域中。

使用者可以用 REG[41h] 來指定圖形樣板記憶體，而使用 REG[66h] 來設定圖形樣板的格式與編號。RA8875 支援 8x8/16x16 像素的圖形樣板樣式，如果圖形樣板為 8x8 像素，RA8875 可以依使用者需求最多定義 16 個樣板。如果圖形樣板為 16x16 像素，RA8875 可以依使用者需求最多定義 4 個樣板。圖形樣板的編號與格式會決定存取樣板的記憶體位置的安排。

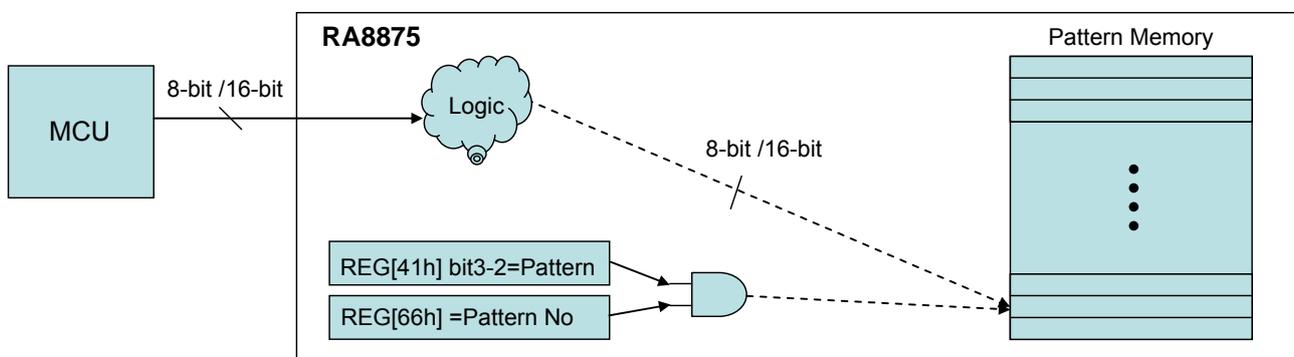


圖 7-17 : 8/16 位元的色彩模式

表 7-7 : 使用 Pattern 相關的暫存器

Register Name	Bit Num	Function Description	Address
MWCR1	3-2	Memory control register for setting pattern memory to access.	[41h]
PTNO	7-0	Pattern Number, the index of pattern for MCU to access pattern memory	[66h]

圖形樣板的詳細功能介紹，請參考章節 7-6 BTE 功能的說明。

7-4 文字

7-4-1 內部文字記憶體

RA8875 內建 8x16 點的 ASCII 字型 ROM，提供使用者更方便的方式用特定編碼 (Code) 輸入文字。內建的字集支援 ISO/IEC 8859-1~4 編碼標準，此外，使用者可以透過 REG[60h~62h] 選擇文字背景顏色，以及透過 REG[63h~65h] 選擇前景顏色，文字寫入的程序請參考下圖。

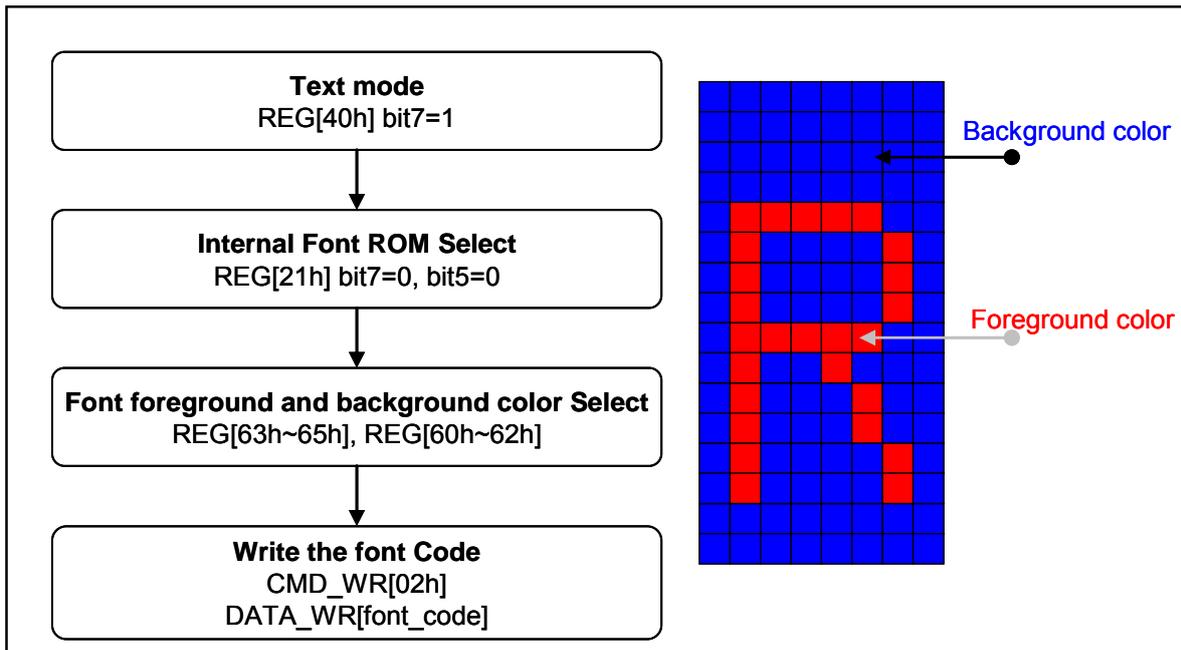


圖 7-18 : ASCII 字型 ROM 的寫入程序

表 7-8 內含ISO/IEC 8859-1 標準的字集。ISO是國際標準化組織的簡稱，ISO/IEC 8859-1 又稱 "Latin-1" 或「西歐語言」，是國際標準化組織內ISO/IEC 8859 的第一個發展的 8 位元字集。以ASCII為基礎，包含了 0xA0-0xFF的範圍內 192 個拉丁字母及符號。此字集編碼使用遍及西歐，包括阿爾巴尼亞語、巴斯克語、布列塔尼語、加泰羅尼亞語、丹麥語、荷蘭語、法羅語、弗裡斯語 (Frisian)、加利西亞語、德語、格陵蘭語、冰島語、愛爾蘭蓋爾語、義大利語、義大利語、拉丁語、盧森堡語、挪威語、葡萄牙語、里托羅曼斯語、蘇格蘭蓋爾語、西班牙語及瑞典語。

英語雖然沒有重音字母，但仍會標明為 ISO 8859-1 編碼，歐洲以外的部份語言，如南非荷蘭語、斯瓦希里語、印尼語及馬來語、菲律賓他加洛語 (Tagalog) 也可使用 ISO8859-1 編碼。

表 7-8 : ASCII 字碼表 1 (ISO/IEC 8859-1)

H/L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☹	♥	♦	♣	♠	●	◻	◯	◼	♂	♀	♪	♫	☼
1	▶	◀	↕	!!	¶	§	■	⬆	⬇	⬇	→	←	┌	↔	▲	▼
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
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	€	„	‚	ƒ	„	…	†	‡	•	‰	Š	‹	Œ		Ž	
9		„	‚	“	”	•	-	-	•	™	Š	›	œ		ž	ÿ
A		ı	ç	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
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
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

註：在表 7-8 中 0x80-0x9F 間的紅色字體，由瑞佑科技保留。

表 7-9 內為ISO/IEC 8859-2 的標準字集，又稱Latin-2 或「中歐語言」，是國際標準化組織內ISO/IEC 8859 的第二個 8 位字元集。此字元集主要支援以下文字：克羅埃西亞語、捷克語、匈牙利語、波蘭語、斯洛伐克語、斯洛維尼亞語、索布語。而阿爾巴尼亞語、英語、德語、拉丁語也可用此字元集顯示。芬蘭語中只有外來語才有â字元，若不考慮此字元，ISO/IEC8859-2 也可用於瑞士及芬蘭語。

表 7-9 : ASCII 字碼表 2 (ISO/IEC 8859-2)

H/L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☹	♥	♦	♣	♠	●	◻	◯	◼	♁	♀	🎵	🎶	☀
1	▶	◀	↕	!!	¶	§	■	↕	↑	↓	→	←	↔	▲	▼	
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
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																
A	SP	À	Á	Â	Ã	Ä	Å	Ā	Š	Ŝ	Š	Ť	Ž	Ž	Ž	Ž
B	°	à	á	â	ã	ä	å	ā	š	ŝ	š	ť	ž	ž	ž	ž
C	Ř	Á	Â	Ä	Å	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
D	Ď	Ň	Ň	Ó	Ô	Õ	Ö	×	Ř	Ů	Ú	Ů	Ů	Ý	Ť	ß
E	ř	á	â	ä	å	ā	ā	ā	ā	ā	ā	ā	ā	ā	ā	ā
F	ď	ň	ň	ó	ô	õ	ö	÷	ř	ů	ú	ů	ů	ý	ť	·

表 7-10 內為ISO/IEC 8859-3 之標準字集，又稱Latin-3 或「南歐語言」，是國際標準化組織內ISO/IEC 8859 的第三個 8 位字元集。它原先設計來表示土耳其語及馬爾他語文字，但土耳其語已改用ISO/IEC 8859-9 顯示，現時只有世界語及馬爾他語仍使用此字元集。此字元集同時能支援以下文字：英語、德語、義大利語、拉丁語及葡萄牙語。

表 7-10 : ASCII 字碼表 3 (ISO/IEC 8859-3)

H/L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☹	♥	♦	♣	♠	●	◻	◯	◼	♂	♀	♪	♫	☀
1	▶	◀	↕	!!	¶	§	■	⤴	↑	↓	→	←	↔	▲	▼	
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
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																
A	SP	Ħ	ˆ	ε	α		Ĥ	§	¨	ı	ş	ğ	ĵ			ž
B	°	ħ	²	³	ˆ	μ	ĥ	·	˙	ı	ş	ğ	ĵ	½		ž
C	À	Á	Â		Ä	Ĉ	Ċ	ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D		Ñ	Ò	Ó	Ô	Ĝ	Ö	×	Ĝ	Ù	Ú	Û	Ü	Ũ	Ŝ	ß
E	à	á	â		ä	ĉ	ċ	ç	è	é	ê	ë	ì	í	î	ï
F		ñ	ò	ó	ô	ĝ	ö	÷	ĝ	ù	ú	û	ü	ũ	ŝ	·

表 7-11 內為ISO/IEC 8859-4 之標準字集，又稱Latin-4 或「北歐語言」，是國際標準化組織內ISO/IEC 8859 的第四個 8 位字元集，它設計來表示愛沙尼亞語、格陵蘭語、拉脫維雅語、立陶宛語及部分薩米語 (Sami) 文字，此字元集同時能支援以下文字：丹麥語、英語、芬蘭語、德語、拉丁語、挪威語、斯洛維尼亞語及瑞典語。

表 7-11 : ASCII 字碼表 4 (ISO/IEC 8859-4)

H/L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☹	♥	♦	♣	♠	●	◻	◯	◼	♁	♀	♪	♫	☀
1	▶	◀	↕	!!	¶	§	■	↑	↓	→	←	↔	▲	▼		
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/	
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																
A	SP	À	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā	Ā
B	°	á	â	ã	ä	å	æ	ç	š	ē	ġ	t	D	ž	ŋ	
C	Ā	Á	Â	Ã	Ä	Å	Æ	Ç	Š	Ē	Ġ	Ė	Ė	Ė	Ė	Ė
D	Ð	Ñ	Ō	Ķ	Ō	Ō	Ö	×	Ø	Ū	Ú	Ū	Ū	Ū	Ū	β
E	ā	á	â	ã	ä	å	æ	ç	š	ē	ġ	Ė	Ė	Ė	Ė	Ė
F	đ	ñ	ō	ķ	ô	õ	ö	÷	ø	ų	ú	û	ü	Û	Ū	•

7-4-2 外部 Font ROM

RA8875 的外部串列 ROM 介面是一個彈性方式，在不同應用上提供更多的字集編碼。此介面相容於集通公司 (Genitop Inc) 的部分串列字型 ROM。支援產品編號包含: GT21L16TW、GT30L16U2W、GT30L24T3Y、GT30L24M1Z 與 GT30L32S4W。針對不同的產品，可提供 16x16、24x24、32x32 字型大小以及不同的字寬。

REG[06h] 提供使用者調整存取外部串列 Flash/ROM 週期的速度，才能與 ROM 需要的存取時間互相配合。是外部 Font ROM 的寫入程序請參考下圖：

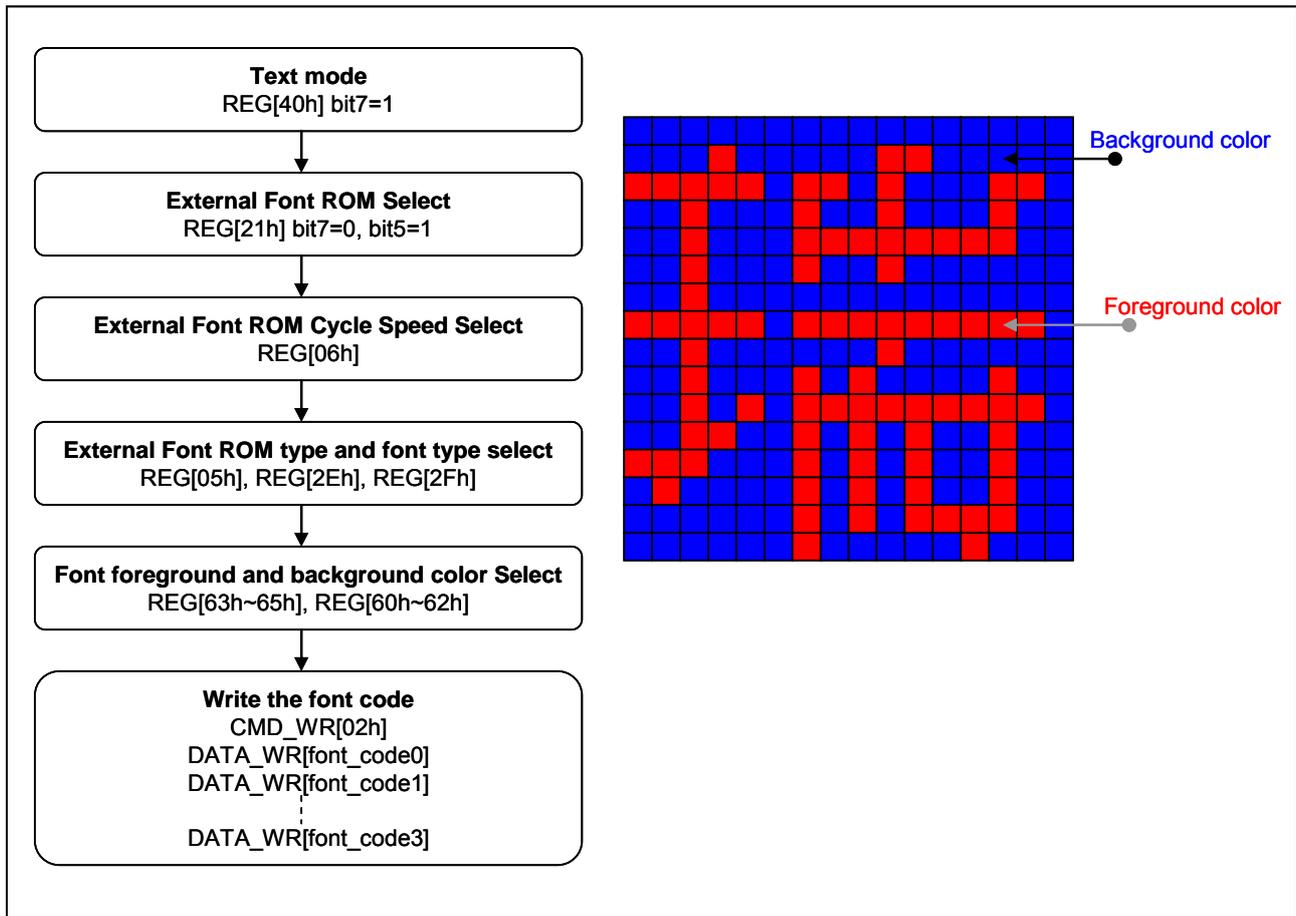


圖 7-19 : 外部 Font ROM 的寫入程序

7-4-3 CGRAM (Character Generation RAM 自建字庫功能)

RA8875 支援 CGRAM 功能，提供 256 個半型字的空間，讓使用者自己創造所要的字型或符號，使用者只要寫入字型或符號到指定字碼位置，然後再寫入相對應的字碼，RA8875 將可寫入字型或符號到 DDRAM。此外設定暫存器 REG[63h~65h]和 REG[60h~62h]可以選擇自建文字的前景顏色和背景顏色。寫入程序請參照下圖：

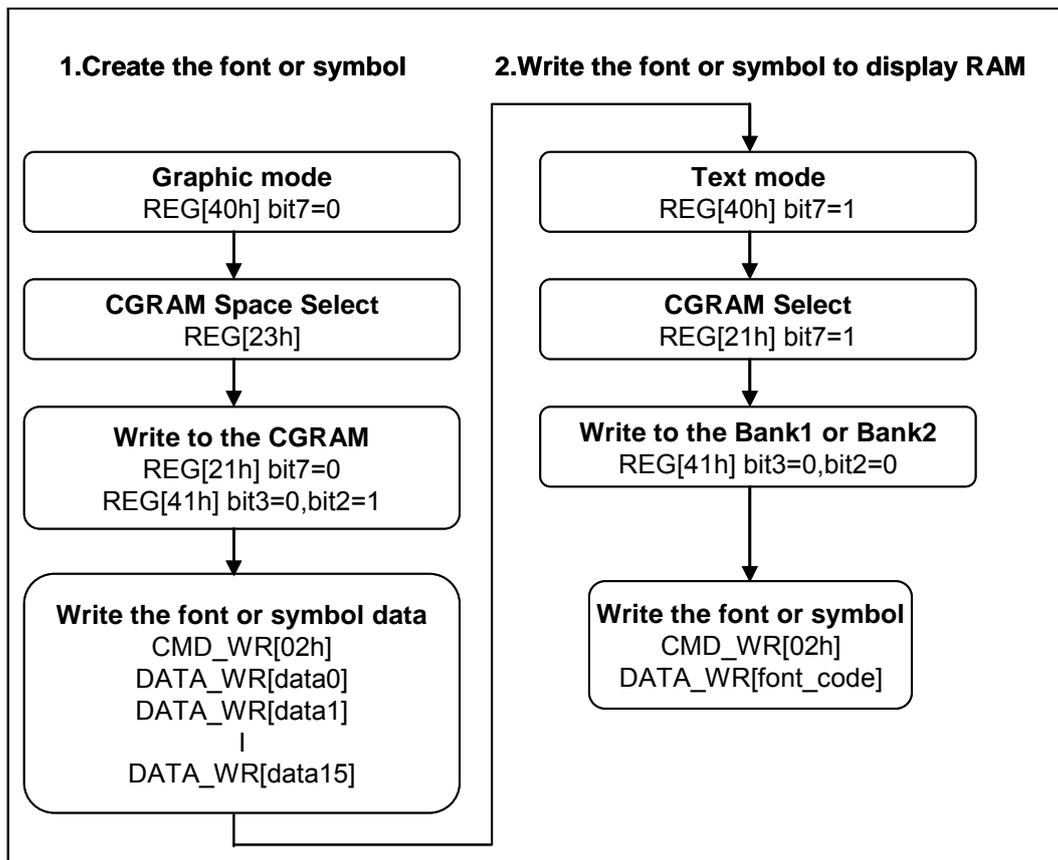


圖 7-20 : CGRAM 的寫入程序

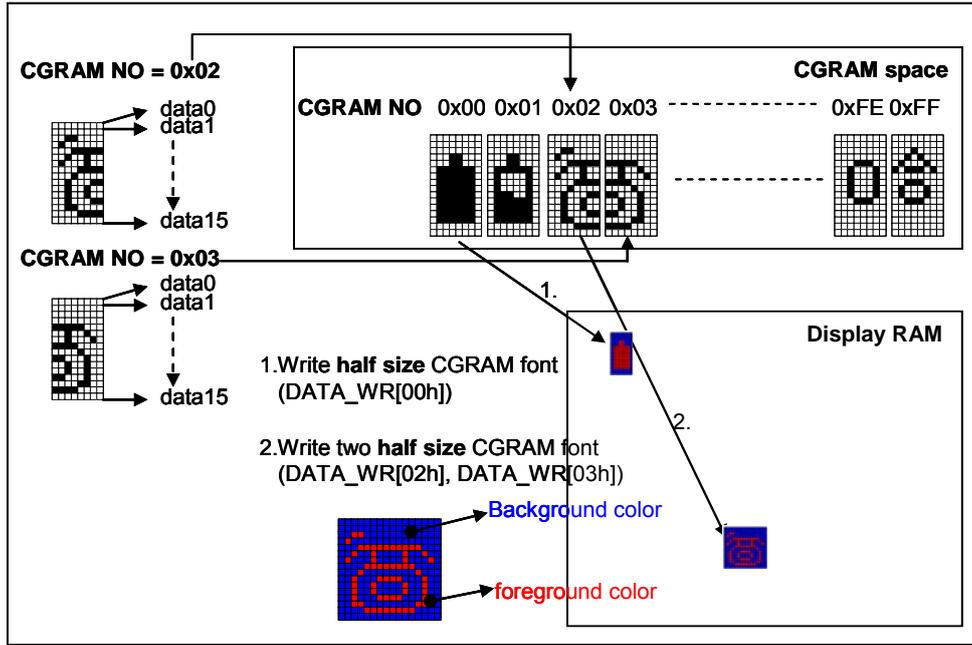


圖 7-21 : CGRAM 的寫入圖示

7-4-4 文字 90 度轉向

RA8875 藉由設定暫存器 REG[22h] Bit4 = 1，可支援文字 90 度轉向的顯示功能，如下圖所示。

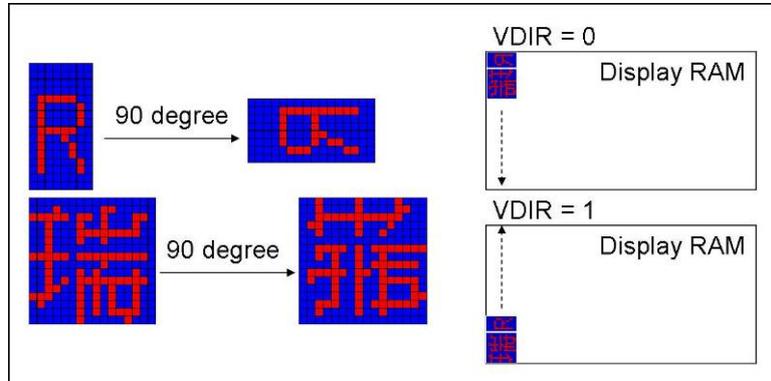


圖 7-22 : 文字 90 度轉向

7-4-5 文字放大與通透功能

設定暫存器 REG[22h] 的 Bit[3:0]，RA8875 支援文字放大功能；暫存器 REG[22h] 的 Bit6 可使用通透功能，而以上這幾個文字功能可同時使用，其顯示效果請參考下圖：

