

RAiO

RA8860

80x2 點 16 色

LCD 驅動器

規 格 書

Version 1.3

August 8, 2013

RAiO Technology Inc.

©Copyright RAiO Technology Inc. 2010, 2011, 2012, 2013

改 版 記 錄		
版 本	日 期	說 明
1.0	May 4, 2010	初版發行
1.1	September 28, 2010	1. 新增第一章及第二章封裝方式說明 2. 新增圖 3-3、表 3-1、圖 3-4 3. 新增表 7-6 4. 更新圖 8-1、圖 8-2、圖 8-3
1.2	March 15, 2011	1. 新增圖 7-9 2. 插入第九章 - 初始化暫存器設定程序 3. 更新第十章 - 示範程式 (LCD_Initial())
1.3	May 31, 2011	1. 更新表 6-1、表 7-4
	August 8, 2013	1. 新增圖 3-2 注釋

章 節	內 容	頁 數
1.	簡介	5
2.	功能	5
3.	接腳說明	5
3-1	內部方塊圖	5
3-2	腳位圖	6
3-3	腳位座標	8
4.	腳位定義	10
5.	電氣特性	12
5-1	最大範圍限制	12
5-2	DC電氣特性	12
5-3	AC電氣特性	13
6.	暫存器與指令	15
7.	功能描述	24
7-1	MCU界面	24
7-1-1	I2C界面	25
7-1-2	3 線SPI	26
7-1-3	4 線SPI (Type-A)	27
7-1-4	4 線SPI (Type-B)	28
7-1-5	記憶體讀取	29
7-2	串接模式	30
7-3	時脈與內部RC振盪器	31
7-4	顯示資料 (Display Data)	32
7-4-1	顯示記憶體 (Display Data RAM)	32
7-4-2	8 色的資料格式	33
7-4-3	16 色的資料格式	33
7-5	閃爍功能	35
7-6	重置 (Reset)	35
7-7	LCD驅動器與電壓供應電路	36
7-7-1	步階升壓電路 (Booster)	37
7-7-2	電壓調整器電路 (Regulator)	38
7-7-3	電壓隨耦器 (Voltage Follower)	40
7-8	LCD驅動輸出	41
7-9	LED驅動輸出	42
7-10	重置 (Reset) 與休眠 (Sleep) 時的輸出信號	44
8.	應用電路	45
9.	初始化暫存器設定程序	47

9-1	主模式初始化程序	47
9-2	電源開啓程序.....	48
9-3	進入休眠模式與喚醒程序.....	49
9-4	主從模式(串接)初始化程序	50
9-5	Note 1) 示範程式.....	51
10.	示範程式.....	52

1. 簡介

RA8860 是一 LCD 驅動控制器，專門用於彩色 TN 或稱為場序驅動顯示的 LCD 面板，藉由 LCD 與 LED 背光的驅動模式可創造出 16 種顏色的顯示效果。它包括了 80x2 的驅動電路、倍壓電路、顯示記憶體、LED 驅動電路及支援高速 3 線、4 線串聯界面 (SPI) 和 I²C 界面，可直接與 MCU 連接。除此之外 RA8860 也提供了多顆串接模式，讓控制 LCD 點數倍增，而 COB 的裸片封裝更可讓使用者快速的推出產品，節省開發時間及費用。針對終端應用不同的需求，也提供 LQFP 128pin 封裝。在彩色 TN LCD 的應用上 RA8860 是一兼具低成本與實用性的選擇。

2. 功能

- ◆ 80SEG x 2COM 驅動輸出
- ◆ 支援串聯 I²C、3/4 線 MCU 界面
- ◆ 支援 8 或 16 色顯示
- ◆ 內建 640-bits 顯示記憶體
- ◆ Duty : Static, 1/2 Duty
- ◆ 內建 Voltage Booster (4X) 與 Voltage Follower，供應 LCD 驅動電壓
- ◆ 支援閃爍模式
- ◆ 內建 RGB LED 驅動輸出，最大驅動電流可達 80mA
- ◆ 內建 RC 振盪器
- ◆ 支援多顆串接模式 (Master / Slave 模式)
- ◆ 封裝: COB Chip 及 LQFP-128 pin
- ◆ 供應電壓 : 2.7~5.5V