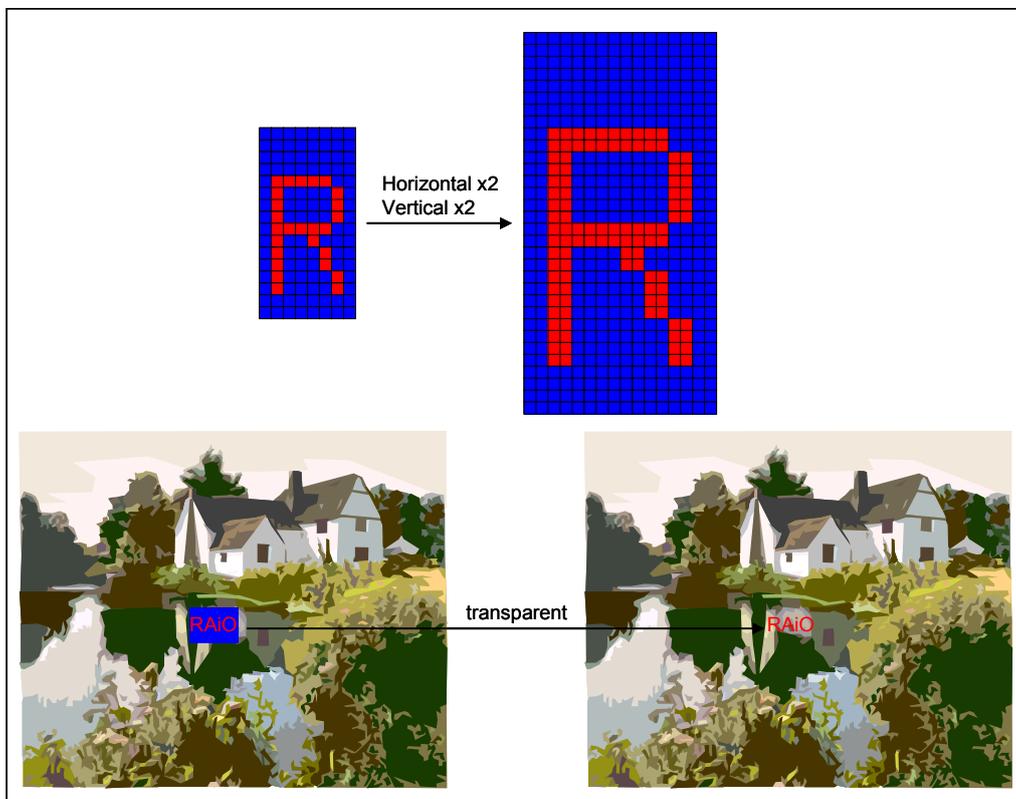


圖 7-23 : 文字放大與通透功能

7-4-6 文字換行

RA8875 支援文字在工作視窗中自動寫入且自動換行，也就是游標自動移位，透過暫存器 REG[40h] Bit1 = 0 設定，當文字超過水平或垂直工作視窗範圍時，文字會自動移動及換行，其顯示效果請參考下圖：

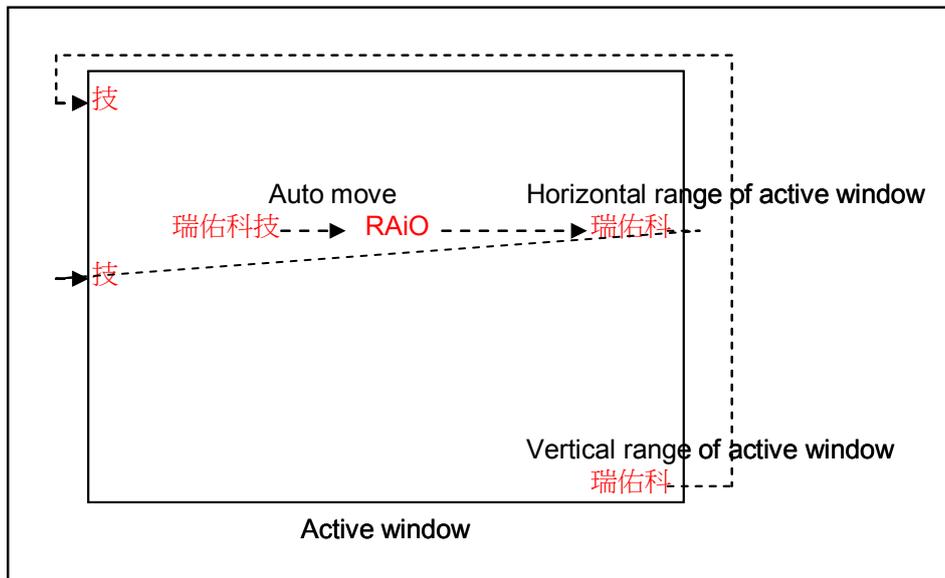


圖 7-24 : 文字自動換行

7-4-7 文字全型對齊

RA8875 支援文字全型對齊，暫存器設定為 REG[22h] Bit7 = 1 後，當寫入半型和全型文字到 DDRAM 時，可自動判斷並排列整齊，在文字的顯示視覺上比較好看。其寫入半型和全型文字的顯示效果如下圖：

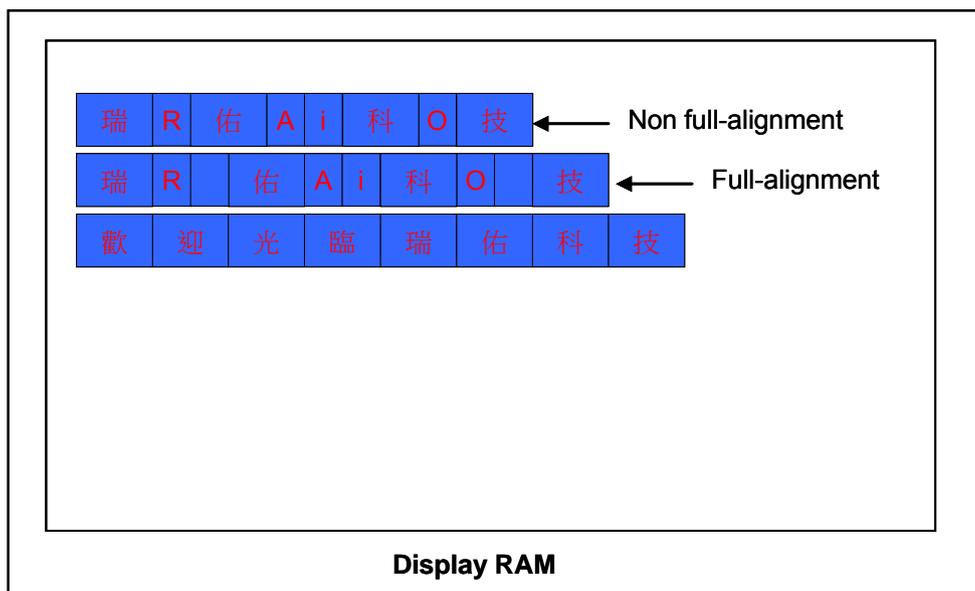


圖 7-25 : 文字對齊

7-5 幾何圖案繪圖引擎

7-5-1 圓形輸入

RA8875 支援圓形繪圖功能，讓使用者以簡易或低速的 MCU 就可以在 TFT 模組上畫圓。先設定圓的中心點 REG[99h~9Ch]，圓的半徑 REG[9Dh]，圓的顏色 REG[63h~65h]，然後啟動繪圖 REG[90h] Bit6 = 1，RA8875 就會將圓的圖形寫入 DDRAM，相對的在 TFT 模組上就可以顯示所畫的圓。若設定 REG[90h] Bit5 = 1，則可畫出一實心圓 (Fill)；若設定 REG[90h] Bit5 = 0，則可畫出空心圓 (Not Fill)，寫入程序請參照下圖：

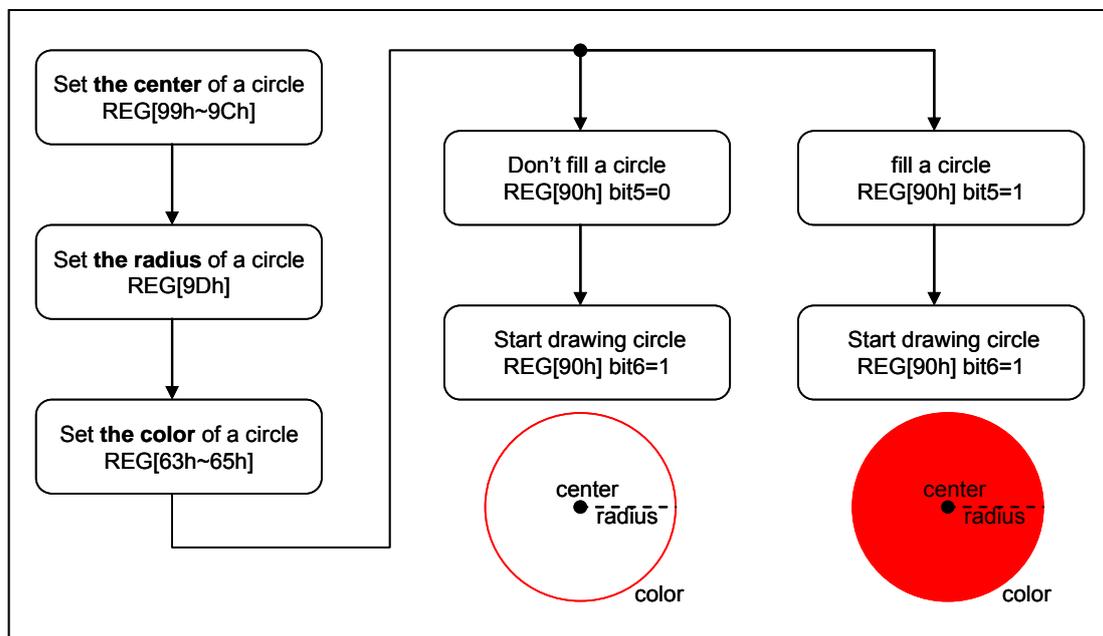


圖 7-26 : 繪圖功能 - 畫圓

7-5-2 橢圓輸入

RA8875 支援橢圓繪圖功能，讓使用者以簡易或低速的 MCU 就可以在 TFT 模組上畫橢圓。先設定橢圓的中心點 REG[A5h~A8h]，橢圓的長軸與短軸 REG[A1h~A4h]，橢圓的顏色 REG[63h~65h]，橢圓的相關參數 REG[A0h] Bit5=0 與 Bit4=0，然後啟動繪圖設定 REG[A0h] Bit7 = 1，RA8875 就會將橢圓的圖形寫入 DDRAM，相對的在 TFT 模組上就可以顯示所畫的橢圓。若設定 REG[A0h] Bit6 = 1，則可畫出一實心橢圓 (Fill)，寫入程序請參照下圖：

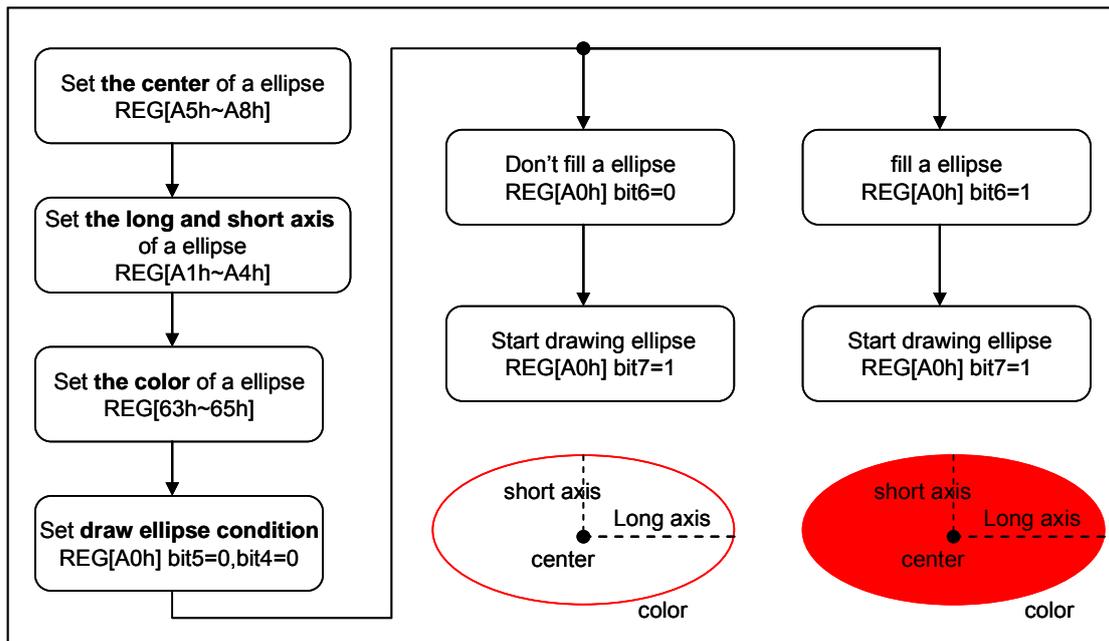


圖 7-27 : 繪圖功能 - 畫橢圓

7-5-3 曲線輸入

RA8875 支援曲線繪圖功能，讓使用者以簡易或低速的 MCU 就可以在 TFT 模組上畫曲線。先設定曲線的中心點 REG[A5h~A8h]，曲線的長軸與短軸 REG[A1h~A4h]，曲線的顏色 REG[63h~65h]，曲線的相關參數為 REG[A0h] Bit5=0 與 Bit4=1，REG[A0h] Bit[1:0] 是橢圓的曲線部份，然後啟動繪圖設定 REG[A0h] Bit7 = 1，RA8875 就會將橢圓的圖形寫入 DDRAM，相對的在 TFT 模組上就可以顯示所畫的曲線。若設定 REG[A0h] Bit6 = 1，則可畫出一實心曲線 (Fill)，寫入程序請參照下圖：

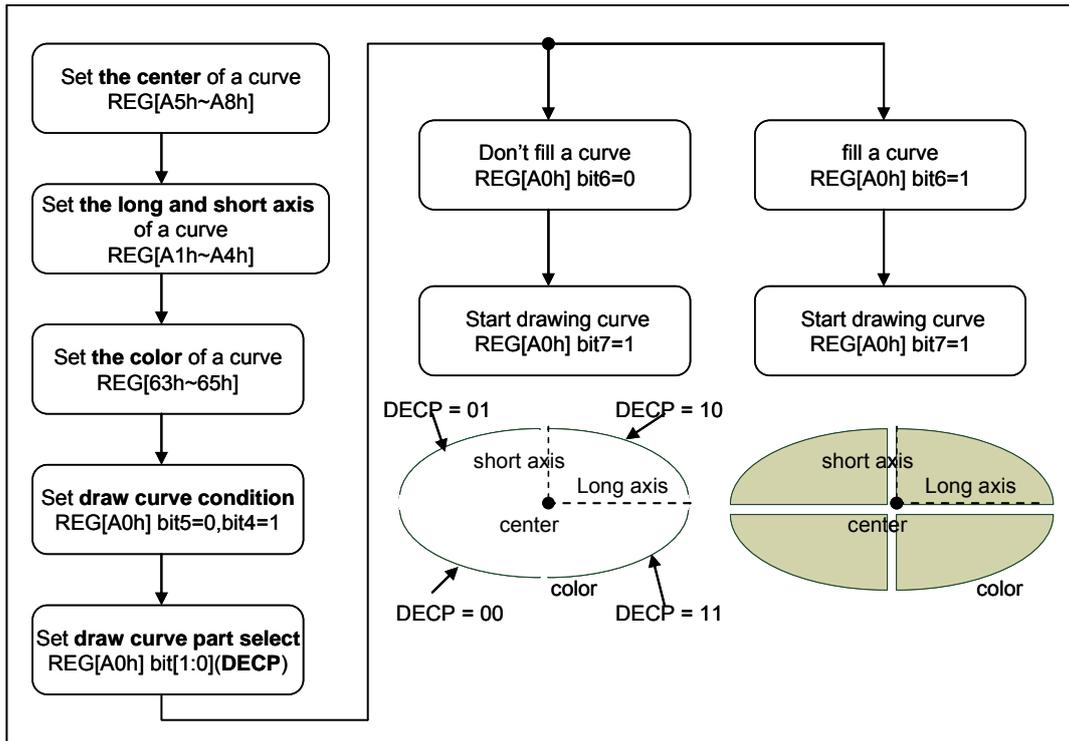


圖 7-28 : 繪圖功能 - 畫曲線

7-5-4 方形輸入

RA8875 支援方形繪圖功能，讓使用者以簡易或低速的 MCU 就可以在 TFT 模組上畫方形。先設定方形的起始點 REG[91h~94h]與結束點 REG[95h~98h]，方形的顏色 REG[63h~65h]，然後啟動繪圖設定 REG[90h] Bit4=1, Bit0=0 且 REG[90h] Bit7 = 1，RA8875 就會將方形的圖形寫入 DDRAM，相對的在 TFT 模組上就可以顯示所畫的方形。若設定 REG[90h] Bit5 = 1，則可畫出一實心方形 (Fill)，寫入程序請參照下圖：

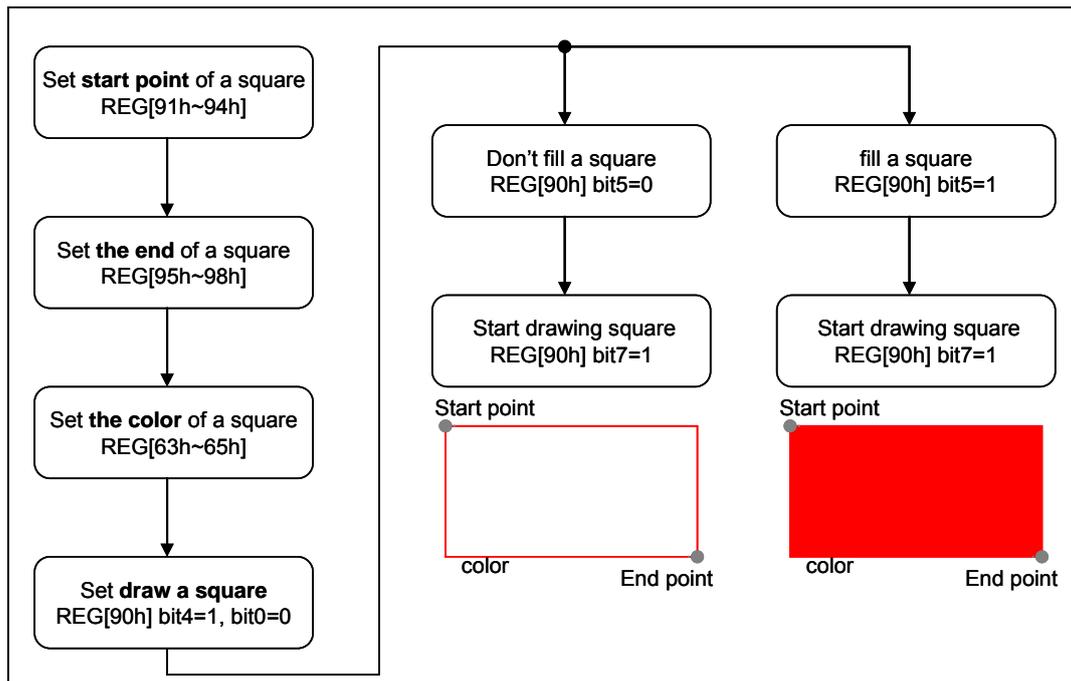


圖 7-29：繪圖功能 – 畫方形

註：起始點與終點位置不相同

7-5-5 直線輸入

RA8875 支援直線繪圖功能，讓使用者以簡易或低速的 MCU 就可以在 TFT 模組上畫直線。先設定直線的起始點 REG[91h~94h] 與結束點 REG[95h~98h]，直線的顏色 REG[63h~65h]，然後啟動繪圖設定 REG[90h] Bit4 = 0, Bit0=0 且 REG[90h] Bit7 = 1，RA8875 就會將直線的圖形寫入 DDRAM，相對的在 TFT 模組上就可以顯示所畫的直線。寫入程序請參照下圖：

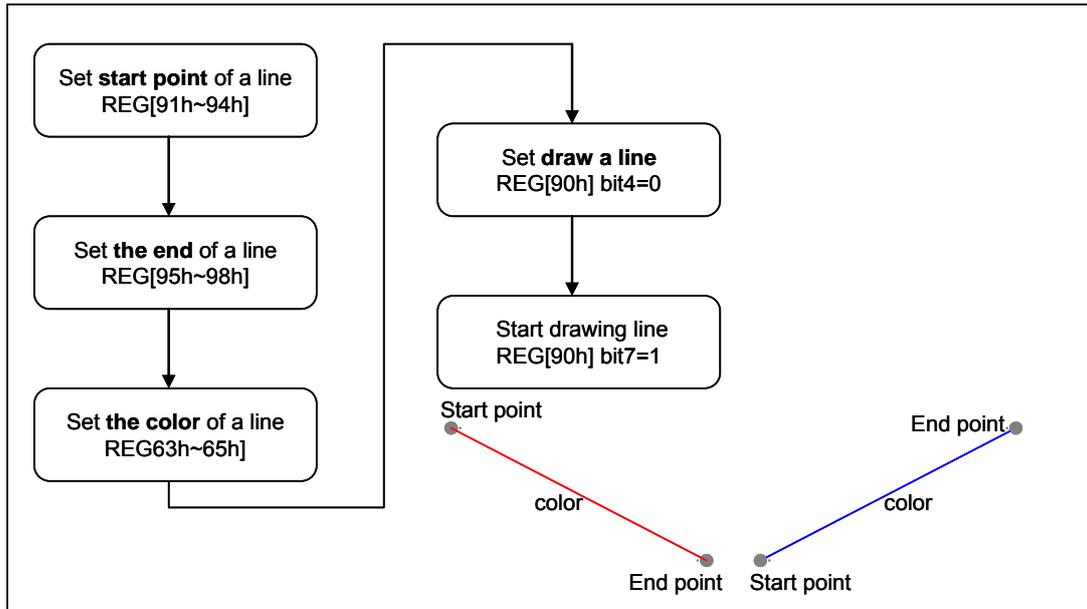


圖 7-30：繪圖功能 - 畫直線

註：起始點與終點位置不相同

7-5-6 三角形輸入

RA8875 支援三角形繪圖功能，讓使用者以簡易或低速的 MCU 就可以在 TFT 模組上畫三角形。先設定三角形的第 0 點 REG[91h~94h]、第 1 點 REG[95h~98h]、第 2 點 REG[A9h~ACh]，三角形的顏色 REG[63h~65h]，然後啟動繪圖設定 REG[90h] Bit0 = 1 且 REG[90h] Bit7 = 1，RA8875 就會將三角形的圖形寫入 DDRAM，相對的在 TFT 模組上就可以顯示所畫的三角形。若設定 REG[90h] Bit5 = 1，則可畫出一實心三角形 (Fill)，寫入程序請參照下圖：

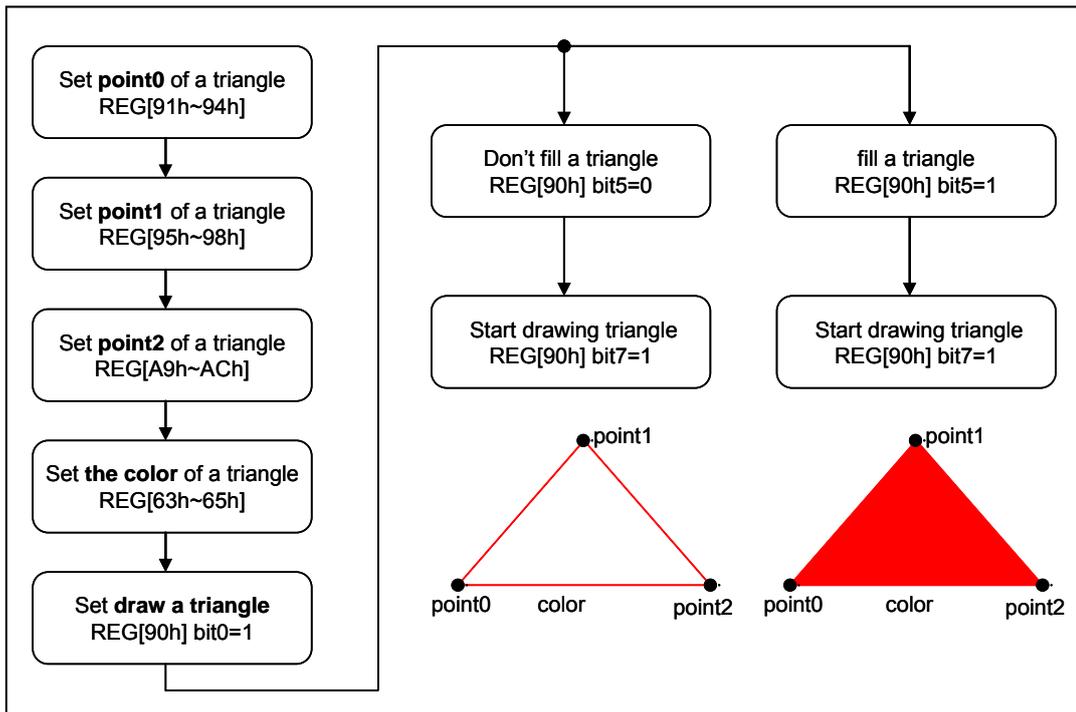


圖 7-31：繪圖功能 - 畫三角形

7-5-7 圓角方形輸入

RA8875 支援圓角方形繪圖功能，讓使用者以簡易或低速的 MCU 就可以在 TFT 模組上畫圓角方形。先設定圓角方形的起始點 REG[91h~94h]、結束點 REG[95h~98h]、圓角 REG[A1h~A4h]，圓角方形的顏色 REG[63h~65h]，然後啟動繪圖設定 REG[A0h] Bit5=1 且 REG[A0h] Bit7 = 1，RA8875 就會將圓角方形的圖形寫入 DDRAM，相對的在 TFT 模組上就可以顯示所畫的圓角方形。若設定 REG[A0h] Bit6 = 1，則可畫出一實心圓角方形 (Fill)，寫入程序請參照下圖：

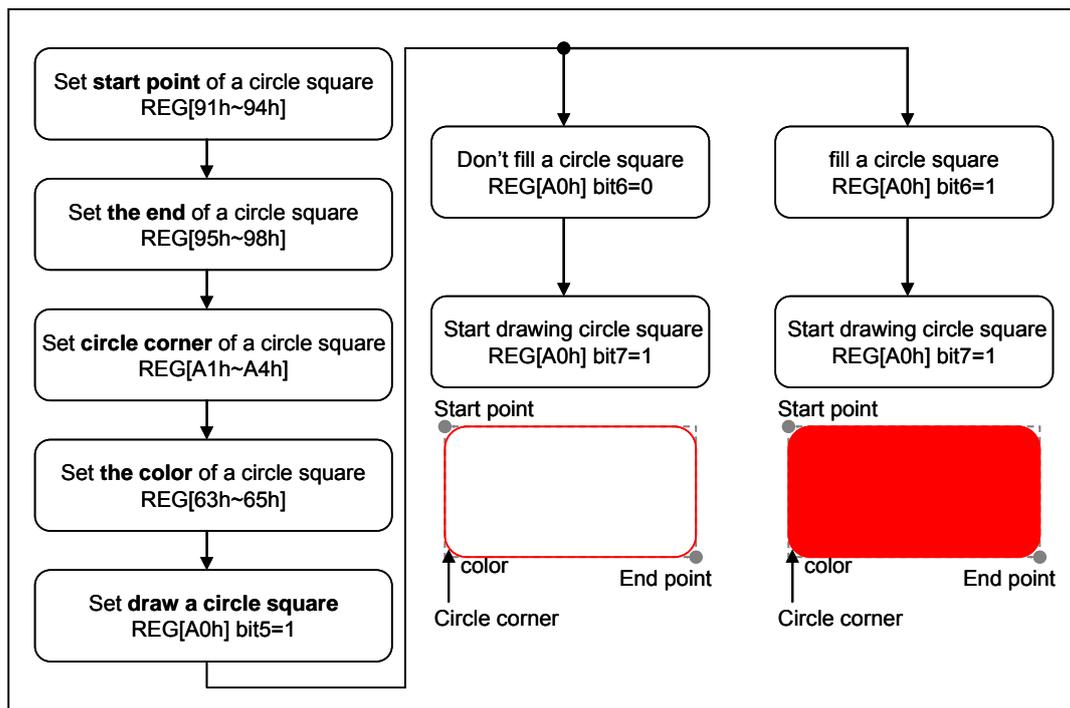


圖 7-32 : 繪圖功能 – 畫圓角方型

7-6 BTE 引擎功能

RA8875 內建一 2D 加速引擎功能，稱為 BTE (Block Transfer Engine)，可增強區塊資料處理的效率。當區塊性資料需要搬移或需特定邏輯處理時，可透過 RA8875 的 BTE 功能快速地完成且可簡化 MCU 的程序。此 BTE 相容於 2D BitBLT 標準功能，而本節將討論 BTE 引擎的運作和功能。

在使用 BTE 引擎功能之前，使用者必須先設定相對應的 BTE 操作碼，選擇想要的操作模式。RA8875 支援 13 種 BTE 操作模式。關於 BTE 引擎操作碼說明，請參考 表 7-12。對於每一種 BTE 操作模式，可搭配最多 16 種的光柵運算碼 (ROP, Raster Pperations)，提供以區塊為範圍的多功能的邏輯運算。光柵運算來源 (ROP Source) 和光柵目的地 (ROP Destination) 可提供不同的邏輯組合，透過 BTE 操作碼及光柵運算碼的組合，使用者可實現許多有用的運用。光柵運算的來源與目的地的設定也提供彈性的方式，使用者可以設定為方形的顯示區域 (區塊模式)，或連續記憶體區塊 (線性定址模式)。關於 BTE 的操作，請參考下面章節的進一步詳細說明。

BTE 引擎共有二種方式來確認 BTE 處理過程的完成，一是處理軟體輪詢 (Polling) 來確認是否忙碌，另一個是使用硬體中斷 (Interrupt)。當 BTE 引擎在處理過程中，狀態暫存器裡的 BTE 忙碌旗標會被設定，藉以反應 BTE 操作完成與否，請參考章節 5-1 的狀態暫存器。硬體中斷 (INT#) 是另一種可確認 BTE 過程結束的方式，使用者可先設定 REG[F0h]，若 BTE 操作完成，RA8875 將發出硬體中斷通知 MCU，MCU 便可藉由檢查中斷狀態去確定 BTE 引擎的狀態。當 BTE 引擎尚未完成運轉前，除了 REG[02h] 或 REG[F1h] 外，使用者不可寫入指令給 RA8875，以免影響正確的顯示結果。而且使用 BTE 時必須在繪圖模式下進行，也就是暫存器 REG [40h] Bit7 = 0。

表 7-12 : BTE 操作碼動作說明

BTE Operation REG[51h] Bits [3:0]	BTE Operation
0000b	Write BTE with ROP. Please refer to 表 7-13.
0001b	Read BTE.
0010b	Move BTE in positive direction with ROP. Please refer to 表 7-13.
0011b	Move BTE negative direction with ROP. Please refer to 表 7-13.
0100b	Transparent Write BTE.
0101b	Transparent Move BTE in positive direction.
0110b	Pattern Fill with ROP. Please refer to 表 7-13.
0111b	Pattern Fill with transparency.
1000b	Color Expansion. Please refer to 表 7-14
1001b	Color Expansion with transparency. Please refer to 表 7-14.
1010b	Move BTE with Color Expansion. Please refer to 表 7-15.
1011b	Move BTE with Color Expansion and transparency. Please refer to 表 7-15.
1100b	Solid Fill.
Other combinations	Reserved

表 7-12 說明RA8875 支援 13 種BTE操作模式，其中BTE操作碼為“0000”、“0010”、“0011”、“0110”時必須配合光柵運算碼，才能知道詳細的動作，請參考表 7-13。

表 7-13 : 光柵運算 (ROP) 功能 (1)

ROP Bits REG[51h] Bit[7:4]	Boolean Function Operation
0000b	0 (Blackness)
0001b	$\sim S \cdot \sim D$ or $\sim (S+D)$
0010b	$\sim S \cdot D$
0011b	$\sim S$
0100b	$S \cdot \sim D$
0101b	$\sim D$
0110b	$S^{\wedge}D$
0111b	$\sim S+\sim D$ or $\sim (S \cdot D)$
1000b	$S \cdot D$
1001b	$\sim (S^{\wedge}D)$
1010b	D
1011b	$\sim S+D$
1100b	S
1101b	$S+\sim D$
1110b	S+D
1111b	1 (Whiteness)

註：

上述 ROP 功能“S”代表來源資料，“D”代表目的資料。以圖形樣板填入 (Pattern Fill) 功能為例，來源資料表示圖形樣板資料。

範例：

若 ROP 功能設定 Ch，則目的資料 = 來源資料

若 ROP 功能設定 Eh，則目的資料 = S + D

若 ROP 功能設定 2h，則目的資料 = $\sim S \cdot D$

若 ROP 功能設定 Ah，則目的資料 = 目的資料

表 7-14 : 光柵運算功能 (2)

ROP Bits REG[51h] Bit[7:4]	Start Bit Position for Color Expansion BTE operation code = 1000 / 1001	
	16-bit MCU Interface	8-bit MCU Interface
0000b	Bit0	Bit0
0001b	Bit1	Bit1
0010b	Bit2	Bit2
0011b	Bit3	Bit3
0100b	Bit4	Bit4
0101b	Bit5	Bit5
0110b	Bit6	Bit6
0111b	Bit7	Bit7
1000b	Bit8	Invalid
1001b	Bit9	Invalid
1010b	Bit10	Invalid
1011b	Bit11	Invalid
1100b	Bit12	Invalid
1101b	Bit13	Invalid
1110b	Bit14	Invalid
1111b	Bit15	Invalid

表 7-15 : 光柵運算功能 (3)

ROP Bits REG[51h] Bit[7:4]	Start Bit Position for Move Color Expansion BTE operation code = 1010 / 1011	
	Color Depth = 65Kcolors	Color Depth = 256 colors
0000b	Bit0	Bit0
0001b	Bit1	Bit1
0010b	Bit2	Bit2
0011b	Bit3	Bit3
0100b	Bit4	Bit4
0101b	Bit5	Bit5
0110b	Bit6	Bit6
0111b	Bit7	Bit7
1000b	Bit8	Invalid
1001b	Bit9	Invalid
1010b	Bit10	Invalid
1011b	Bit11	Invalid
1100b	Bit12	Invalid
1101b	Bit13	Invalid
1110b	Bit14	Invalid
1111b	Bit15	Invalid

7-6-1 選擇BTE起始點位置及圖層

在雙層顯示的組態下，光柵運算的來源和目的資料可以選擇從哪一個圖層提供。要設定光柵運算的來源或目的前，要先設定水平和垂直起始點位置，請參考暫存器 VSBE0/1 和 VDBE0/1。圖層的選擇也請參考 VSBE1 Bit[7] 與 VDBE1 Bit[7]，VSBE1 Bit[7] 則用來設定光柵運算目的的圖層。

7-6-2 BTE 操作說明

7-6-2-1 BTE寫入

BTE 寫入功能提供 16 種雙運算元 (2 Operands) 的光柵運算。BTE 會將光柵運算的結果寫入目的位置。

7-6-2-2 BTE讀取

BTE 讀取功能支援資料從來源位置讀取資料至 MCU 主機端的功能。此功能不需考慮光柵運算。

7-6-2-3 BTE移動

BTE 移動功能提供 16 種雙運算元 (2 Operands) 的光柵運算。此功能也支援正向與反向移動的選擇。

7-6-2-4 單色填滿

單色填滿 BTE 功能提供使用者可將特定的區域 (BTE 來源區域) 以特定顏色 BTE 前景色填滿的功能。

7-6-2-5 圖案填滿

圖案填滿功能提供使用者將特定的 BTE 區域以特定的 8*8 像素圖案填滿的功能，此圖案被設定於顯示範圍外的 DDRAM 中。

7-6-2-6 BTE通透填滿

BTE 通透填滿功能提供將特定的 BTE 區域以特定的 8*8 像素圖案填滿的功能，此圖案被設定於顯示範圍外的 DDRAM 中。當圖案中的顏色與特定的顏色相同時，在此功能設定為 BTE 前景色 (定義於 BTE Foreground Color 暫存器)，目的區域的資料變不會被覆蓋，會保持通透性，此功能不需考慮光柵運算。

7-6-2-7 BTE 通透寫入

BTE 通透寫入功能支援從主機端寫入字元區塊至 DDRAM 區域的功能。當資料來源中的顏色與特定的顏色相同時，在此功能設定為 BTE 前景色 (定義於 BTE Foreground Color 暫存器)，目的區域的資料便不會被覆蓋，會保持通透性，此功能不需考慮光柵運算。

7-6-2-8 BTE 通透移動

BTE 通透移動功能支援從 DDRAM 來源以正向方式寫入字元區塊至 DDRAM 區域的功能。當資料來源中的顏色與特定的顏色相同時，在此功能設定為 BTE 前景色 (定義於 BTE Foreground Color 暫存器)，目的區域的資料便不會被覆蓋，會保持通透性，此功能不需考慮光柵運算。

7-6-2-9 顏色擴充

BTE 顏色擴充功能可將主機端的單色資料，以 8 或 16 位元的顏色深度格式擴充，並寫入 DDRAM 中，擴充方式如下：

- ◆ 資料"1" 擴充為 BTE 前景色暫存器中設定的顏色。
- ◆ 資料"0" 擴充為 BTE 背景色暫存器中設定的顏色。

若背景通透格式被開啟，目標顏色將會維持不變。

7-6-2-10 顏色移動

BTE 顏色移動功能可將不在顯示範圍中的單色資料，以 8 或 16 位元的顏色深度格式移動寫入 DDRAM 中，若記憶體資料為"1" 則擴充為 BTE 前景色暫存器中設定的顏色，若記憶體中資料來源為"0" 擴充為 BTE 背景色暫存器中設定的顏色，若背景通透格式被開啟，目標顏色將會維持不變。

7-6-3 BTE記憶體存取方式

BTE 引擎有兩種存取記憶體，分別是區塊記憶體和線性記憶體，其範圍及大小的設定定義於暫存器 REG[5Ch]、[5Dh]、[5Eh] 和 [5Fh]。有關這兩種記憶體讀取方式的說明，請參考下面章節。

7-6-3-1 區塊記憶體讀取

使用此設定，BTE記憶體來源/目的資料會被視為一個顯示區域中的區塊，區塊寬度和高度定義於REG[5Ch-5Fh]，圖 7-33 範例表示來源和目的資料皆被設定為區塊記憶體讀取方式。

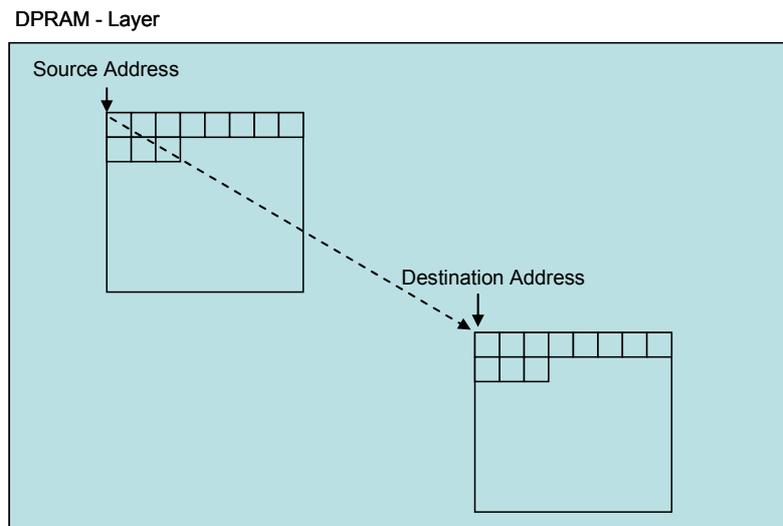


圖 7-33 : BTE 區塊記憶體讀取

7-6-3-2 線性記憶體讀取

使用此設定，BTE記憶體來源/目的資料會被視為是一個顯示區域中的連續定址區域，其區域長度定義於REG[5Ch-5Fh]，其中長度計算為(BTE_WIDTH x BTE_HEIGHT)，圖 7-34 範例表示來源和目的資料皆被設定為連續記憶體讀取方式。

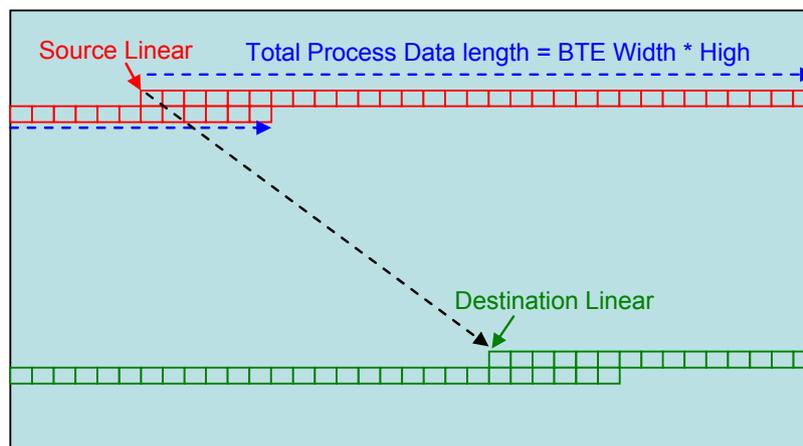


圖 7-34 : BTE 的線性記憶體讀取

7-6-4 BTE 功能說明

7-6-4-1 BTE寫入搭配光柵運算功能

BTE 寫入功能可加快 MCU 到 DDRAM 的資料傳送速度。BTE 寫入搭配光柵運算功能可將 MCU 寫入的資料，經過光柵運算後，填入指定的 DDRAM 位置。BTE 寫入功能支援全部的 16 種光柵運算，也支援目標記憶體線性模式和目標記憶體區塊模式。BTE 寫入功能的資料來源則由 MCU 提供。

使用者可使用硬體中斷或軟體忙碌確認的方式來得知 BTE 執行過程狀況。若使用者採取軟體方式，可以讀取暫存器 BECR0 (REG[50h]) 的 Bit7 或是由狀態暫存器 (STSR) 的 Bit6 的狀態而得知。若使用者採硬體中斷方式，必須確認 INT# 腳位必須連接到 MCU 的中斷腳位，再用中斷暫存器 REG[F1h] 來確認中斷的來源是否為 BTE，以得知 BTE 功能是否完成。

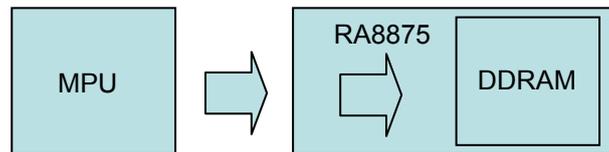


圖 7-35 : BTE 寫入搭配光柵運算功能

以下為 BTE 搭配光柵運算功能執行的建議步驟，請參考以下暫存器設定。

1. 設定目的位置 → REG[58h], [59h], [5Ah], [5Bh]
2. 設定 BTE 寬度暫存器 → REG[5Ch], [5Dh]
3. 設定 BTE 高度暫存器 → REG[5Eh], [5Fh]
4. 設定光柵運算為目的 = 來源 → REG[51h] = Ch
5. 開啟 BTE 功能 → REG[50h] Bit7 = 1
6. 致能 REG[02h]
7. 檢查 STSR Bit7
8. 寫入下一筆影像資料
9. 繼續第 7 和第 8 步驟直到影像資料(資料筆度 = 長度*寬度) 寫完或是檢查 STSR 的 Bit6 來確定資料是否全部寫入



圖 7-36 : BTE 功能完成畫面

以下步驟採用中斷方式來進行 BTE 狀態確認，使用此方式必須先行將中斷信號 INT# 與 MCU 進行連接。

1. 設定 INTC1 暫存器 → REG[F0h]
2. 設定目的位置 → REG[58h], [59h], [5Ah], [5Bh]
3. 設定 BTE 寬度暫存器 → REG[5Ch], [5Dh]
4. 設定 BTE 高度暫存器 → REG[5Eh], [5Fh]
5. 設定光柵運算為目的 = 來源 → REG[51h] = C0h
6. 開啟 BTE 功能 → REG[50h] Bit7 = 1
7. 偵測中斷信號產生
8. 檢查得知 BTE 讀寫中斷，並清除中斷狀態暫存器 → REG[F1h] Bit0 = 1
9. 致能 REG[02h]
10. 寫入影像資料
11. 偵測中斷信號產生
12. 檢查得知 BTE 讀寫中斷，並清除中斷狀態暫存器 → REG[F1h] Bit0 = 1
13. 繼續步驟 9~12 直到影像資料全部寫入(資料筆度 = 長度*寬度) 或是由 STSR Bit6 來確定所有資料是否全數寫入。

7-6-4-2 BTE讀取功能

BTE 讀取功能可加速從 DDRAM 到 MCU 的資料傳送速度，動作類似 Burst Read 功能。此功能一般用來加速將部分資料由 DDRAM 到系統記憶體搬移的動作，一旦 BTE 讀取功能開始，BTE 引擎會持續提供 DDRAM 資料給 MCU 讀取，直到所有的資料都被讀取完畢，BTE 處理資料的筆數由 REG[5Ch-5Fh] 來設定 (BTE_WIDTH x BTE_HEIGHT)。

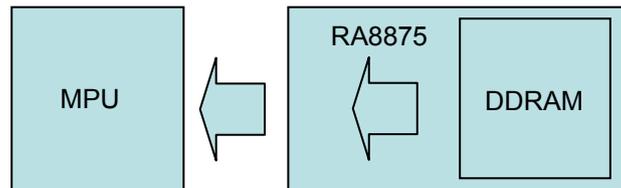


圖 7-37 : BTE 讀取功能

以下為 BTE 讀取功能執行時建議的步驟，請參考以下暫存器設定。

1. 設定來源位置 → REG[54h], [55h], [56h], [57h]
2. 設定 BTE 寬度暫存器 → REG[5Ch], [5Dh]
3. 設定 BTE 高度暫存器 → REG[5Eh], [5Fh]
4. 設定 BTE 控制暫存器 → REG[51h] = 01h
5. 開啟 BTE 功能 → REG[50h] Bit7 = 1
6. 致能 REG[02h]
7. 檢查 STSR Bit7
8. 讀取下一個圖像資料
9. 繼續步驟 6、7 直到圖像資料全部被讀出

以下步驟採用中斷方式來進行 BTE 狀態確認，使用此方式必須先將中斷信號 INT# 與 MCU 進行連接。

1. 設定 INT# → REG[F0h]
2. 設定來源位置 → REG[54h], [55h], [56h], [57h]
3. 設定 BTE 寬度暫存器 → REG[5Ch], [5Dh]
4. 設定 BTE 高度暫存器 → REG[5Eh], [5Fh]
5. 設定 BTE 控制暫存器 → REG[51h] = 01h
6. 開啟 BTE 功能 → REG[50h] Bit7 = 1
7. 等待中斷產生
8. 致能 REG[02h]
9. 讀取下一個圖像資料
10. 得知 BTE 讀寫中斷，並清除中斷狀態暫存器 → REG[F1h] Bit1 = 1
11. 繼續步驟 7~10 直到圖像資料全部被讀出，或由 TSR Bit6 來確定所有資料是否全數讀出