3. 接腳說明

3-1 內部方塊圖

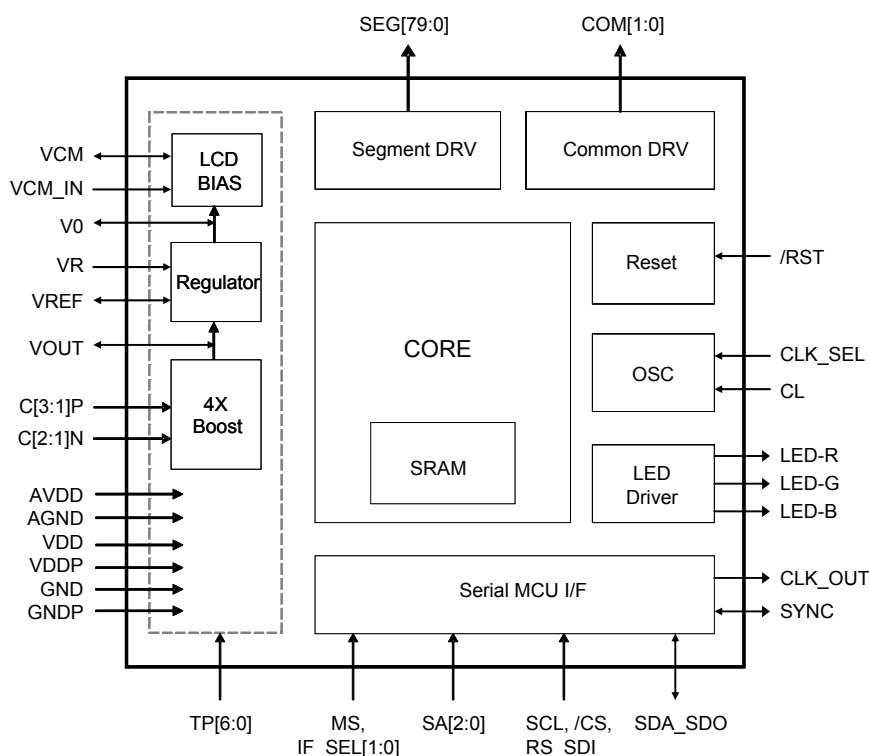


圖 3-1 : RA8860 內部方塊圖

3-2 腳位圖

注: 腳位圖如下圖，請將 substrate 接地或浮接。

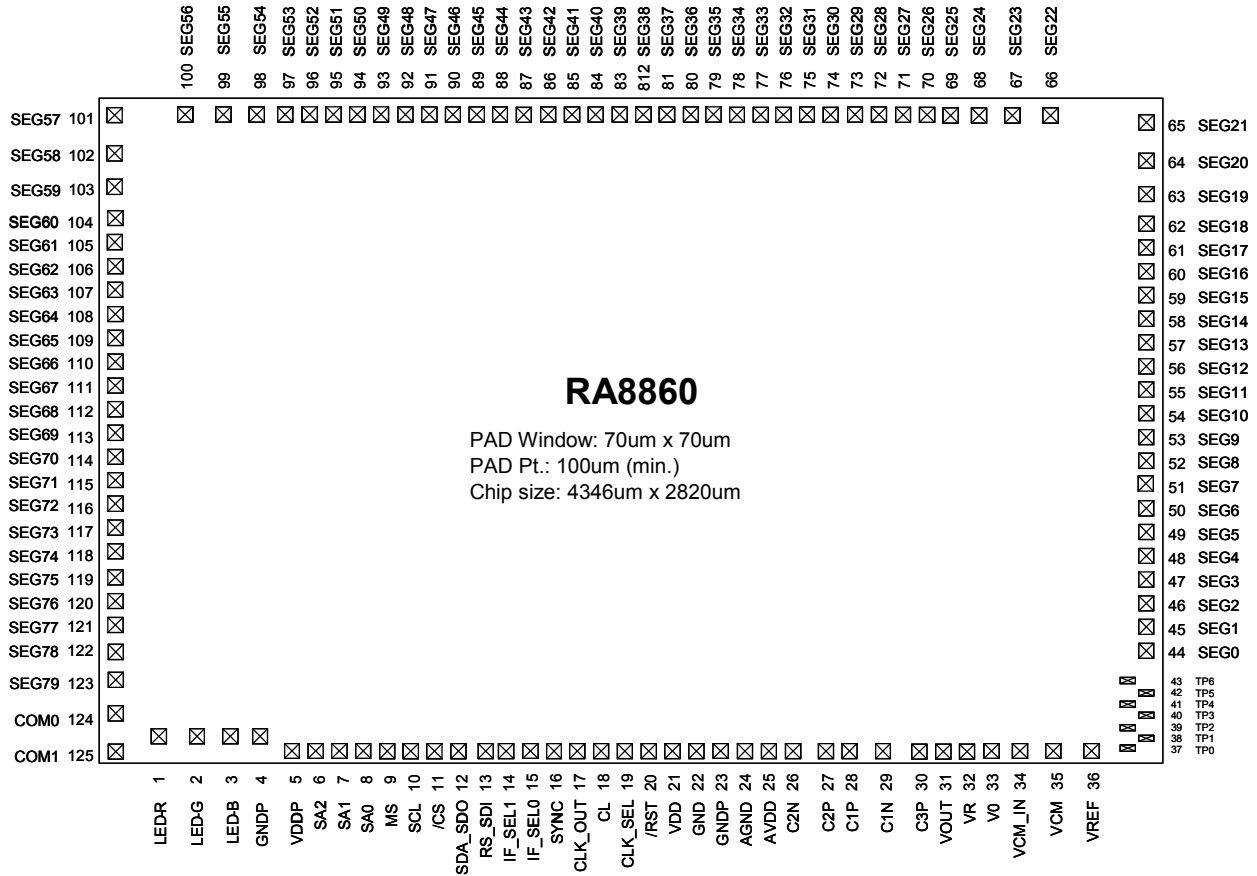


圖 3-2 : RA8860 腳位圖

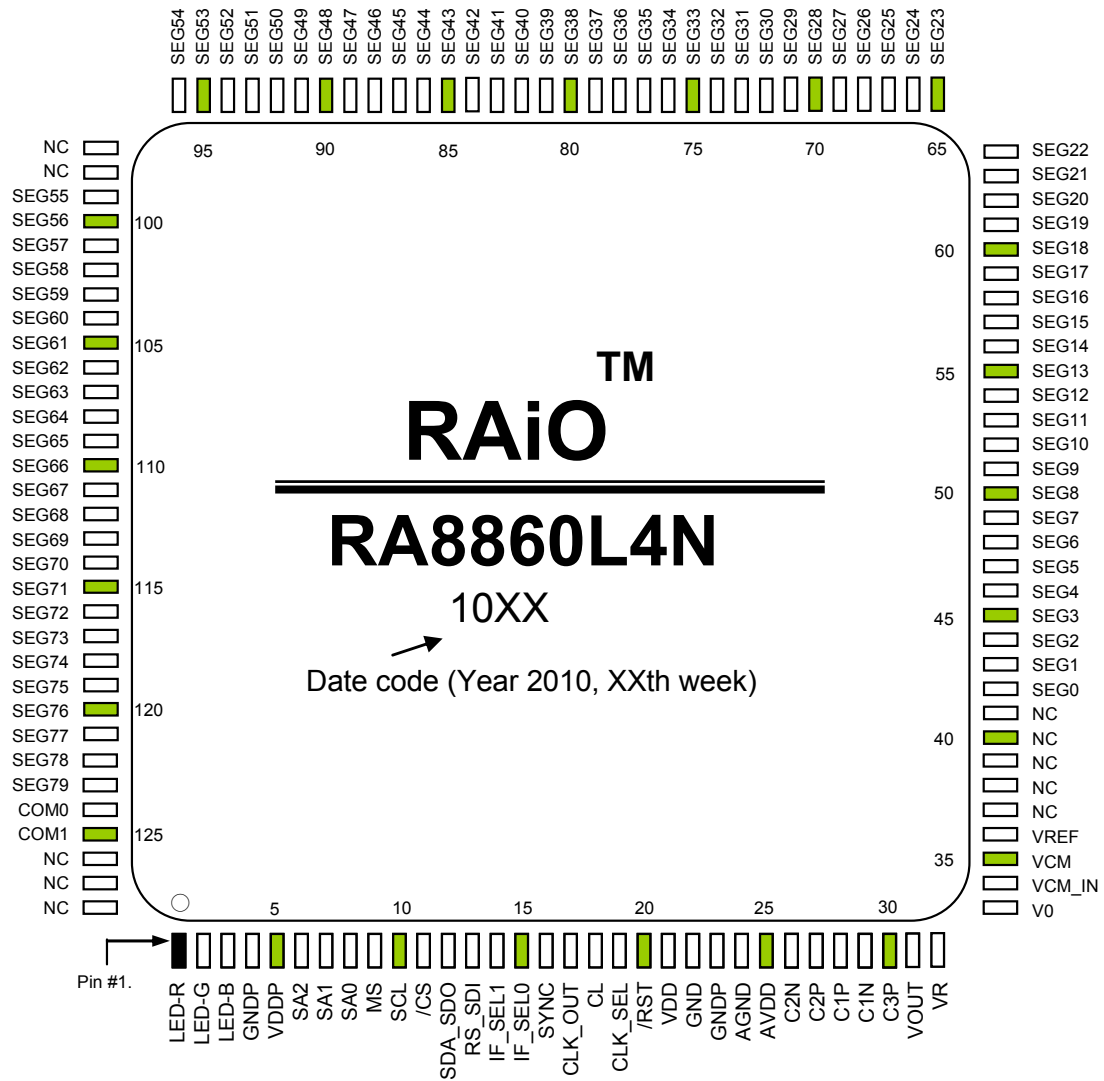


圖 3-3 : RA8860 LQFP-128 Pin Assignment

表 3-1 : 產品編號

產品編號	封裝
RA8860	Bare Die
RA8860L4N	LQFP-128pin

3-3 腳位座標

Pad No.	腳位名稱	X 座標	Y 座標
1	LED-R	-1836.585	-1210.29
2	LED-G	-1686.585	-1210.29
3	LED-B	-1556.145	-1210.29
4	GNDP	-1440.925	-1210.29
5	VDDP	-1250.925	-1274.97