7-6-4-3 BTE正向移動搭配光柵運算功能

BTE 正向移動搭配光柵運算功能可執行將 DDRAM 中特訂區域移動至 DDRAM 的另一塊區域，此功能操作可以加速不同區塊資料複製並且搭配光柵運算，節省大量 MCU 執行時間及負載。

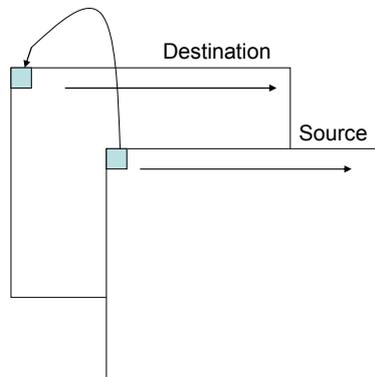


圖 7-38 : BTE 正向移動搭配光柵運算

BTE 移動來源/目的可以是一個方形的區域或是一個線性區域。此功能可用於暫時儲存 DDRAM 中部份的顯示區域資料到另一個非顯示區，以供後續利用，或是快速複製一個非顯示區資料到顯示區域。以下為 BTE 正向移動搭配光柵運算功能執行時建議的步驟，請參考以下暫存器設定。

- | | |
|------------------------------------|---------------------------------|
| 1. 設定來源圖層和位址 | → REG[54h], [55h], [56h], [57h] |
| 2. 設定目的圖層和位址 | → REG[58h], [59h], [5Ah], [5Bh] |
| 3. 設定 BTE 寬度和高度 | → REG[5Ch], [5Dh], [5Eh], [5Fh] |
| 4. 設定 BTE 操作碼和光柵運算碼 | → REG[51h] Bit[3:0] = 2h |
| 5. 啟動 BTE 功能 | → REG[50h] Bit7 = 1 |
| 6. 檢查狀態暫存器 STSR 的 Bit6，判斷 BTE 是否完成 | → check 2D final |

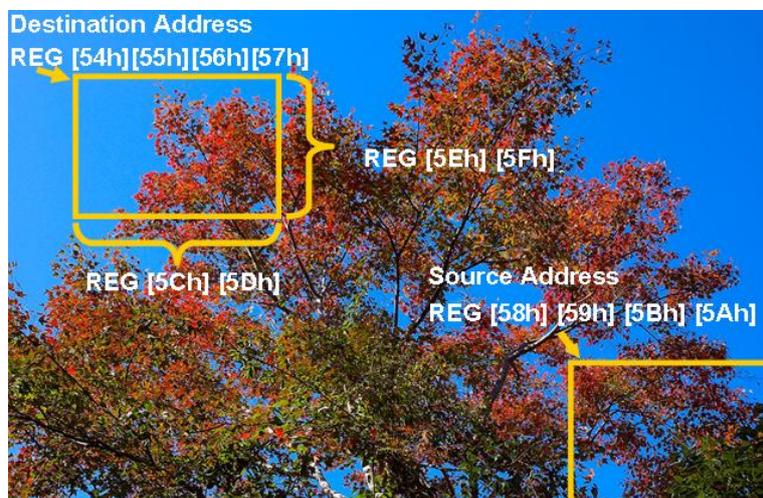


圖 7-39 : BTE 功能運作前畫面

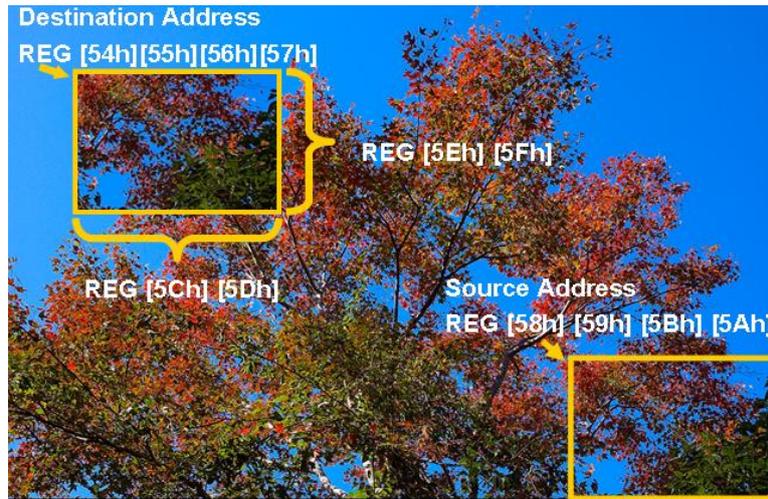


圖 7-40 : BTE 功能運作後畫面

7-6-4-4 BTE反向移動搭配光柵運算功能

BTE 反向移動搭配光柵運算功能與正向移動功能幾乎是相同的功能，唯一的差別為移動的方向。此功能先執行來源端至目的端的最後一筆資料的移動，再以反向的方式朝來源/目的端區域的第一筆資料，逐筆進行 BTE 的移動動作。特別要注意的是，在來源端與目的端有重疊的情況下，正向移動與反向移動功能會得到不同的結果。

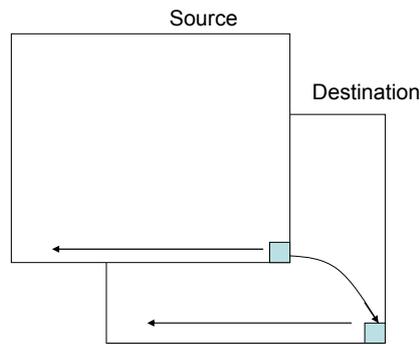


圖 7-41 : BTE 反向移動功能搭配光柵運算

BTE 移動功能可執行將 DDRAM 中特定區域移動至 DDRAM 另一塊不同區域的功能，此功能操作可以加速不同區塊資料複製並且搭配光柵運算，節省大量 MCU 執行時間及負載。

以下為 BTE 反向移動搭配光柵運算功能執行時的建議步驟，請參考以下暫存器設定。

- | | |
|------------------------------------|---------------------------------|
| 1. 設定來源圖層和位置 | → REG[54h], [55h], [56h], [57h] |
| 2. 設定目的圖層和位置 | → REG[58h], [59h], [5Ah], [5Bh] |
| 3. 設定 BTE 寬度和高度 | → REG[5Ch], [5Dh], [5Eh], [5Fh] |
| 4. 設定 BTE 操作碼和光柵運算碼 | → REG[51h] Bit[3:0] = 3h |
| 5. 開啟 BTE 功能 | → REG[50h] Bit[7] = 1 |
| 6. 檢查狀態暫存器 (STSR) Bit6，判斷 BTE 是否完成 | → check 2D final |

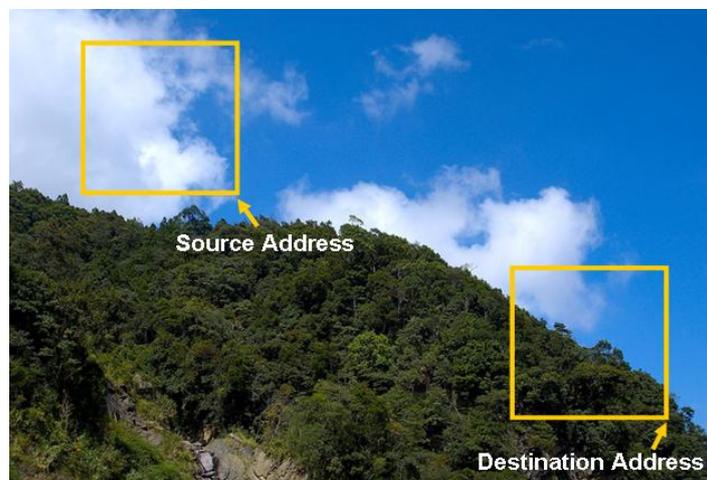


圖 7-42 : BTE 功能運作前畫面



圖 7-43 : BTE 功能運作後畫面

7-6-4-5 BTE 通透性寫入功能

BTE 通透性寫入功能可以加增加 MCU 端系統記憶體至 DDRAM 的傳送速度，一旦 BTE 通透性寫入功能開始，BTE 引擎會持續動作直到所有的像素都被寫入為止。

BTE 通透性寫入功能可用來更新一個 DDRAM 的特定區域，而由 MCU 提供資料來源，不同於 BTE 寫入功能，BTE 通透性寫入功會忽略某些特定顏色的操作，此特定通透色可由使用者設定，在 RA8875 中，此特定通透色設定於暫存器中的「BTE 前景色」中，當讀到來源顏色為通透色時，便不執行寫入的功能。此功能在處理將一張圖片的部分圖形複製到 DDRAM 時很有幫助。不需被複製的地方，在來源圖片中便以「通透色」來處理，在 BTE 通透性寫入功能執行時，便不會進入寫入功能，利用此功能可以很快的在任意背景圖上，寫入一個前景圖案，例如來源圖案為一個藍色的背景搭配紅色的圓形圖案，藉著設定藍色為「通透色」並且執行 BTE 通透性寫入功能，就相當於將一個紅色的圖形圖案貼到目的地位置功能。BTE 通透性寫入功能在來源和目的資料設定皆支援線性和區塊定址模式。

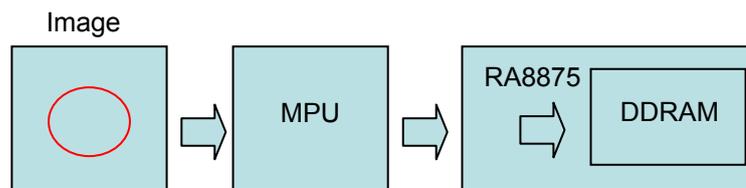


圖 7-44 : BTE 通透性寫入功能

以下為 BTE 通透性寫入功能執行時的建議步驟，請參考以下暫存器設定。

1. 設定目的位置 → REG[58h], [59h], [5Ah], [5Bh]
2. 設定 BTE 寬度暫存器 → REG[5Ch], [5Dh]
3. 設定 BTE 高度暫存器 → REG[5Eh], [5Fh]
4. 設定通透色 (BTE 前景色) → REG[63h], [64h], [65h]
5. 設定 BTE 操作碼和光柵運算碼 → REG[51h] = C4h
6. 開啟 BTE 功能 → REG[50h] Bit7 = 1
7. 致能 REG[02h]
8. 寫入圖像資料
9. 檢查狀態暫存器 (STSR) Bit7，判斷資料是否完成
10. 繼續執行步驟 8、9 直到影像資料等於區塊影像資料，或檢查暫存器 STSR Bit6



圖 7-45 : BTE 功能執行前畫面



圖 7-46 : BTE 通透性寫入功能執行後畫面

以下為執行時的建議步驟，請參考以下暫存器設定。

1. 設定 INT# → REG[F0h]
2. 設定目的位置 → REG[58h], [59h], [5Ah], [5Bh]
3. 設定 BTE 寬度暫存器 → REG[5Ch], [5Dh]
4. 設定 BTE 高度暫存器 → REG[5Eh], [5Fh]
5. 設定暫存器目的 = 來源 → REG[51h] = C4h
6. 開啟 BTE 功能 → REG[50h] Bit7 = 1
7. 等待中斷產生
8. 清除 INT# BTE 讀取/寫入狀態 → REG[F1h] Bit0 = 1
9. 致能 REG [02h]
10. 寫入圖像資料
11. 等待中斷產生
12. 清除 INT# BTE 讀取/寫入狀態 → REG[F1h] Bit0 = 1
13. 繼續執行步驟 9 ~ 12 直到影像資料等於區塊影像資料，或檢查暫存器 STSR Bit6

7-6-4-6 BTE正向通透性移動功能

BTE 正向通透性移動功能執行將 DDRAM 的某一區塊至另一區塊的移動功能，但忽略通透色的動作。與 BTE 通透性寫入功能相同的，它允許使用者設定某一個顏色為通透色，並且在遇到通透色時，不執行移動的功能。「通透性寫入」與「通透性移動」不同之處在於操作的來源設定，「通透性寫入」的來源資料來自系統記憶體或是 MCU，而「通透性移動」的來源則為 DDRAM。因為來源資料來自 DDRAM，BTE 動作的方向必須被定義，否則會造成執行結果的不確定，在「通透性移動」功能上，RA8875 僅支援正向的動作。

根據使用者設定，BTE 通透性移動功能的來源可以指定為線性模式或是區塊模式。值得注意的是某些來源與目的區域重疊的情況，來源區域的資料可能會在 BTE 執行的過程中被覆蓋。

以下為 BTE 正向通透性移動功能執行時建議的步驟，請參考以下暫存器設定。

- | | |
|------------------------------------|---------------------------------|
| 1. 設定來源圖層與位置 | → REG[54h], [55h], [56h], [57h] |
| 2. 設定目的圖層與位置 | → REG[58h], [59h], [5Ah], [5Bh] |
| 3. 設定 BTE 寬度與高度 | → REG[5Ch], [5Dh], [5Eh], [5Fh] |
| 4. 設定通透色，也就是 BTE 前景色 | → REG[63h], [64h], [65h] |
| 5. 設定 BTE 操作碼光柵運算功能 | → REG[51h] Bit[3:0] = 5h |
| 6. 開啟 BTE 功能 | → REG[50h] Bit7 = 1 |
| 7. 檢查狀態暫存器 (STSR) Bit6，確認 BTE 是否完成 | → check 2D final |

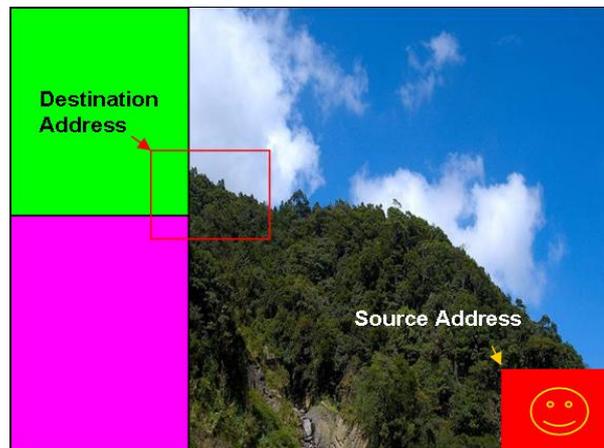


圖 7-47 : BTE 功能執行前畫面

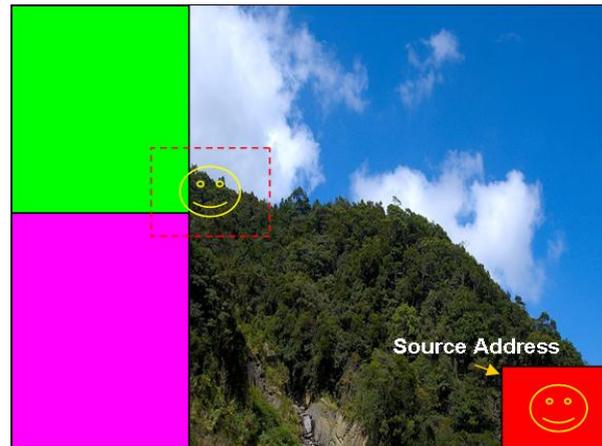


圖 7-48 : BTE 功能執行後畫面

7-6-4-7 圖形樣板填入搭配光柵運算功能

圖形樣板填入搭配光柵運算功能可設定一個在 DDRAM 中的特定方形記憶區塊，並填入重覆的特定圖形樣板。圖形樣板是 8x8/16x16 的像素圖形，存放在 DDRAM 中的非顯示區的特定位置，圖形樣板並且可以配合 16 種光柵運算和目的資料做邏輯運算。此操作可以用來加速某些需要在特定區域重覆貼入某一種圖形，像是背景圖案等應用

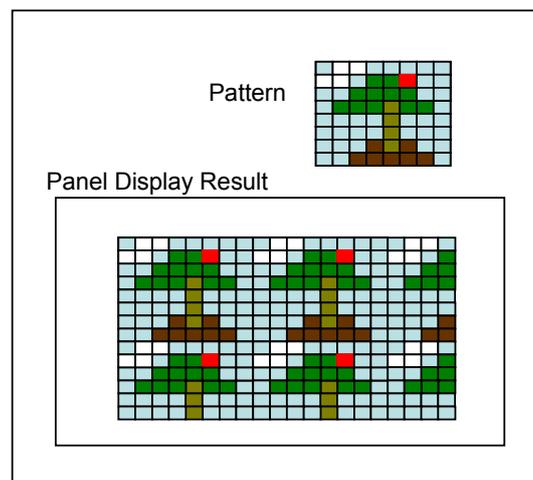


圖 7-49 : 圖形樣板填入搭配光柵運算功能

以下為圖形樣板填入搭配光柵運算功能執行時建議的步驟，請參考以下暫存器設定。

- | | |
|----------------------------------|---------------------------------|
| 1. 設定目的圖層和位置 | → REG[58h], [59h], [5Ah], [5Bh] |
| 2. 設定 BTE 寬度和高度 | → REG[5Ch], [5Dh], [5Eh], [5Fh] |
| 3. 設定 BTE 操作碼和光柵運算碼 | → REG[51h] Bit[3:0] = 06h |
| 4. 開啟 BTE 功能 | → REG[50h] Bit7 = 1 |
| 5. 檢查狀態暫存器 STSR Bit6，確認 BTE 是否完成 | → check 2D final |



圖 7-50 : BTE 功能執行前畫面

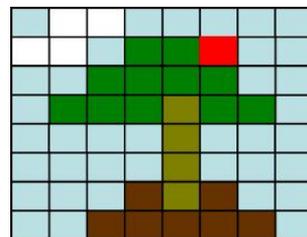


圖 7-51 : 樣版範例

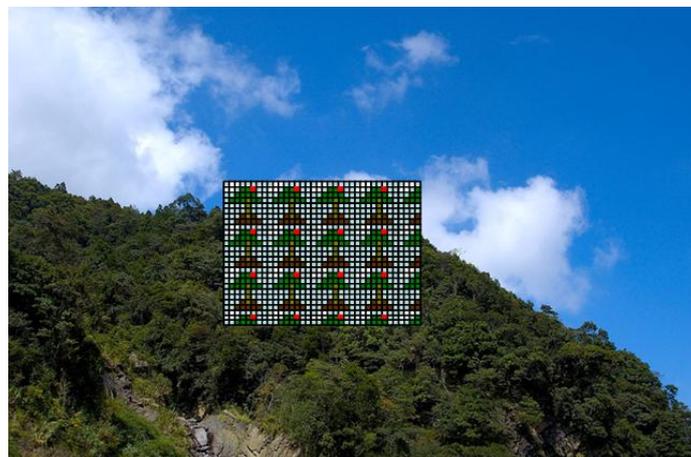


圖 7-52 : BTE 圖形樣版填入功能執行後畫面

7-6-4-8 通透性圖形樣板填入功能

通透性圖形樣板填入功能可設定在一個 DDRAM 中的特定方形記憶體，並填入重覆的特並圖形樣板。此功能與「圖形樣板填入搭配光柵運算」功能有相同的功能並且加入通透性的功能。也就是對於特定的「通透色」，此 BTE 功能會予以忽略。圖形樣板是一個 8*8 像素大小的圖形，存放在非顯示區的 DDRAM 中，在 BTE 啟動前，必須要先將圖形樣板填好。值得注意的是，對通透圖形樣板而言，通透色僅判別高位元之 256 色部分。也就是說 REG[63h] 的 BIT[4:2]、REG[64h] 的 BIT [5:3] 與 REG[65h] 的 BIT [4:3] 為有效。詳細設定請參考相關暫存器之說明。

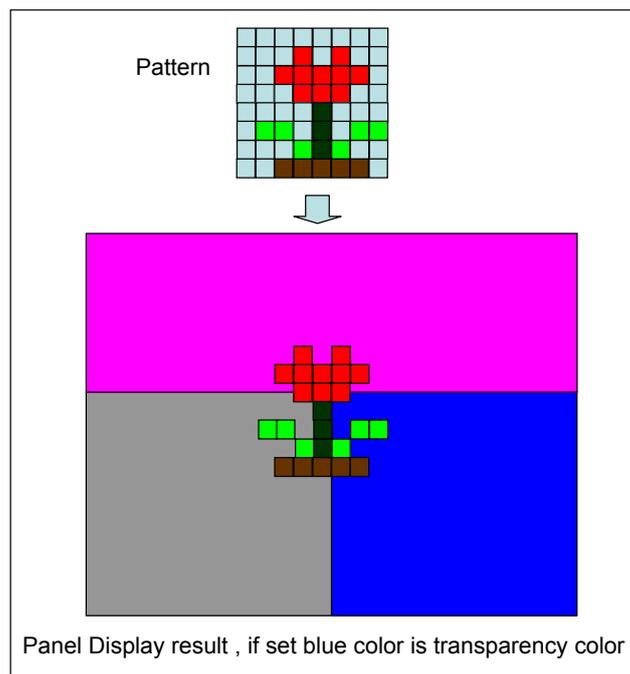


圖 7-53：通透性圖形樣板填入功能示意圖

以下為通透性圖形樣板填入功能執行時的建議步驟，請參考以下暫存器設定。

- | | |
|----------------------------------|---------------------------------|
| 1. 設定目的圖層和位置 | → REG[58h], [59h], [5Ah], [5Bh] |
| 2. 設定 BTE 寬度和高度 | → REG[5Ch], [5Dh], [5Eh], [5Fh] |
| 3. 設定通透色-BTE 前景色 | → REG[63h], [64h], [65h] |
| 4. 設定 BTE 操作碼和光柵運算碼 | → REG[51h] Bit[3:0] = 07h |
| 5. 開啟 BTE 功能 | → REG[50h] Bit7 = 1 |
| 6. 檢查狀態暫存器 STSR Bit6，確認 BTE 是否完成 | → check 2D final |



圖 7-54 : BTE 功能執行前畫面

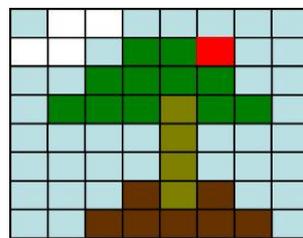


圖 7-55 : 樣板圖例

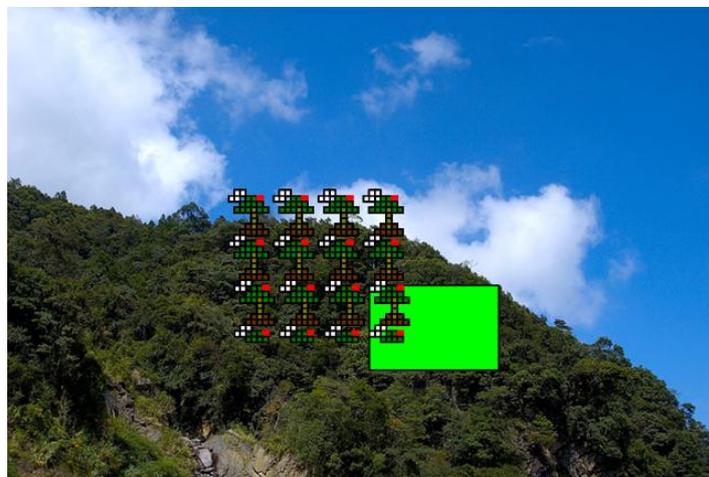


圖 7-56 : 通透性圖形樣板填入功能執行後畫面

7-6-4-9 顏色擴充功能

顏色擴充是一個很有用的功能，用來處理 MCU 的單色圖形資料轉換為彩色圖形資料，並寫入 DDRAM 中。此功能的來源資料為 MCU 提供的單色圖形資料 (Monochromes Bitmap)。而每一個位元根據內容被轉換為 BTE 前景色或背景色。若來源位置值為"1" 則會被轉換為 BTE 前景色，若為"0" 則會轉換為 BTE 背景色。此功能可以大大降低將單色系統資料轉換為彩色系統資料的成本。顏色擴充功能會根據 MCU 的資料匯流排寬度，持續讀入 16 位元或 8 位元的資料做轉換，並且可以位元為單位，設定每一行的第一筆單色圖形資料的起始轉換位元，並且在每一行的最後一筆資料讀入後，超過範圍的位元也會被忽略而不寫入，而下一行則從下一筆資料開始執行同樣的操作。這樣以位元為單位的運算大大增加此功能的彈性。另外，每一筆資料的處理方向是從最高位元 (MSB) 至最低位元 (LSB)。

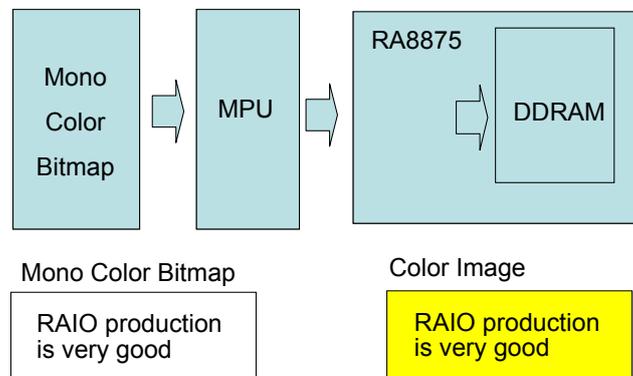


圖 7-57 : 顏色擴充功能的資料區塊圖

以下為顏色擴充功能執行時建議步驟，請參考以下暫存器設定。

1. 設定目的位置 → REG[58h], [59h], [5Ah], [5Bh]
2. 設定 BTE 寬度暫存器 → REG[5Ch], [5Dh]
3. 設定 BTE 高度暫存器 → REG[5Eh], [5Fh]
4. 設定 BTE 背景色 – 若輸入位元為 0，則轉換為此顏色 → REG[60h], [61h], [62h]
5. 設定 BTE 前景色 – 若輸入位元為 1，則轉換為此顏色 → REG[63h], [64h], [65h]
6. 設定 BTE 操作碼和光柵運算碼 → REG[51h] Bit[3:0] = 08h
7. 開啟 BTE 功能 → REG[50h] Bit7 = 1
8. 致能 REG[02h]
9. 檢查狀態暫存器 STSR Bit7，確認資料是否寫入
10. 寫入單色圖像資料
11. 繼續執行步驟 8、9、10 直到圖像更新完畢，或檢查暫存器 STSR Bit6 確認 BTE 執行完成

以下步驟採用中斷方式來進行 BTE 狀態確認，使用此方式必須先行將中斷信號 INT# 與 MCU 進行連接。

- | | |
|---|---------------------------------|
| 1. 設定中斷控制暫存器 | → REG[F0h] |
| 2. 設定目的位置 | → REG[58h], [59h], [5Ah], [5Bh] |
| 3. 設定 BTE 寬度暫存器 | → REG[5Ch], [5Dh] |
| 4. 設定 BTE 高度暫存器 | → REG[5Eh], [5Fh] |
| 5. 設定 BTE 背景色 – 若輸入位元為 0，則轉換為此顏色 | → REG[60h], [61h], [62h] |
| 6. 設定 BTE 前景色 – 若輸入位元為 1，則轉換為此顏色 | → REG[63h], [64h], [65h] |
| 7. 設定 BTE 操作碼和光柵運算碼 | → REG[51h] Bit[3:0] = 08h |
| 8. 開啟 BTE 功能 | → REG[50h] Bit7 = 1 |
| 9. 等待中斷信號產生 | |
| 10. 檢查得知 BTE 中斷，並清除中斷狀態暫存器 | → REG[F1h] Bit0 = 1 |
| 11. 致能 REG[02h] | |
| 12. 寫入單色圖像資料 | |
| 13. 等待中斷信號產生。 | |
| 14. 檢查得知 BTE 中斷，並清除中斷狀態暫存器。 | → REG[F1h] Bit0 = 1 |
| 15. 繼續執行步驟 11~14 直到圖像資料被更新完畢，或檢查狀態暫存器 STSR Bit6 確認 BTE 執行完成 | |



圖 7-58 : BTE 功能執行前畫面

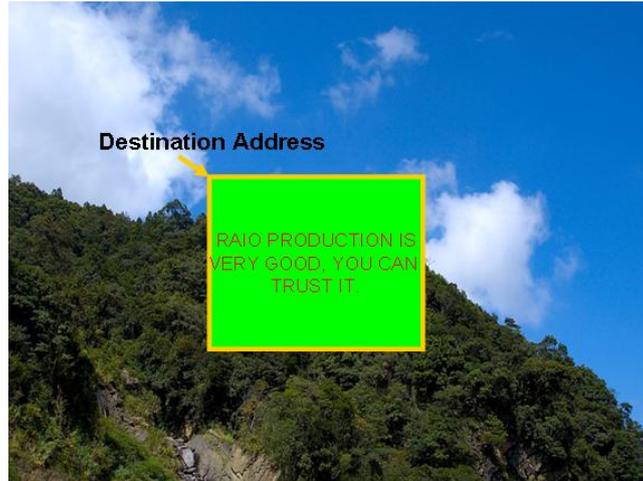


圖 7-59 : BTE 功能執行後畫面

註:

1. 每列需送出的資料筆數 = $((\text{BTE 寬度} - (\text{MCU 介面資料寬度} - (\text{起始位元數} + 1))) / \text{MCU 介面資料寬度}) + ((\text{起始位元數} + 1) \% (\text{MCU 介面資料寬度}))$
2. 所有需傳送的資料筆數 = (每列送出的資料筆數) x BTE 高度設定

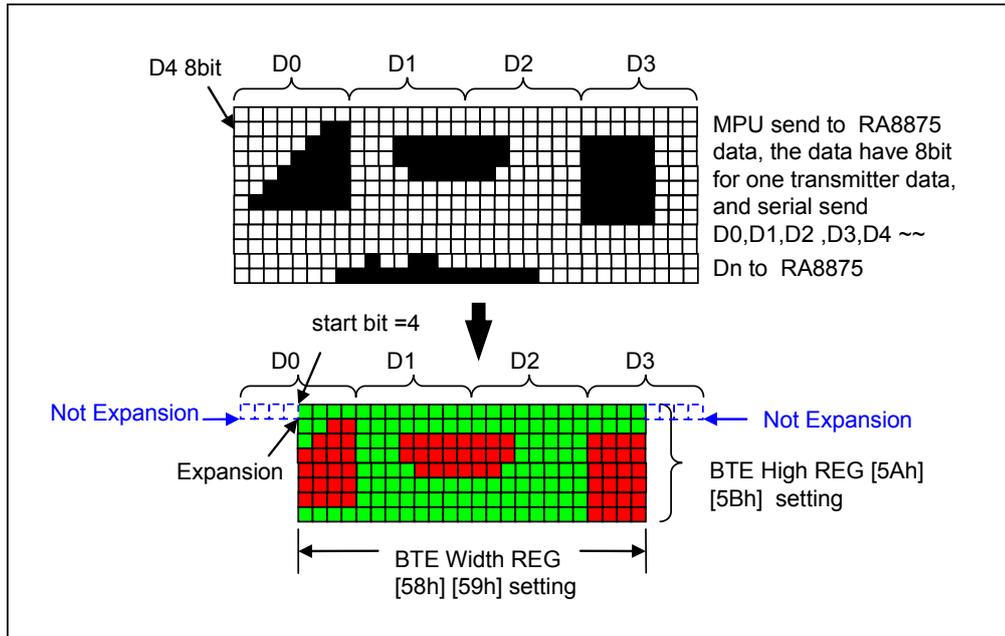


圖 7-60 : 顏色擴充功能資料示意圖

7-6-4-10 通透性顏色擴充功能

此 BTE 功能與顏色擴充功能幾乎是相同，除了加入通透性的功能。也就是對於特定的「通透資料位元值」，此 BTE 功能會予以忽略。在此功能，通透資料位元值為「0」，也就是所有輸入值為「1」的位元將會被轉換為 BTE 的前景色並且寫入目的位置，所有的輸入值為「0」的位元將不被轉換，而保持原來的目的資料顏色值。

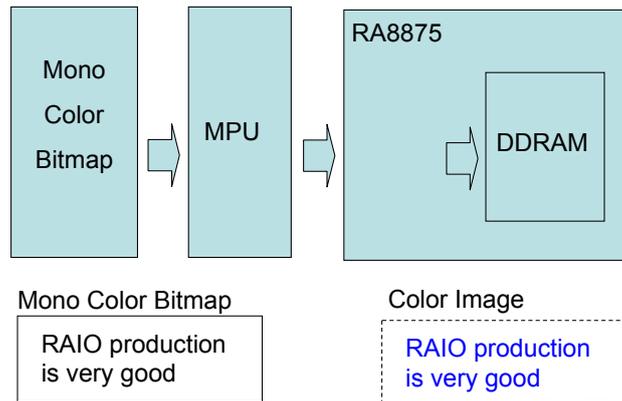


圖 7-61：通透性顏色擴充功能資料區塊圖

以下為通透性顏色擴充功能執行時建議的步驟，請參考以下暫存器設定。

1. 設定目的位置 → REG[58h], [59h], [5Ah], [5Bh]
2. 設定 BTE 寬度暫存器 → REG[5Ch], [5Dh]
3. 設定 BTE 高度暫存器 → REG[5Eh], [5Fh]
4. 設定 BTE 前景色，若輸入位元為 1，則轉換為此顏色 → REG[63h], [64h], [65h]
5. 設定 BTE 操作碼和光柵運算碼 → REG[51h] Bit[3:0] = 09h
6. 開啟 BTE 功能 → REG[50h] Bit7 = 1
7. 致能 REG[02h]
8. 檢查狀態暫存器 STSR Bit7，確認資料是否寫入
9. 寫入單色圖像資料
10. 繼續執行步驟 8、9，直到圖像資料被更新完畢，或檢查狀態暫存器 STSR Bit6 確認 BTE 執行完成

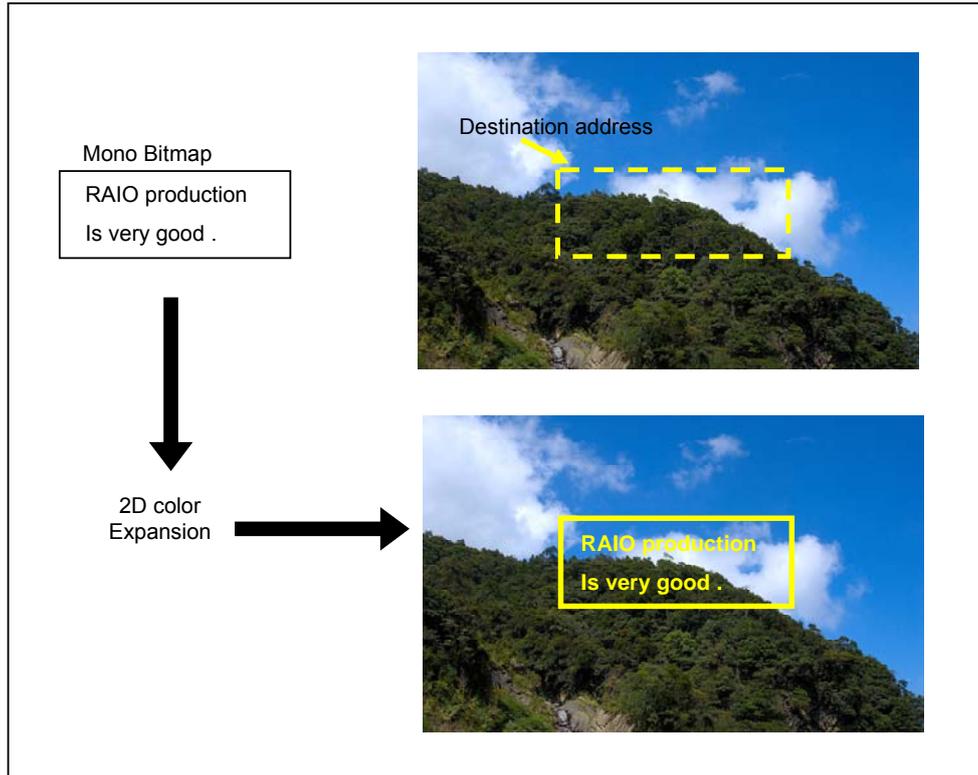


圖 7-62 : 通透性顏色擴充功能範例圖

以下步驟採用中斷方式來進行 BTE 狀態確認，使用此方式必須先行將中斷信號 INT# 與 MCU 進行連接。

1. 設定中斷控制暫存器 → REG[F0h]
2. 設定目的位置 → REG[58h], [59h], [5Ah], [5Bh]
3. 設定 BTE 寬度暫存器 → REG[5Ch], [5Dh]
4. 設定 BTE 高度暫存器 → REG[5Eh], [5Fh]
5. 設定 BTE 操作碼和光柵運算碼 → REG[51h] Bit[3:0] = 09h
6. 開啟 BTE 功能 → REG[50h] Bit7 = 1
7. 等待中斷信號產生
8. 檢查得知 BTE 中斷，並清除中斷狀態暫存器 → REG[F1h] Bit0 = 1
9. 致能 REG[02h]
10. 寫入單色圖像資料
11. 等待中斷信號產生
12. 檢查得知 BTE 中斷，並清除中斷狀態暫存器 → REG[F1h] Bit0 = 1
13. 繼續執行步驟 9 ~ 12 直到圖像資料被更新完畢，或檢查狀態暫存器 STSR Bit6 確認 BTE 執行完成

7-6-4-11 BTE移動搭配顏色擴充功能

BTE 移動搭配顏色擴充功能從 DDRAM 中的來源位置取得單色的圖像資料，並且將其轉換為彩色資料並移動至目的位置。所有來源會被視為是以位元為單位的單色資料，所有值為”1”的位元會被轉換為 BTE 前景色，而值為”0”的位元則會轉換為 BTE 背景色。

BTE 移動搭配顏色擴充功能可用來加速單色圖像轉換為彩色圖像的應用。在非顯示區的一個單色圖案所占的空間非常小，藉由硬體加速的優點，可以讓系統的負擔大大降低。也可用於文字為主的圖案應用。

此功能可從一個區塊移動資料到另外一塊，來源/目的資料皆可設定為線性或區塊模式，值得注意的是，當來源/目的定義為線性模式時，資料便視為每一列的資料都是連續的且相鄰。圖像的寬度則由暫存器中的 BTE 寬度來設定。

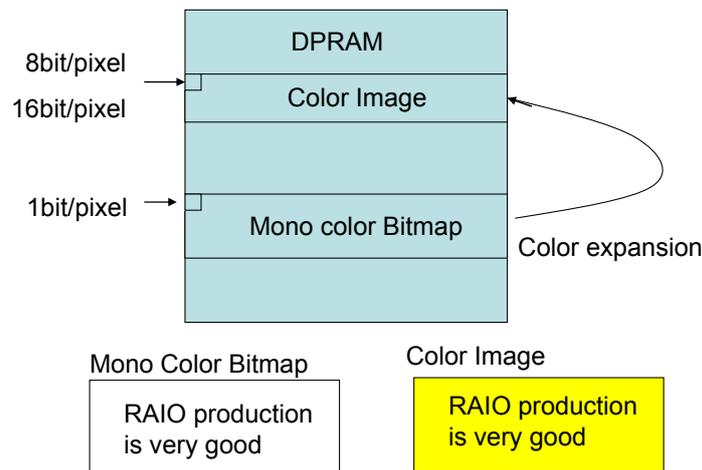


圖 7-63 : BTE 移動搭配顏色擴充功能資料轉換示意圖

以下為 BTE 移動搭配顏色擴充功能執行時建議的步驟，請參考以下暫存器設定。

1. 設定來源圖層和位置 → REG[54h], [55h], [56h], [57h]
2. 設定目的圖層和位置 → REG[58h], [59h], [5Ah], [5Bh]
3. 設定 BTE 寬度和高度 → REG[5Ch], [5Dh], [5Eh], [5Fh]
4. 設定 BTE 背景色，若輸入位元為 0，則轉換為此顏色 → REG[60h], [61h], [62h]
5. 設定 BTE 前景色，若輸入位元為 1，則轉換為此顏色 → REG[63h], [64h], [65h]
6. 設定 BTE 操作碼和光柵運算碼 → REG[51h] Bit[3:0] = 0Ah
7. 開啟 BTE 功能 → REG[50h] Bit7 = 1
8. 檢查狀態暫存器 STSR Bit6，確認 BTE 是否完成 → check 2D final



圖 7-64 : BTE 功能執行前畫面

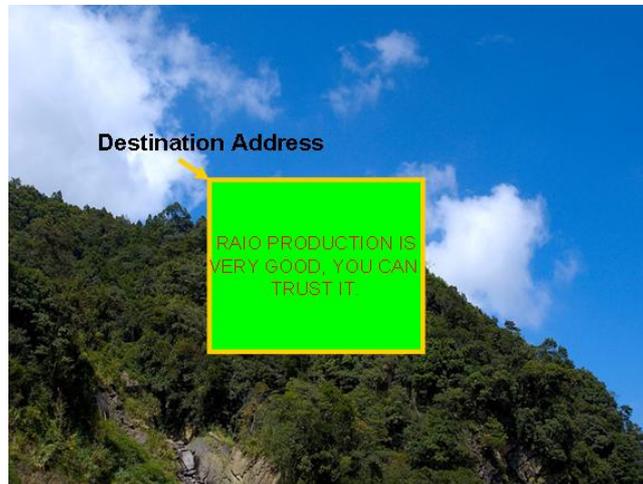


圖 7-65 : BTE 功能執行後畫面

7-6-4-12 通透性BTE移動功能搭配顏色擴充

「通透性顏色移動功能搭配顏色擴充」與「BTE 移動搭配 BTE 顏色擴充功能」幾乎是相同，除了加入通透性的功能。也就是背景色會被忽略。當所有輸入值為“1”的位元將會被轉換為 BTE 的前景色，而所有輸入值為“0”的位元將不被轉換。

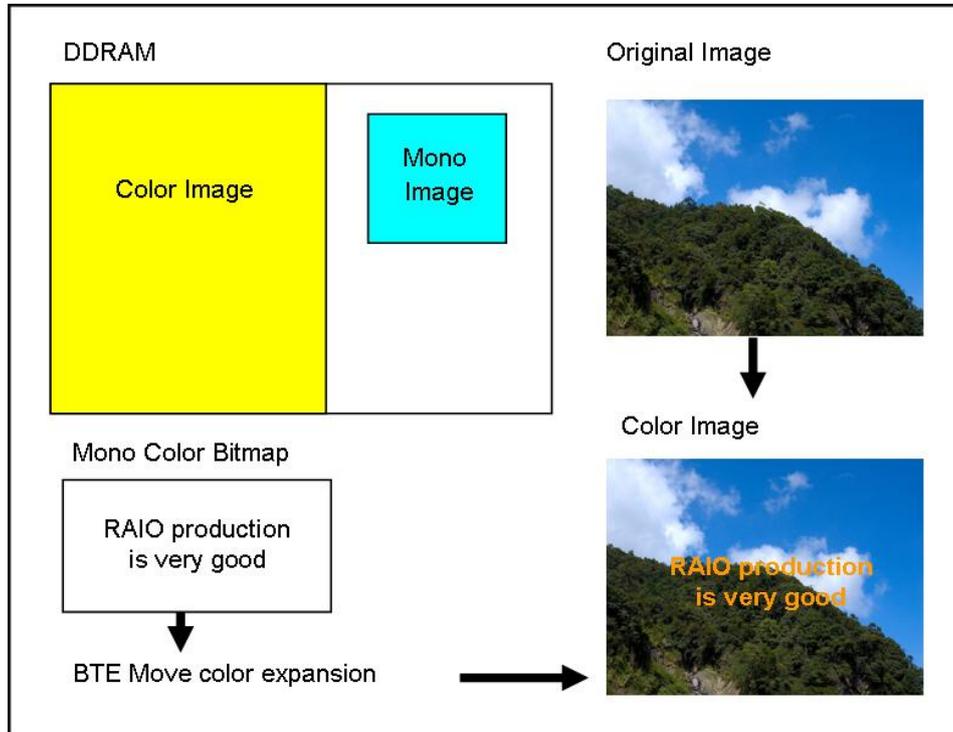


圖 7-66：通透性 BTE 移動功能搭配顏色擴充效果

建議步驟如下：

1. 設定來源圖層和位置 → REG[54h], [55h], [56h], [57h]
2. 設定目的圖層和位置 → REG[58h], [59h], [5Ah], [5Bh]
3. 設定 BTE 寬度和高度 → REG[5Ch], [5Dh], [5Eh], [5Fh]
4. 設定 BTE 前景色，若輸入位元為 1，則轉換為此顏色 → REG[63h], [64h], [65h]
5. 設定 BTE 操作碼和光柵運算碼 → REG[51h] Bit[3:0] = 0Bh
6. 開啟 BTE 功能 → REG[50h] Bit7 = 1
7. 檢查狀態暫存器 STSR REG Bit6，確認 BTE 是否完成

7-6-4-13 單色填滿功能

BTE 單色填滿功能可將 DDRAM 中選定的區塊填入一種單色。此功能使用於將選定特定區域畫面清除或是填入給定某種前景色，R8875 填入的單色設定為 BTE 前景色。



圖 7-67 : 單色填滿功能

以下為單色填滿功能執行時建議的步驟，請參考以下暫存器設定。

1. 設定目的圖層和位置 → REG[58h], [59h], [5Ah], [5Bh]
2. 設定 BTE 寬度和高度 → REG[5Ch], [5Dh], [5Eh], [5Fh]
3. 設定 BTE 操作碼和光柵運算碼 → REG[51h] Bit[3:0] = 0Ch
4. 設定前景色 → REG[63h], [64h], [65h]
5. 開啟 BTE 功能 → REG[50h] Bit7 = 1
6. 檢查狀態暫存器 STSR Bit6，確認 BTE 是否完成

7-7 圖層混合功能

RA8875 提供兩種圖層顯示功能，當選擇雙圖層 DPCR(REG[20h] Bit7=1) 時，可使用 LTPR0(REG[52h])、LTPR1(REG[53h]) 和 BGTR(REG[67h] ~ REG[69h]) 來產生圖層 1 和圖層 2 不同的組合效果。相關的功能設定請參考表 7-16。

表 7-16 : LTPR0、LTPR1 和 BGTR 功能

Reg. NO.	Abbreviation	Description
Layer Transparency Register 0		
52h	LTPR0	B[5] Floating Windows Display Related With BGTR
		B[2:0] Layer1/2 Display Mode
		000b: Only Layer 1 is visible
		001b: Only Layer 2 is visible
		011b: Transparent mode
		010b: Lighten-overlay mode
		100b: Boolean OR
		101b: Boolean AND
		110b: Floating Windows
		111b: Reserved
Layer Transparency Register 1		
53h	LTPR1	B[7:4] Layer Transparency Setting for Layer 2
		0000b: Total display
		0001b: 7/8 display
		0010b: 3/4 display
		0011b: 5/8 display
		0100b: 1/2 display
		0101b: 3/8 display
		0110b: 1/4 display
		0111b: 1/8 display
		1000b: Display disable
		B[3:0] Layer Transparency Setting for Layer 1
		0000b: Total display
		0001b: 7/8 display
		0010b: 3/4 display
		0011b: 5/8 display
		0100b: 1/2 display
		0101b: 3/8 display
		0110b: 1/4 display
		0111b: 1/8 display
		1000b: Display disable
Background Color Register for Transparent		
67h	BGTR0	B[4:0] Background Color for Transparent Red
68h	BGTR1	B[5:0] Background Color for Transparent Green
69h	BGTR2	B[4:0] Background Color for Transparent Blue