6	SA2	-1150.925	-1274.97
7	SA1	-1050.925	-1274.97
8	SA0	-950.925	-1274.97
9	MS	-850.925	-1274.97
10	SCL	-750.925	-1274.97
11	/CS	-650.925	-1274.97
12	SDA SDO	-550.925	-1274.97
13	RS SDI	-450.925	-1274.97
14	IF_SEL1	-350.925	-1274.97
15	IF_SEL0	-250.925	-1274.97
16	SYNC	-150.925	-1274.97
17	CLK_OUT	-50.925	-1274.97
18	CL	49.075	-1274.97
19	CLK_SEL	149.075	-1274.97
20	/RST	249.075	-1274.97
21	VDD	349.075	-1274.97
22	GND	449.075	-1274.97
23	GNDP	549.075	-1274.97
24	AGND	649.075	-1274.97
25	AVDD	749.075	-1274.97
26	C2N	849.075	-1274.97
27	C2P	970.315	-1260.87
28	C1P	1070.315	-1260.87
29	C1N	1191.555	-1267.97
30	C3P	1312.795	-1260.87
31	VOOUT	1412.795	-1260.87
32	VR	1512.795	-1260.87
33	V0	1612.795	-1260.87
34	VCM_IN	1722.795	-1260.87
35	VCM	1847.795	-1260.87
36	VREF	1997.795	-1274.97
37	TP0	2099.7	-1226.8
38	TP1	2133.0	-1184.0
39	TP2	2099.7	-1141.2
40	TP3	2133.0	-1098.4
41	TP4	2099.7	-1055.6
42	TP5	2133.0	-1012.8
43	TP6	2099.7	-970.0
44	SEG0	2113.0	-900.0