7-7-1 顯示圖層一

若暫存器LTPR0 B[2:0] 設定為 000b，畫面只會顯示圖層一，請參考圖 7-68 範例。這個功能也可結合LTPR1[3:0] 和BGTR來顯示相似的效果，請參照下面範例，如圖 7-69。

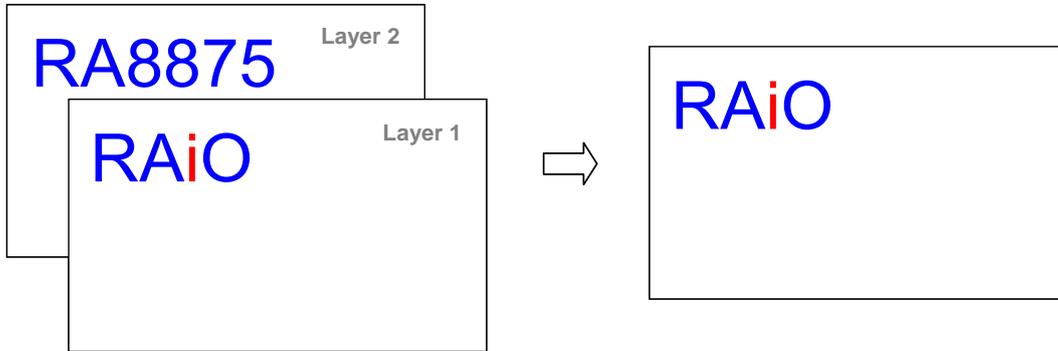


圖 7-68 : 只顯示圖層一

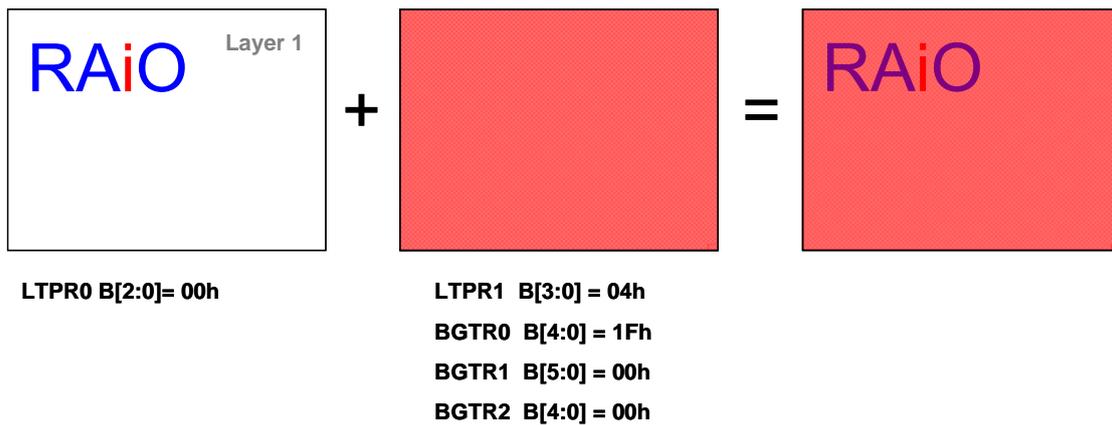


圖 7-69 : 圖層一與通透模式的顯示效果

7-7-2 顯示圖層二

若暫存器LTPR0 B[2:0] 設定為 001b，畫面只會顯示圖層二，請參照 圖 7-70 範例。這個功能也可結合暫存器LTPR1[7:4] 和 BGTR來顯示相似的效果，請參照下面範例，如 圖 7-71。

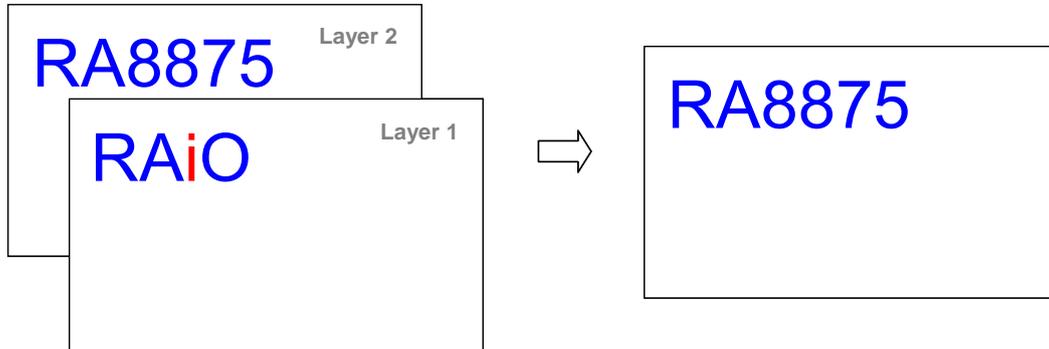


圖 7-70：只顯示圖層二

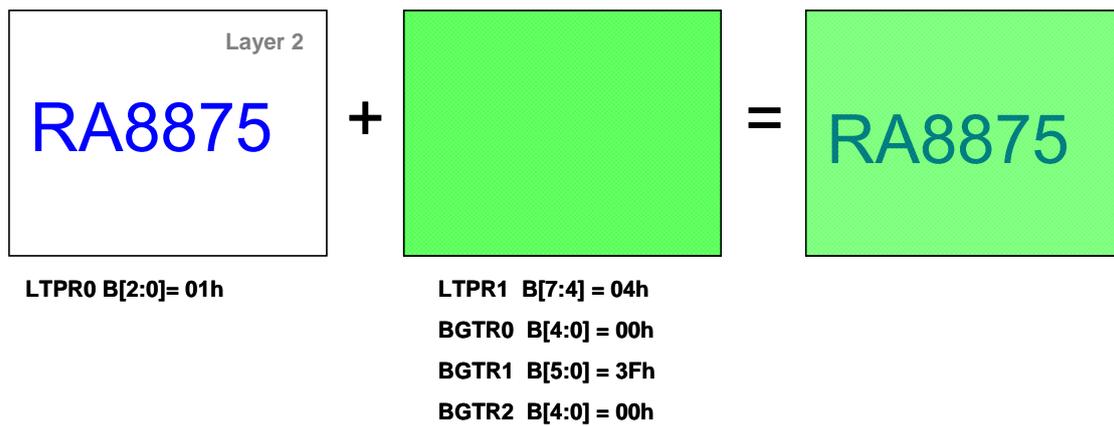


圖 7-71：圖層二與通透模式的顯示效果

7-7-3 漸入漸出模式

通透模式使得圖層一的背景色以透明的方式顯示，漸入漸出模式可以進一步變換圖片顯示效果，以下公式描述漸入漸出 (Lighten-Overlay) 的使用技術。

$$[r,g,b]_{\text{Lighten-Overlay}} = \chi [r,g,b]_{\text{Layer 1}} + (1 - \chi) [r,g,b]_{\text{Layer 2}}$$

其中 $[r,g,b]$ 是像素的資料，而 χ 則是顯示的權重係數，此係數大小是依據 LTPR1[3:0] 設定。換句話說，如果 LTPR1[3:0] 設定為 0100b，顯示的權重係數 χ 便等於 1/2，公式如下：

$$[r,g,b]_{\text{Lighten-Overlay}} = 1/2 [r,g,b]_{\text{Layer 1}} + 1/2 [r,g,b]_{\text{Layer 2}}$$

其顯示效果請參照 圖 7-72。



圖 7-72：漸入漸出模式的效果

7-7-4 通透模式

通透模式可使得圖層一的背景色以透明的方式顯示，也就是在圖層一的背景色部份，圖層二的部份可以被顯示。這個功能使得兩個圖層可以重疊顯示，有前景和背景的效果。其中前景的部份放在圖層一，背景的部分則寫至圖層二。而需要透明的部份則寫入由暫存器BGTR所設定的背景色。其顯示效果請參考圖 7-73。



圖 7-73 : 圖層的通透模式顯示效果

7-7-5 布林運算OR

設定暫存器 REG[52h]，圖層一與圖層二可以進行”OR”的結合顯示，如下圖所示。

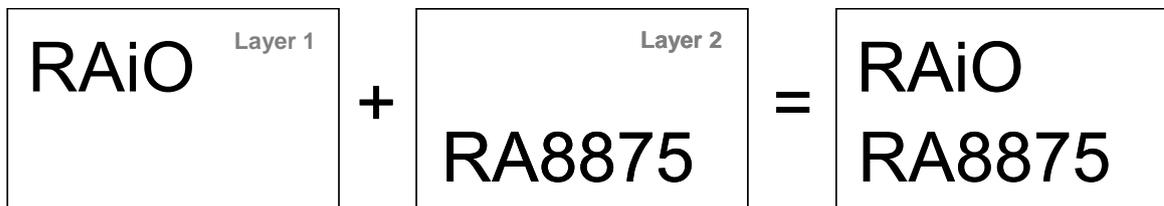


圖 7-74 : 圖層的布林運算 - OR

7-7-6 布林運算AND

設定暫存器 REG[52h]，圖層一與圖層二可以進行”AND”的結合顯示，如下圖所示。

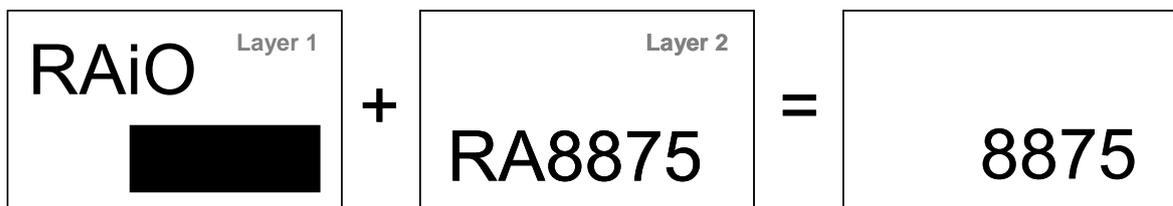


圖 7-75 : 圖層的布林運算 - AND

7-7-7 浮動視窗

浮動視窗模式提供圖中圖的顯示效果。我們可以使用浮動視窗來顯示圖層二的特定部分影像到圖層一的顯示視窗中，有關顯示效果請參考圖 7-76。當REG[52h] 的Bit[5] 設為”1” 的時候，浮動視窗也可以被設定為相關的BGTR，與通透模式類似，圖層二的BGTR是透明色，代表只會顯示圖層一，相關的顯示效果請參考圖 7-77。

使用方式：

1. 透過暫存器 FWSAXA0[D0h]、FWSAXA1[D1h]、FWSAYA0[D2h] 和 FWSAYA0[D3h] 設定浮動視窗起始位置。
2. 透過暫存器 FWW0[D4h]、FWW1[D5h]、FWH0[D6h] 和 FWH1[D7h] 設定浮動視窗寬度和高度。
3. 透過設定暫存器 FWDXA0[D8h]、FWDXA 1[D9h]、FWDYA 0[DAh] 和 FWDYA 0[DBh] 設定浮動視窗顯示位置
4. 若使用浮動視窗相關 BGTR 功能，記得開啟 REG[52h] 的 Bit 5，透過 BGTR0[67h]、BGTR1[68h] and BGTR2[69h] 設定 BGTR 顏色。

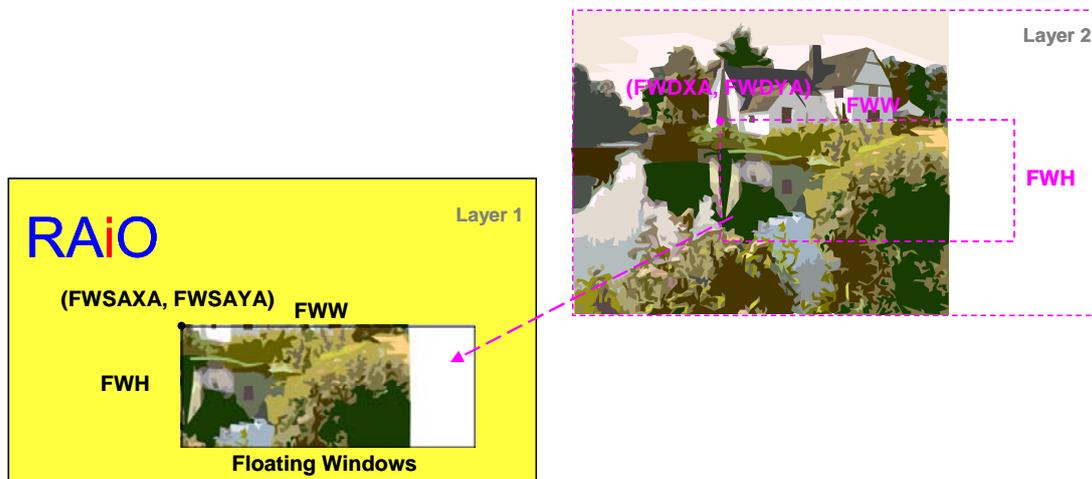


圖 7-76：浮動視窗的顯示效果

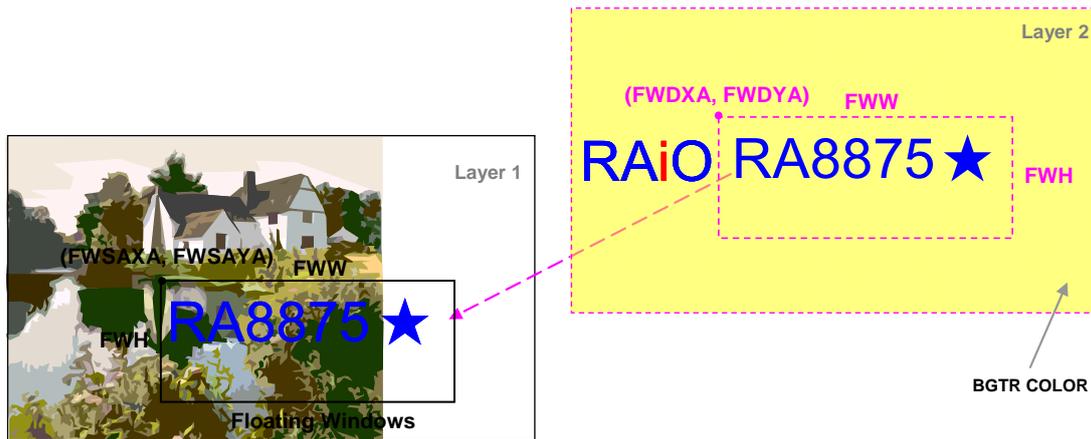


圖 7-77：與 BGTR 相關的浮動視窗顯示效果

7-8 觸控面板功能

RA8875 內建一組 10 位元 ADC 和控制電路，以連接 4-wire 電阻式的觸控面板。其工作原理及應用電路請參考章節 6-5。而整個觸控面板控制器可分為「自動模式」與「手動模式」，每種模式又分成「硬體中斷模式」與「軟體輪詢模式」，下面將分別介紹動作流程。

RA8875 有四種狀態的觸控面板控制器，分別為「閒置狀態」、「觸控事件確認狀態」、「門鎖 X 軸 Data」、「門鎖 Y 軸 Data」。RA8875 提供「自動模式」與「手動模式」兩種操作模式。「自動模式」會自動確認調整觸控事件確認狀態；「手動模式」透過手動的操作，使用在不穩定的或特殊的應用上，使用者可以更有彈性自行安排。

當開啟「觸控事件」時，RA8875 提供兩種偵測方法，分別為「硬體中斷模式」與「軟體輪詢模式」，請參考表 7-17 的觸控功能的模式。

表 7-17：觸控功能的模式

Operation Mode	Event Detection	Description
Auto	Interrupt	When touch event happens, read the corresponding X, Y coordination.
	Polling	Polling the touch event, read the corresponding X, Y coordination.
Manual	Interrupt	Set the operation state to "Checking touch event" for checking the touch event, when touch event interrupt happens, set the state to "Latch X data" and "Latch Y data" for latching the corresponding X, Y coordination, then read the X, Y data and set operation state to "Idle state"
	Polling	Polling the touch event, and read the corresponding X, Y coordination. Set the operation state to "Checking touch event" for checking the touch event. Polling the touch event status before confirming the touch event, set the state to "Latch X data" and "Latch Y data" for latching the corresponding X, Y coordination, then read the X, Y data and set operation state to "Idle state"

7-8-1 觸控面板操作模式

7-8-1-1 自動模式

自動模式是一種最簡單的觸控面板功能的控制方式。只要開啟相關的暫存器，RA8875 會自動執行觸控面板偵測功能以及閃鎖觸控資料，請參考下列流程圖。

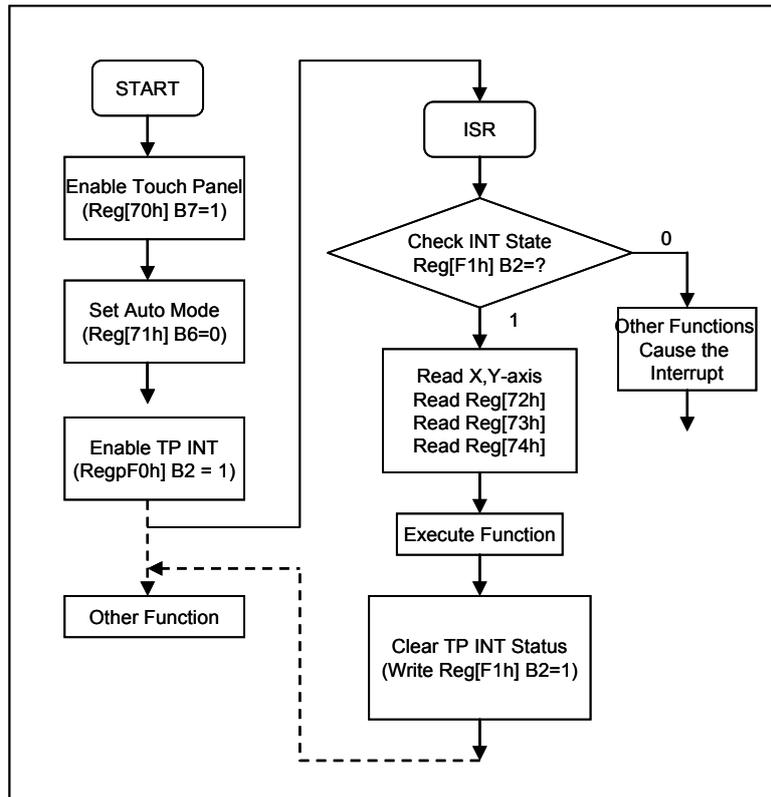


圖 7-78 : 觸控面板自動模式流程圖

表 7-18 : 自動模式相關的暫存器

Reg.	Bit_Nu m	Description	Reference
TPCR0	Bit7	Enable Touch Panel function	REG[70h]
TPCR1	Bit6	“Auto-Mode” = 0	REG[71h]
	Bit2	Set de-bounce enable for ADET(note)	
TPXH	Bit[7:0]	Touch Panel SEG data MSB byte	REG[72h]
TPYH	Bit[7:0]	Touch Panel COM data MSB byte	REG[73h]
TPXYL	Bit[3:2]	Touch Panel COM data LSB 2bit	REG[74h]
	Bit[1:0]	Touch Panel SEG data LSB 2bit	

註：建議在 ADET 自動模式中設定消除機械彈跳功能，否則干擾會造成觸控事件判斷錯誤。

7-8-1-2 手動模式

手動模式是指使用者自行以手動操作方式來完成「偵測觸控事件」、「門鎖 X 軸 Data」與「門鎖 Y 軸 Data」。整個過程由設定暫存器 TPCR1[1:0] 來完成。手動模式的優點是應用上更有彈性。可自行決定 X 軸與 Y 軸 Data 的消除機械彈跳與模式切換時間，降低自動模式在某些情況下錯誤發生的機率。

在手動模式下，使用者需要透過持續輪詢“狀態暫存器”來確定觸控事件的正確性。一般來說，當連續輪詢到足夠次數的狀態暫存器中的觸控事件時，我們便認為正確的觸控事件。手動模式允許更多彈性且在不同應用上更少發生失誤，但是會佔用到較多的 MCU 的資源。

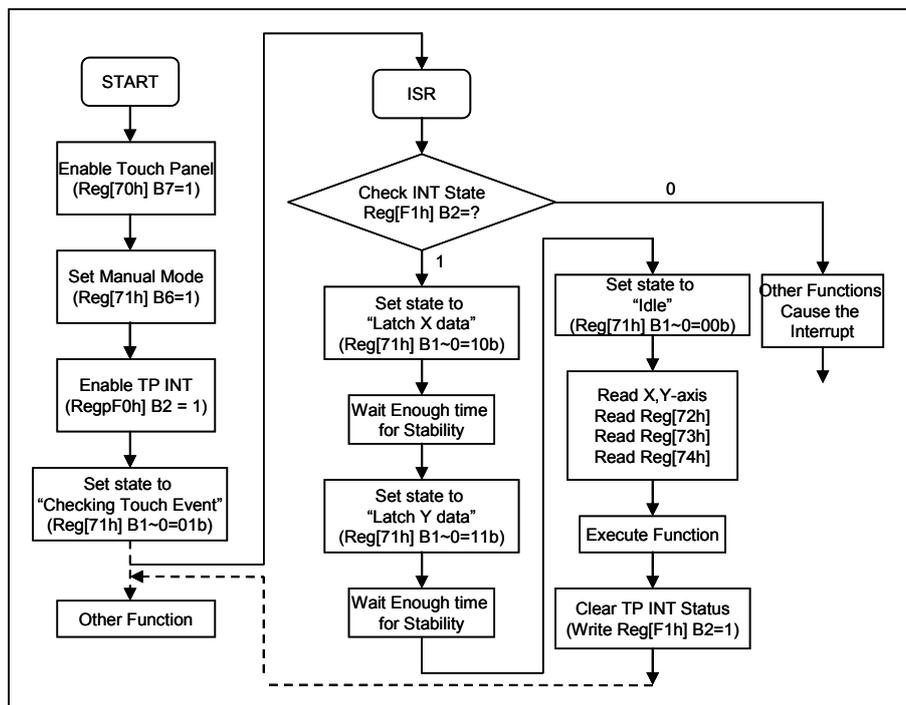


圖 7-79 : 觸控面板手動模式流程圖

表 7-19 : 手動模式相關的暫存器

Reg.	Bit_Num	Description	Reference
TPCR0	Bit7	Enable Touch Panel function	REG[70h]
TPCR1	Bit6	"Manual-Mode" = 1	REG[71h]
	Bit2	Set de-bounce function for ADET(note)	
	Bit[1:0]	Mode Selection for TP Manual Mode	
TPXH	Bit[7:0]	Touch Panel SEG data MSB byte	REG[72h]
TPYH	Bit[7:0]	Touch Panel COM data MSB byte	REG[73h]
TPXYL	Bit[3:2]	Touch Panel COM data LSB 2bit	REG[74h]
	Bit[1:0]	Touch Panel SEG data LSB 2bit	

註：若使用者在觸控事件上不用軟體消除機械彈跳，可以設定開啟消除機械彈跳。否則使用者可以透過軟體消除機械彈跳，然後設定關閉此功能。

7-8-2 觸控事件偵測模式

觸控事件依據系統組態可從中斷模式與輪詢模式中被偵測，相關的說明請參考接下來的章節。

7-8-2-1 外部中斷模式

在中斷模式下，RA8875 硬體中斷腳位(INT) 必須先正確地連接到 MCU 中斷輸入腳位，主要的過程如下：

1. 開啟觸控面板功能 (REG[70h] Bit7 = 1)
2. 設定觸控面板控制器操作模式為自動模式或手動模式 (REG[71h] Bit6)
3. 開啟觸控面板中斷功能 (REG[F0h] Bit6 = 1)
4. 當中斷發生時，MCU 的 IP 會跳到 ISR 的入口，並檢查是否為觸控事件所產生的中斷
5. 若是，則依據操作模式，進行門鎖 X、Y 軸的 Data
6. 執行觸控事件的要進行的程序
7. 清除中斷狀態位元 (set REG[F1h] Bit2 = 1)，並且離開 ISR

7-8-2-2 軟體輪詢模式

在輪詢模式下，需要連接不中斷的腳位。觸控事件的狀態可以由以下三種方式被讀取。

1. 來自狀態暫存器(STSR) 的 Bit 5，狀態暫存器直接來自硬體而且不做任何消除機械彈跳。建議透過軟體消除彈跳機制來確認觸控事件。
2. 來自 TPXYL(REG[74h]) Bit 7，此位元也是直接來自硬體，與 STSR Bit 5 相同。
3. 來自 INTC2(REG[F1h]) Bit2，與硬體中斷同樣的行為，透過軟體輪詢中斷事件的模式。

總結來說，程式設計師可以從 STSR 的 Bit5 或 INTC2 的 Bit2 來確認觸控事件的狀態，其中差異說明如下：

1. 狀態暫存器 STSR 的 Bit5 反應目前的觸控狀態，當觸控事件發生時，Bit5 被設定為 1。反之，當無觸控事件發生時，Bit5 會自動更新為 0。此方法通常被用在手動模式。
2. 暫存器 INTC2 的 Bit2 紀錄觸控的狀態。當觸控事件發生時，此位元被設定為 1。請特別注意，當無觸控事件發生時，INTC2 的 Bit2 不會被自動清除為 0，須由程式設計師來清除。此位元功能通常用在外部中斷模式。

註：STSR 的 Bit5 是由 ADC 電路的直接輸出，只要有螢幕被碰觸，此位元會被設定為 1。若碰觸狀態還不穩定，需要消除機械彈跳，來確保此一碰觸是有效的觸控事件。因此，STSR 的 Bit5 只在手動模式下動作。當設定 RA8875 為自動模式時，觸控事件將自動被偵測，並且由系統來檢查是否為有效事件，只要是有效的觸控事件，中斷才會產生。

7-8-3 觸控掃描與取樣時間

在使用觸控螢幕功能的自動模式且觸控事件發生時，RA8875 採用特定的等待取樣時間讓 X, Y Data 穩定。建議選擇適當地觸控取樣時間以避免 ADC 資料門鎖，請參照下表的 ADC 取樣時間與轉換速度對照表。

表 7-20：取樣時間與轉換速度對照表

Touch Panel Sampling Time - REG[70h] Bit[6:4]					
SYS_CLK REG[70h] [2:0]	10M	20M	30M	40M	50M
000b	000	--	--	--	--
001b	000	--	--	--	--
010b	000	000	000	--	--
011b	001	001	000	000	000
100b	010	010	001	001	001
101b	011	011	010	010	010
110b	100	100	011	011	011
111b	101	101	100	100	100

註：ADC 的輸入時脈設定不能超過 10MHz。

7-9 鍵盤

RA8875 的鍵盤掃描控制器提供一個更靈敏的鍵盤應用介面，相關的暫存器為 KSCR(REG[C0h],[C1h])、KSDR(REG[C2h], [C3h], [C4h])，鍵盤掃描功能具有下列特色：

1. 支援到 4x5 鍵盤模組。
2. 可用程式自行設定取樣次數 (Sampling Times) 與鍵盤掃描的頻率。
3. 可調整長按鍵 (Long key-press) 之時序。
4. 允許多重按鍵 (Multi-Key) 組合，最多同時允許三個按鍵組合。
5. 當系統在睡眠模式時，允許按鍵來喚醒 (Wake-up) 系統。

KSCR是鍵盤掃描控制與狀態暫存器，是用來設定鍵盤掃描功能的選項，例如資料取樣時間、取樣時脈頻率或開啟長按鍵功能等。當按鍵動作時，使用者可以感覺到來自鍵盤掃描的中斷。KSCR2(REG[C1h]) 的 bit1~0 會更新目前按鍵的號碼。之後使用者可以直接得到對應碼 (Key Code)。表 7-21是鍵盤矩陣中每個短按鍵的對應碼 (Key Code)，當有案件發生短按時，按鍵的對應碼將被儲存在KSDR0~2(REG[C2h~C4h])。至於長按鍵(Long Time Press) 的對應碼請參考表 7-22。

表 7-21：短按鍵的對應碼 (Normal Key)

		Column# (KIN#)				
		C0	C1	C2	C3	C4
Row# (KOUT#)	R0	00h	01h	02h	03h	04h
	R1	10h	11h	12h	13h	14h
	R2	20h	21h	22h	23h	24h
	R3	30h	31h	32h	33h	34h

表 7-22：長按鍵的對應碼 (Long Key)

		Column# (KIN#)				
		C0	C1	C2	C3	C4
Row# (KOUT#)	R0	80h	81h	82h	83h	84h
	R1	90h	91h	92h	93h	94h
	R2	A0h	A1h	A2h	A3h	A4h
	R3	B0h	B1h	B2h	B3h	B4h

註：短按鍵 (Normal Key) 指的是鍵盤的按壓是透過 RA8875 的取樣時間。長按鍵(Long Key) 指的是鍵盤按壓維持一段較長的時間。所以要長按鍵前要先進行短按鍵，有時在某些應用上會分開使用兩種功能。

當應用多重組合按鍵時，最多可儲存三個按鍵的對應碼，存在暫存器 KSDR0、KSDR1 和 KSDR2，值得注意的是，數個對應碼儲存在暫存器主要是以對應碼值大小來排序，而與先後按鍵順序無關，請參考下面的範例：

假設先後按下三個鍵，其對應碼分別為 0x34、0x00、0x22，則暫存器 KSDR0~2 所儲存的内容如下：

KSDR0 = 0x00
KSDR1 = 0x22
KSDR2 = 0x34

針對鍵盤掃描功能相關的暫存器，列表如下：

表 7-23

Reg.	Bit_Num	Description	Reference
KSCR1	Bit 7	Key-Scan enable bit	REG[C0h]
	Bit 6	Long Key Enable bit	
	Bit [5:4]	Key-Scan sampling times setting	
	Bit [2:0]	Key-Scan scan frequency setting	
KSCR2	Bit [7]	Key-Scan Wakeup Function Enable Bit	REG[C1h]
	Bit [3:2]	long key timing adjustment	
	Bit [1:0]	The number of key hit	
KSDR0 KSDR1 KSDR2	Bit [7:0]	Key code for pressed key	REG[C2h ~ C4h]
INTC1	Bit 4	Key-Scan interrupt enable	REG[F0h]
INTC2	Bit 4	Key-Scan Interrupt Status bit	REG[F1h]

當啟動鍵盤掃描的功能之後，程式設計師可以使用兩種方法來檢查按鍵是否被按：

- 1) **軟體檢查的方式**：不斷檢查暫存器(INTC2 REG[F1h] 的 Bit-4) 來得知是否有按鍵被按。
- 2) **硬體檢查的方式**：由外部中斷的產生來得知有按鍵被按。

值得注意的是當開啟鍵盤掃描中斷時，INTC1 的位元 4 都將被設定為 1，而且鍵盤中斷事件發生時，鍵盤掃描的中斷狀態 INTC2 的位元 4 永遠都設為 1，無論使用哪種方法，因此程式設計師在正確讀回按鍵的對應碼之後，必須將該位元清除為 0，否則之後的按鍵將無法正常發出中斷告知系統。

此外，RA8875 允許在睡眠模式時使用鍵盤喚醒功能 (Key-stroke Wakeup Function)。透過開啟功能設定，任何正當的鍵盤事件皆可從睡眠模式中喚醒RA8875。為了判斷喚醒事件，RA8875 可以顯示MCU的硬體中斷，MCU可以使用RA8875 的軟體流程 (Software Polling)。表 7-24 列出相關暫存器的功能描述。

表 7-24

Reg.	Bit_Num	Description	Reference
KSCR2	Bit 7	Enable Key-Scan wake-up function	REG[C1h]
INTC1	Bit 4	Wake-up interrupt enable bit	REG[F0h]
INTC2	Bit4	Key-Scan Interrupt Status bit	REG[F1h]

針對以上的應用，有關暫存器設定的流程圖如下：

1. 軟體檢查方式

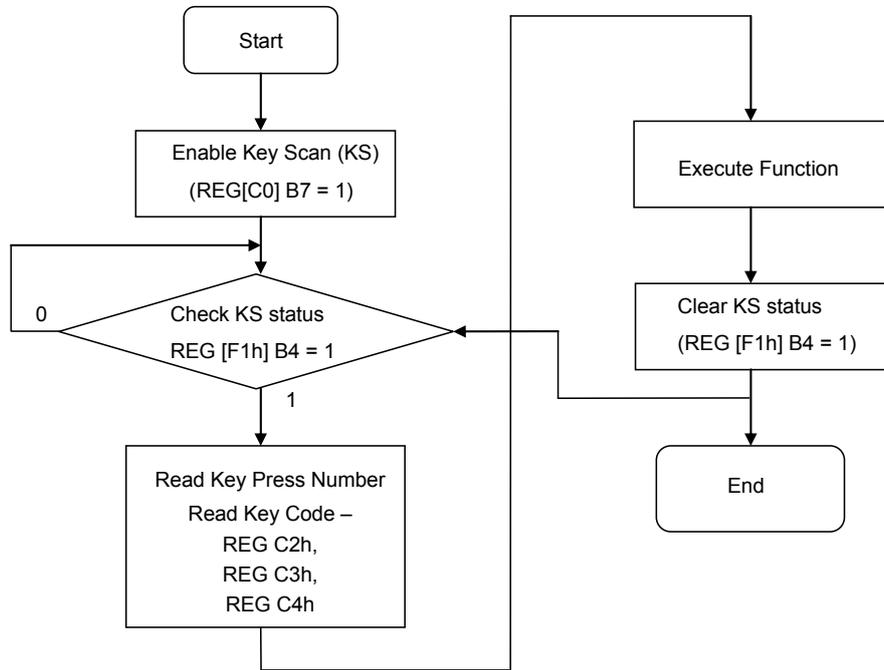


圖 7-80：鍵盤掃描功能的流程圖 _ 軟體輪詢

2. 硬體檢查方式

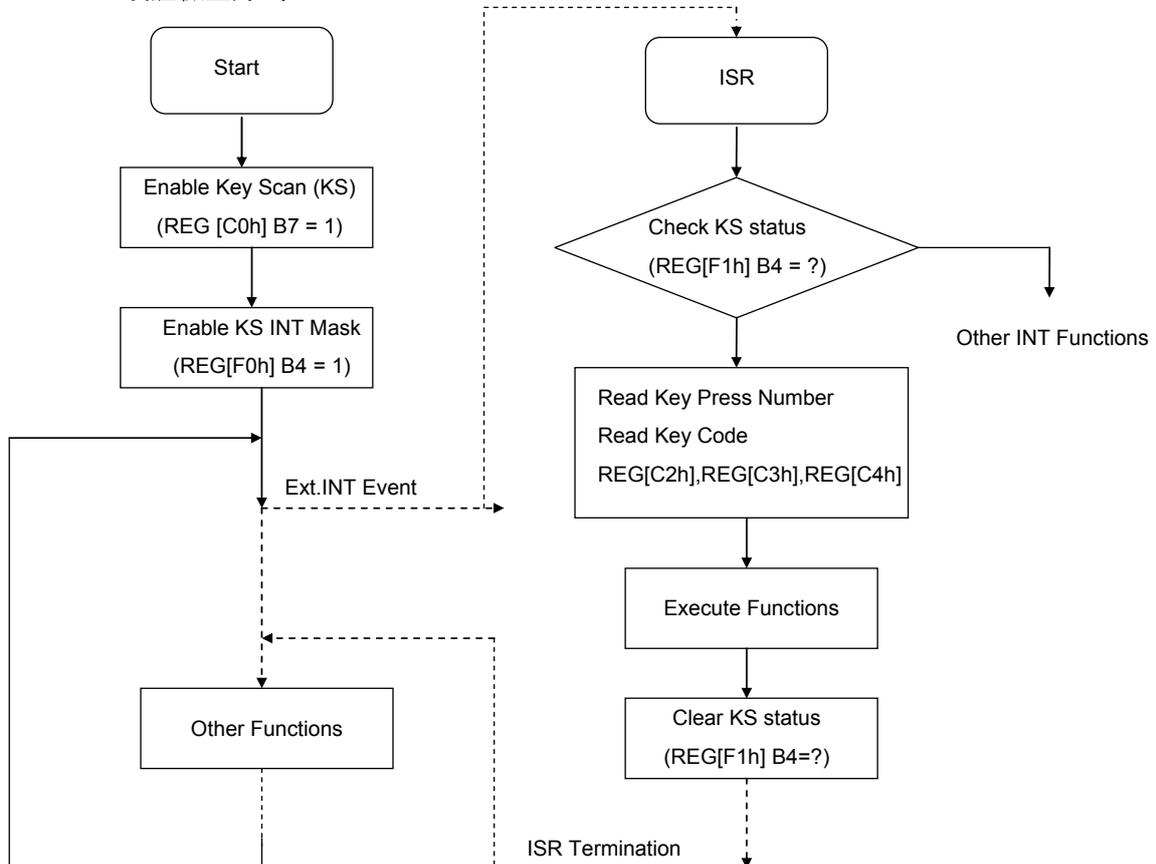


圖 7-81：鍵盤掃描流程圖 _ 硬體中斷

7-10 記憶體直接存取功能

記憶體直接存取功能提供使用者更快速、搬移大量的資料到顯示記憶體的方法。在RA8875 中，記憶體直接存取功能的資料來源是外部「Serial Flash/ROM介面」。可為兩種資料格式：「連續資料模式」與「區塊資料模式」，提供使用者更靈活的應用。記憶體直接存取資料寫入位置則依照在顯示記憶體中所設定的工作視窗，當「Serial Flash/ROM介面」特定資料是依照色彩深度設定(REG[10h] Bit 3-2)，請參考 圖 7-82。當此功能運作時，Serial Flash/ROM裡所指定的資料會自動地一個接一個搬移到顯示記憶體裡，執行完成後，中斷訊號則將被觸發而去通知MCU，請參照以下的章節。

24'h000	R2	R1	R0	G2	G1	G0	B1	B0
24'h001	R2	R1	R0	G2	G1	G0	B1	B0
24'h002	R2	R1	R0	G2	G1	G0	B1	B0
24'h003	R2	R1	R0	G2	G1	G0	B1	B0
·				·				
·				·				
·				·				

the specific 8-bit data in serial Flash/ROM

24'h000	R4	R3	R2	R1	R0	G5	G4	G3
24'h001	G2	G1	G0	B4	B3	B2	B1	B0
24'h002	R4	R3	R2	R1	R0	G5	G4	G3
24'h003	G2	G1	G0	B4	B3	B2	B1	B0
·				·				
·				·				
·				·				

the specific 16-bit data in serial Flash/ROM

圖 7-82 : Serial Flash/ROM 介面的特定資料

7-10-1 連續記憶體直接存取模式

在此模式下，記憶體直接存取控制器從所設定在 Serial Flash/ROM 資料來源起始到結束位置 (SSAR)，加上記憶體直接存取搬移資料數目(DTNR) 讀取資料。使用者只需要設定工作視窗後，便可以將資料搬移到顯示記憶體裡。

使用方法：

1. 設定工作視窗範圍 (REG[30h] ~REG[37h])和記憶體寫入位置 (REG[46h] ~REG[49h])
2. 設定 Serial Flash/ROM 組態 (REG[05h])
3. 設定記憶體直接存取資料來源起始位置 (REG[B0h] ~REG[B2h])
4. 設定記憶體直接存取資料搬移數目 (REG[B4h], REG[B6h] 和 REG[B8h])
5. 開啟記憶體直接存取起始訊號和檢查記憶體直接存取忙碌訊號 (REG[BFh] bit 0)

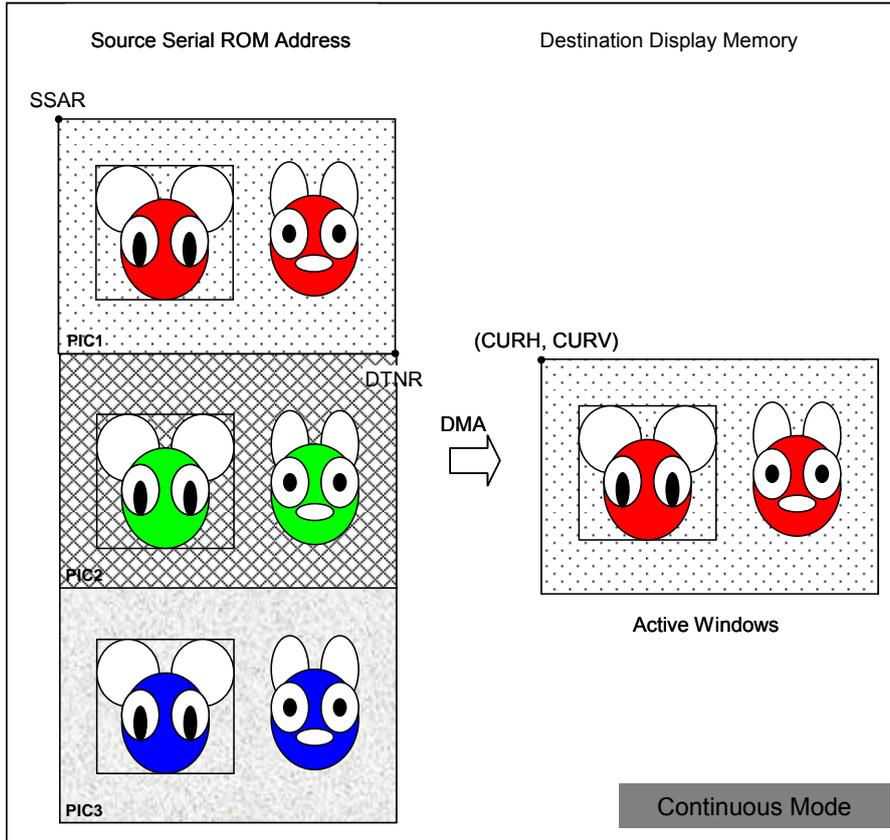


圖 7-83 : 連續資料記憶體存取模式

7-10-2 區塊資料記憶體直接存取模式

在此模式下，使用者可以靈活地讀取區塊資料。記憶體直接存取控制器從所設定在 Serial Flash/ROM 資料來源起始到結束位置 (SSAR) 和依照區塊寬度設定值(BWR)，區塊高度設定值 (BHR) 和來源圖片寬度 (SPWR) 來計算區塊位置。使用者只需要設定工作視窗，便可將資料搬移到顯示記憶體裡。

1. 設定工作視窗範圍 (REG[30h] ~REG[37h])和記憶體寫入位置 (REG[46h] ~REG[49h])
2. 設定 Serial Flash/ROM 組態 (REG[05h])
3. 設定 記憶體直接存取資料來源起始位置 (REG[B0h] ~REG[B2h])
4. 設定 記憶體直接存取區塊寬度 (REG[B4h] 和 REG[B5h])
5. 設定 記憶體直接存取區塊高度 (REG[B6h] 和 REG[B7h])
6. 設定記憶體直接存取來源圖片寬度 (REG[B8h] 和 REG[B9h])
7. 開啟記憶體直接存取為區塊搬移模式 (REG[BFh] bit 1)
8. 開啟記憶體直接存取起始訊號且檢查記憶體直接存取忙碌訊號 (REG[BFh] bit 0)

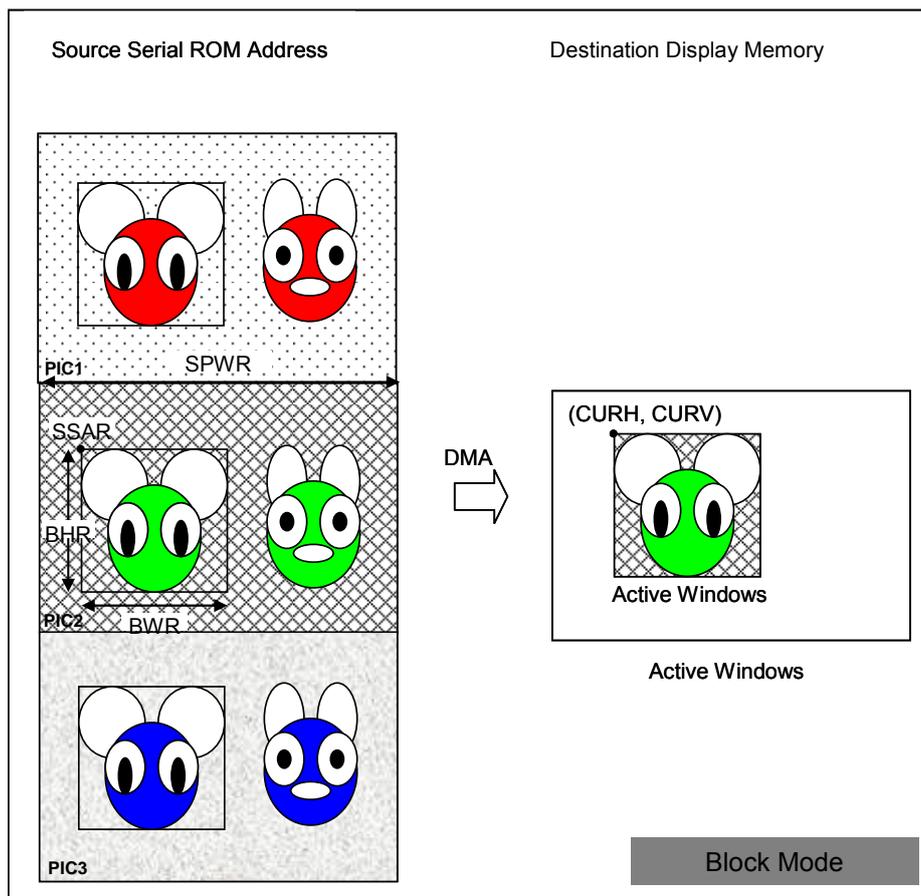


圖 7-84 : 區塊資料記憶體直接存取模式

7-11 脈寬調變

RA8875 提供二個可調節的脈寬調變 (PWM) 輸出，其PWM的頻率和工作週期 (Duty Cycle) 都可以透過相關暫存器的設置來調整，如果PWM功能被禁能 (Disable)，此腳位也可當成一般的IO信號使用，相關的功能設定，請參考以下的表 7-25。

表 7-25 : PWM 設定

暫存器	Bit_Num	說明	暫存器編號
P1CR	Bit7	PWM1 功能致能位元	REG[8Ah]
	Bit6	PWM1 關閉時的準位	
	Bit[3:0]	PWM1 來源時脈的除頻設定	
P1DCR	Bit[7:0]	PWM1 工作週期(Duty Cycle) 選擇	REG[8Bh]
P2CR	Bit7	PWM2 功能致能位元	REG[8Ch]
	Bit6	PWM2 關閉時的準位	
	Bit[3:0]	PWM2 來源時脈的除頻設定	
P2DCR	Bit[7:0]	PWM2 工作週期(Duty Cycle) 選擇	REG[8Dh]

RA8875 的二個可程式化PWM輸出都可獨立控制，暫存器 REG[8Bh] 與REG[8Dh] 分別調整它們的Duty輸出，最常用的就是拿來作TFT面板的背光控制，請參考章節 6-6 的圖 6-50。下圖示兩個關於PWM輸出的例子：

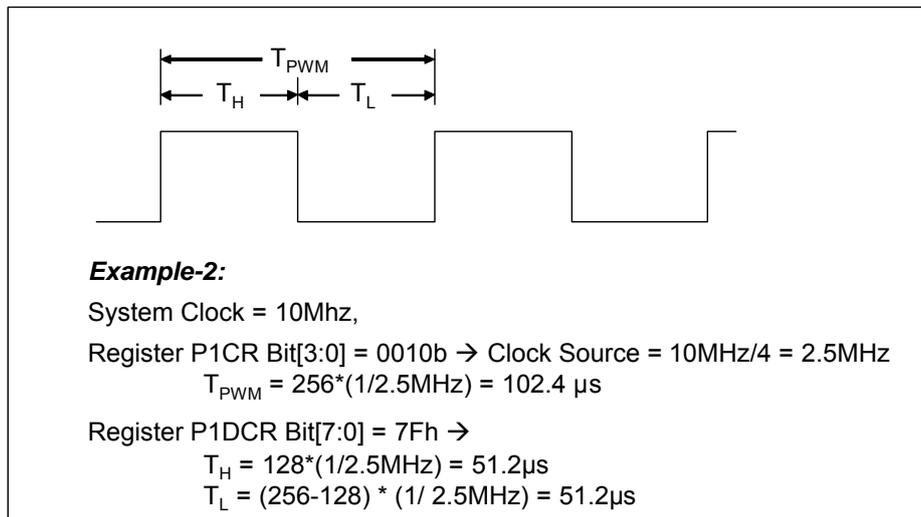


圖 7-85 : PWM 輸出脈衝範例一

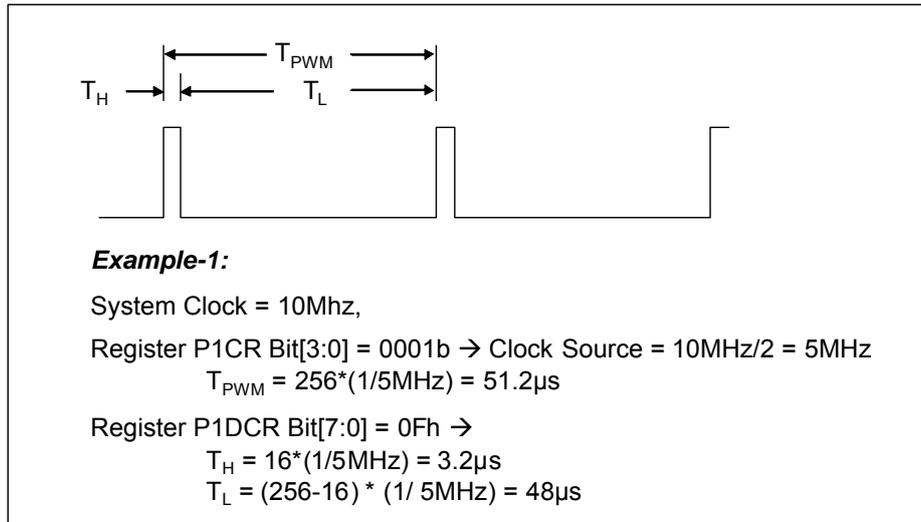


圖 7-86 : PWM 輸出脈衝範例二

7-12 睡眠模式

RA8875 提供睡眠模式 (Sleep Mode)，在沒有使用的狀態下，使用者可減少 RA8875 或 LCD 模組的功耗。在睡眠模式下，除了少許的靜態電流外，系統時序、內部記憶體 (如 DDRAM)、Font ROM 等都會關閉，PWM 的輸出準位也會維持原先暫存器的設定。

若要從睡眠模式喚醒 (Wake-up)，有三種方式：

1. 暫存器設定：利用 MCU 將暫存器 [01h] 的 Bit1 設為 0。
2. 觸控面板：進入睡眠模式前，利用 MCU 將暫存器 [70h] 的 Bit7 與 Bit3 設為 1。需注意觸控面板設定為手動模式時，等待觸控面板事件模式 (Wait for TP event) 須在進入睡眠模式前被設定，否則不會偵測到觸控事件便無法喚醒 RA8875。
3. 鍵盤：類似前面提到的喚醒觸控面板，鍵盤功能的致能位元及鍵盤喚醒功能都應先被設定。暫存器 [C0h] 的 Bit7 及 [C1h] 的 Bit7 在進入睡眠模式前皆設為 1。需注意當 RA8875 離開睡眠模式時，所按壓的鍵碼並不會被記錄在 RA8875 中。

睡眠模式啟動時，在存取 RA8875 前建議需等待一段時間。因為喚醒後晶體振盪電路器及 PLL 電路會被重新啟動，因此要有一段時間等待系統時脈 (System Clock) 穩定，RA8875 才能接受 MCU 下的指令，此時間約 10ms 左右，下表為相關暫存器的說明。

表 7-26：睡眠與喚醒模式相關的暫存器設定

暫存器	Bit_Num	說 明	暫存器編號
PWRR	Bit1	Sleep Mode 0 : Normal mode. 1 : Sleep mode.	REG[01h]
TPCR0	Bit7	Touch Panel Enable Bit 0 : Disable 1 : Enable	REG[70h]
	Bit3	Touch Panel Wakeup Enable 0 : Disable the Touch Panel wake-up function. 1 : Touch Panel can wake-up the sleep mode.	
KSCR1	Bit7	Key-Scan Enable Bit(KEY_EN) 0 : Disable. 1 : Enable.	REG[C0h]
KSCR2	Bit7	Key-Scan Wakeup Function Enable Bit 0: Key-Scan Wakeup function is disable. 1: Key-Scan Wakeup function is enable.	REG[C1h]

RA8875 在睡眠模式時，相關輸出信號的狀態如表 7-27 所示。

表 7-27：睡眠模式時相關輸出信號的狀態

信號名稱	輸出狀態
WAIT#	High
INT#	High
PWM1, PWM2	Low
GPIO[5:0]	Low
VA[18:0]	Low
RAM_OE#	Low
RAM_CS#, RAM_WR#, ROM_CS#	High
PDAT[15:0]	Low
VSYNC, HSYNC	High
PCLK, DE	High

8. AC/DC 特性

8-1 最大範圍限制

表 8-1：最大範圍

參數描述	符號	規格範圍	單位
Supply Voltage Range (Note 4)	V _{DDP} OSC_VDDP ADC_VDD	-0.3V~4.0V	V
Input Voltage Range	V _{IN}	-0.3 to V _{DD} +0.3	V
Power Dissipation	P _D	≤ 150	mW
Operation Temperature Range	T _{OPR}	-30 to +85	°C
Storage Temperature	T _{ST}	-45 to +125	°C
Soldering Temperature (10 seconds, Note 1)	T _{SOLDER}	260	°C

註：

1. 假如該封裝被焊料侵入，平薄式封裝的濕度抵抗性是會減少的。當進行焊接作業時，勿過度施壓於封裝上。
2. 當供應電源端為高阻抗時，供應電源/輸入電源可能存在著一個很大壓差，須適度考量 RA8875 的電源端及其電源接線及佈局。
3. 全部的電源準位皆以 GND = 0V 為基準。
4. CORE_VDD、LDO_OUT、OSC_VDD 為電源輸出，不包含在內。

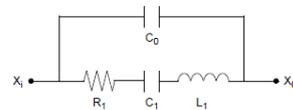
8-2 DC 電氣特性

表 8-2 : DC 電氣特性

參數描述	符號	最小值	一般值	最大值	單位	工作條件
System Voltage(V_{DD3})	ADC_VDD VDDP	3.0	3.3	3.6	V	
Core Voltage(V_{DD18})	LDO_OUT CORE_VDD	1.6	1.8	2.0	V	Add External 1uF Capacitor
ADC Reference Voltage	ADC_VREF	--	$0.5V_{DD3}$ (±5%)	--	V	Add External 1uF Capacitor
Oscillator Clock	F_{OSC}	15	20	30	MHz	$V_{DD3} = 3.3 V$ (Note 3)
PLL Output Clock	SYS_CLK	1	20~30	60	MHz	$V_{DD3} = 3.3 V$
Input						
Input High Voltage	V_{IH}	$0.8 V_{DD3}$	--	V_{DD3}	V	
Input Low Voltage	V_{IL}	GND	--	$0.2 V_{DD3}$	V	
Output						
Output High Voltage	V_{OH}	$V_{DD}-0.4$	--	V_{DD3}	V	
Output Low Voltage	V_{OL}	GND	--	$G_{ND} + 0.4$	V	
Schmitt-Trigger Input (註 1)						
Input High Voltage	V_{IH}	$0.7 V_{DD3}$	--	V_{DD3}	V	
Input Low Voltage	V_{IL}	GND	--	$0.3 V_{DD3}$	V	
Input Leakage Current 1	I_{IH}	--	--	+2	μA	(註 2)
Input Leakage Current 2	I_{IL}	--	--	-2	μA	(註 2)
Operation Current	I_{OPR}	20	--	50	mA	
Sleep Mode	I_{SLP}	--	320	--	μA	(註 2)

註：

- 腳位 RD#、WR#、CS#、RS 與 RST# 為 Schmitt-trigger 輸入。
- 測試條件如下： $VDDP = VDD3 = 3.3V$, Oscillator Clock = 25MHz, System Clock = 20~60MHz, Source = 800, Gate = 480, VSYNC = 45~65Hz, $TA=25^{\circ}C$.
- 使用石英震盪器的參考數值如下：
R1: 20 ~ 75 OHM
C1: 22 ~ 25 fF
L1: 1.125 ~ 5.117 mH
C0: 7pf



9. 產品封裝

9-1 封裝腳位圖

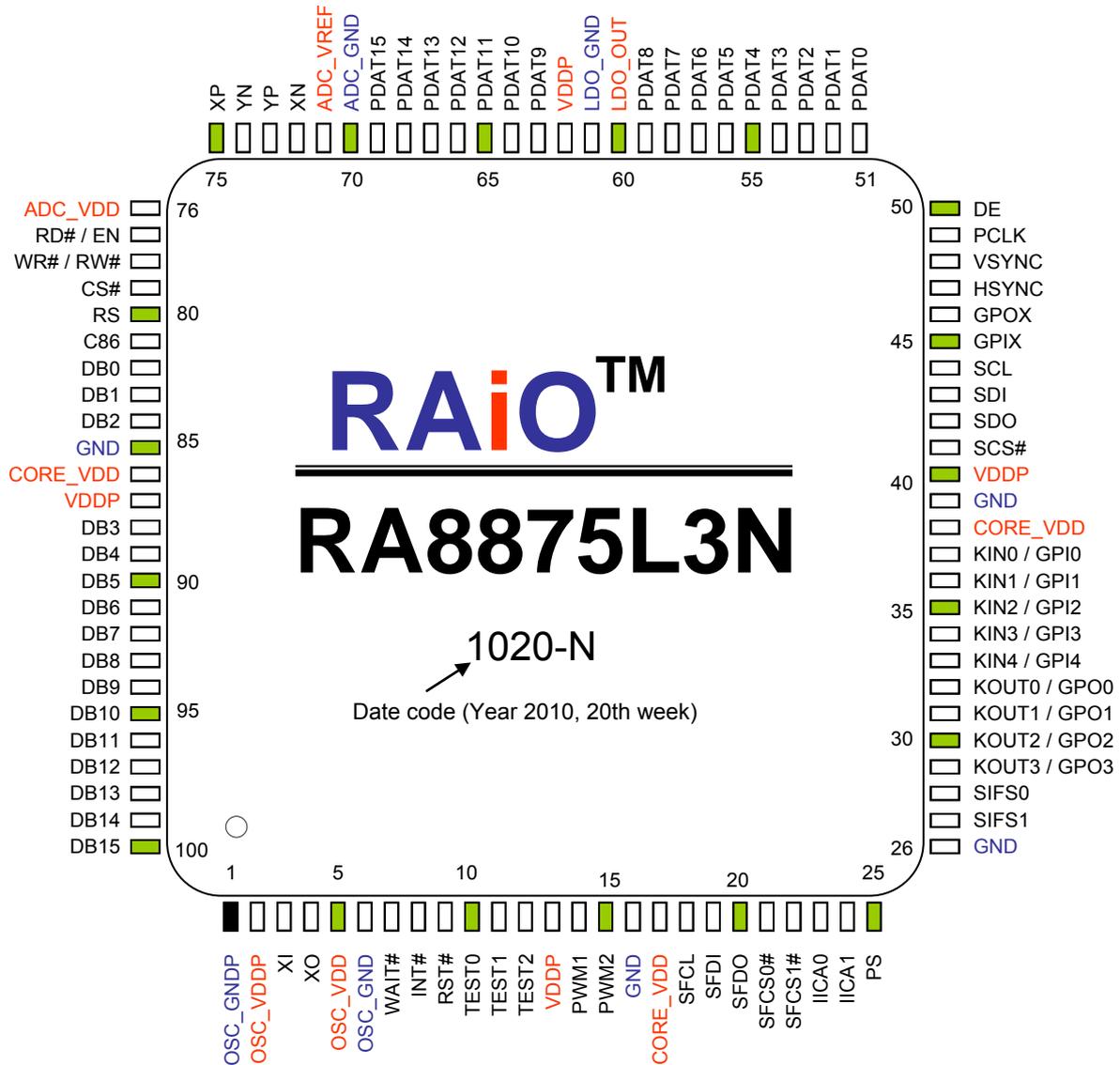


圖 9-1 : RA8875 的接腳圖

9-2 封裝尺寸

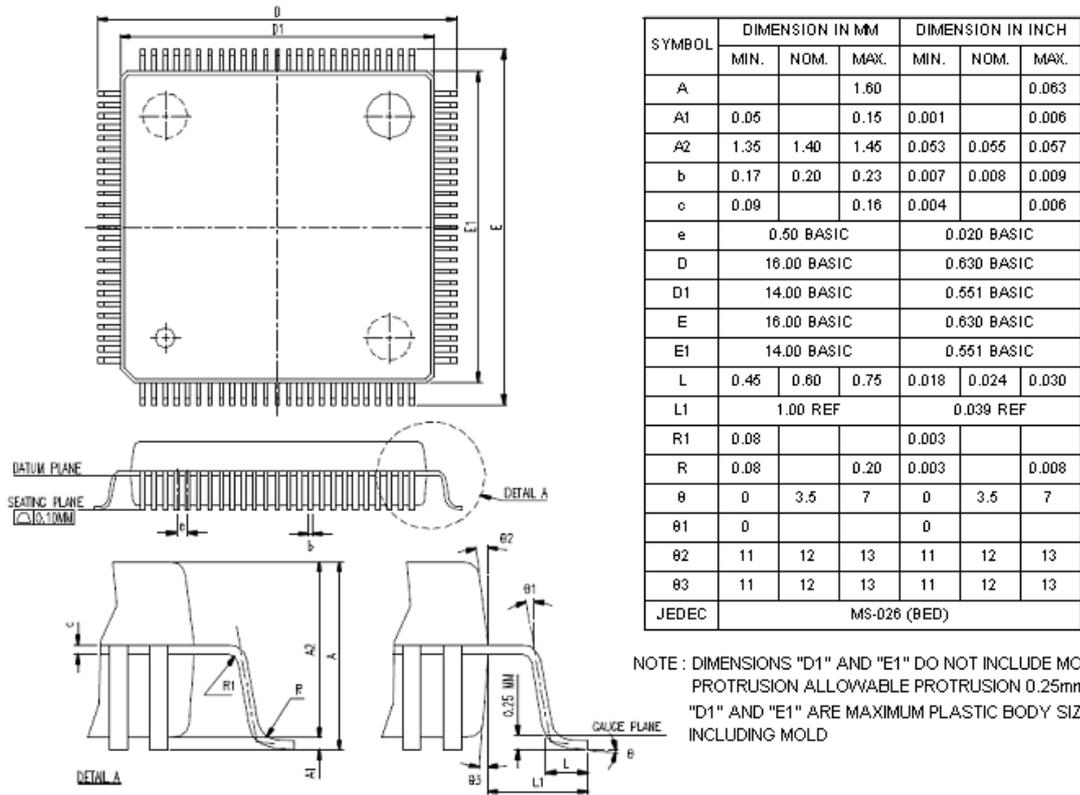


圖 9-2 : RA8875 的外觀尺寸

9-3 產品編號

RA8875 產品完整編號為“RA8875L3N”。其符合環保 RoHS 規範，並通過歐盟要求的 PFOS、PFOA 及 REACH 的檢測，符合的有害物質限用管理指令如下：

- 歐盟有害物質限用指令 RoHS (2002/95/EC)
- 歐盟限制全氟辛烷磺酸的銷售與使用指令 PFOS & PFOA (2006/122/EC)
- 歐盟化學品政策 REACH (EC 1907/2006)

Appendix A. RA8875 支援的集通字型列表

Table A- 1

● : 支援, - : 不支援

GT21L16T1W 支援字型	RA8875 是否支援	備註
15X16 點 GB12345 漢字字符	●	
15X16 點 BIG5 基本集漢字字符	●	
15X16 點 JIS 日文字符	●	部份字型(Table A-2), 集通以軟體方式補丁, RA8875 不支援這些字型
15X16 點 Unicode 日文字符	●	
5X7 點 ASCII 字符	-	
7X8 點 ASCII 字符	-	
6X12 點 ASCII 字符	-	
8X16 點 ASCII 字符	●	
8X16 點粗體 ASCII 字符	●	
12 點不等寬 ASCII 方塊(Arial)字符	-	
16 點陣不等寬 ASCII 方塊(Arial)字符	●	
8X16 點拉丁文系字符	●	
8X16 點希臘文系字符	●	
8X16 點基里爾文系字符	●	
12 點陣不等寬拉丁文系字符	-	
12 點陣不等寬希臘文系字符	-	
12 點陣不等寬基里爾文系字符	-	
16 點陣不等寬拉丁文系字符	●	
16 點陣不等寬希臘文系字符	●	
16 點陣不等寬基里爾文系字符	●	
12 點陣不等寬阿拉伯文系字符	-	
12 點陣不等寬阿拉伯文系變體字符	-	
16 點陣不等寬阿拉伯文系字符	●	
16 點陣不等寬阿拉伯文系變體字符	●	

Table A- 2 : JIS0208 字碼表 (RA8875 沒有支援)

	≤	≥	♁	▽	▼	o	き	ぎ	溯
0135	0169	0170	0173	0206	0207	0379	0413	0414	3344
墮	陳	梯	届	汎	篋	墨	冀	寫	幕
3436	3636	3680	3847	4038	4247	4347	4935	4948	4949
刳	卮	哈	營	塢	幫	憇	掀	斛	暫
4974	5036	5093	5159	5229	5483	5660	5756	5847	5881
桿	淦	箏	紉	繡	閭	霖	騙	熙	。
5969	6232	6823	6913	6962	7967	8035	8157	8406	8503
	≤	≥	♁	¥					
8565	8569	8570	8573	8579					

Table A- 3

GT30L24M1Z 支援字型	RA8875 是否支援	備註
24X24 點 GB18030 標準點陣字庫	●	
12X24 點國際擴展字符	●	
12X24 點 ASCII 字符	●	
24 點陣不等寬 ASCII 方塊(Arial)字符	●	
24 點陣不等寬 ASCII 白正(Times New Roman)字符	●	

Table A- 4

GT30L32S4W 支援字型	RA8875 是否支援	備註
11X12 點 GB2312 標準點陣字庫	—	
15X16 點 GB2312 標準點陣字庫	●	
24X24 點 GB2312 標準點陣字庫	●	
32X32 點 GB2312 標準點陣字庫	●	
6X12 點國際擴展字符	—	
8X16 點國際擴展字符	●	
8X16 點 GB2312 特殊字符	●	
12X24 點國際擴展字符	●	
16X32 點國際擴展字符	●	
5X7 點 ASCII 字符	—	
7X8 點 ASCII 字符	—	
6X12 點 ASCII 字符	—	
8X16 點 ASCII 字符	●	
12X24 點 ASCII 字符	●	
16X32 點 ASCII 字符	●	
12 點陣不等寬 ASCII 方塊(Arial)字符	—	
12 點陣不等寬 ASCII 白正(Times New Roman)字符	—	
16 點陣不等寬 ASCII 方塊(Arial)字符	●	
16 點陣不等寬 ASCII 白正(Times New Roman)字符	●	
24 點陣不等寬 ASCII 方塊(Arial)字符	●	
24 點陣不等寬 ASCII 白正(Times New Roman)字符	●	
32 點陣不等寬 ASCII 方塊(Arial)字符	●	
32 點陣不等寬 ASCII 白正(Times New Roman)字符	●	

Table A- 5

GT30L16U2W 支援字型	RA8875 是否支援	備註
11X12 點 Unicode 點陣字庫	—	
15X16 點 Unicode 點陣字庫	●	
8X16 點特殊字符	●	
5X7 點 ASCII 字符	—	
7X8 點 ASCII 字符	—	
6X12 點 ASCII 字符	—	
8X16 點 ASCII 字符	●	
12 點陣不等寬 ASCII 方塊(Arial)字符	—	
12 點陣不等寬 ASCII 白正(Times New Roman)字符	—	
16 點陣不等寬 ASCII 方塊(Arial)字符	●	
16 點陣不等寬 ASCII 白正(Times New Roman)字符	●	
8X16 點拉丁文系字符	●	
8X16 點希臘文系字符	●	
8X16 點基里爾文系字符	●	
12 點陣不等寬拉丁文系字符	—	
12 點陣不等寬希臘文系字符	—	
12 點陣不等寬基里爾文系字符	—	
12 點陣不等寬阿拉伯文系字符	—	
12 點陣不等寬阿拉伯文系變體字符	—	
16 點陣不等寬拉丁文系字符	●	
16 點陣不等寬希臘文系字符	●	
16 點陣不等寬基里爾文系字符	●	
16 點陣不等寬阿拉伯文系字符	●	
16 點陣不等寬阿拉伯文系變體字符	●	

Table A- 6

GT30L24T3Y 支援字型	RA8875 是否支援	備註
11X12 點 GB2312 標準點陣字庫	—	
15X16 點 GB2312 標準點陣字庫	●	
24X24 點 GB2312 標準點陣字庫	●	
11X12 點 GB12345 字符集漢字字 符	—	
15X16 點 GB12345 字符集漢字字 符	●	
24X24 點 GB12345 字符集漢字字 符	●	
11X12 點 BIG5 字符集漢字字符	—	
15X16 點 BIG5 字符集漢字字符	●	
24X24 點 BIG5 基本集漢字字符	●	
11X12 點 Unicode 中文字符	—	
15X16 點 Unicode 中文字符	●	
24x24 點 Unicode 中文字符	●	
5X7 點 ASCII 字符	—	
7X8 點 ASCII 字符	—	
6X12 點 ASCII 字符	—	
8X16 點 ASCII 字符	●	
12 點陣不等寬 ASCII 方塊(Arial)字符	—	
16 點陣不等寬 ASCII 方塊(Arial)字符	●	
24 點陣不等寬 ASCII 方塊(Arial)字符	●	