Pad No.	腳位名稱	X 座標	Y 座標
45	SEG1	2113.0	-800.0
46	SEG2	2113.0	-700.0
47	SEG3	2113.0	-600.0
48	SEG4	2113.0	-500.0
49	SEG5	2113.0	-400.0
50	SEG6	2113.0	-300.0
51	SEG7	2113.0	-200.0
52	SEG8	2113.0	-100.0
53	SEG9	2113.0	0.0
54	SEG10	2113.0	100.0
55	SEG11	2113.0	200.0
56	SEG12	2113.0	300.0
57	SEG13	2113.0	400.0
58	SEG14	2113.0	500.0
59	SEG15	2113.0	600.0
60	SEG16	2113.0	700.0
61	SEG17	2113.0	800.0
62	SEG18	2113.0	900.0
63	SEG19	2113.0	1010.0
64	SEG20	2113.0	1135.0
65	SEG21	2113.0	1285.0
66	SEG22	1785.0	1350.0
67	SEG23	1635.0	1350.0
68	SEG24	1510.0	1350.0
69	SEG25	1400.0	1350.0
70	SEG26	1300.0	1350.0
71	SEG27	1200.0	1350.0
72	SEG28	1100.0	1350.0
73	SEG29	1000.0	1350.0
74	SEG30	900.0	1350.0
75	SEG31	800.0	1350.0
76	SEG32	700.0	1350.0
77	SEG33	600.0	1350.0
78	SEG34	500.0	1350.0
79	SEG35	400.0	1350.0
80	SEG36	300.0	1350.0
81	SEG37	200.0	1350.0
82	SEG38	100.0	1350.0
83	SEG39	0.0	1350.0
84	SEG40	-100.0	1350.0
85	SEG41	-200.0	1350.0
86	SEG42	-300.0	1350.0
87	SEG43	-400.0	1350.0
88	SEG44	-500.0	1350.0

Pad No.	腳位名稱	X 座標	Y 座標
89	SEG45	-600.0	1350.0
90	SEG46	-700.0	1350.0
91	SEG47	-800.0	1350.0
92	SEG48	-900.0	1350.0
93	SEG49	-1000.0	1350.0
94	SEG50	-1100.0	1350.0
95	SEG51	-1200.0	1350.0
96	SEG52	-1300.0	1350.0
97	SEG53	-1400.0	1350.0
98	SEG54	-1510.0	1350.0
99	SEG55	-1635.0	1350.0
100	SEG56	-1785.0	1350.0
101	SEG57	-2113.0	1285.0
102	SEG58	-2113.0	1135.0
103	SEG59	-2113.0	1010.0
104	SEG60	-2113.0	900.0
105	SEG61	-2113.0	800.0
106	SEG62	-2113.0	700.0
107	SEG63	-2113.0	600.0

Pad No.	腳位名稱	X 座標	Y 座標
108	SEG64	-2113.0	500.0
109	SEG65	-2113.0	400.0
110	SEG66	-2113.0	300.0
111	SEG67	-2113.0	200.0
112	SEG68	-2113.0	100.0
113	SEG69	-2113.0	0.0
114	SEG70	-2113.0	-100.0
115	SEG71	-2113.0	-200.0
116	SEG72	-2113.0	-300.0
117	SEG73	-2113.0	-400.0
118	SEG74	-2113.0	-500.0
119	SEG75	-2113.0	-600.0
120	SEG76	-2113.0	-700.0
121	SEG77	-2113.0	-800.0
122	SEG78	-2113.0	-900.0
123	SEG79	-2113.0	-1010.0
124	COM0	-2113.0	-1135.0
125	COM1	-2113.0	-1285.0

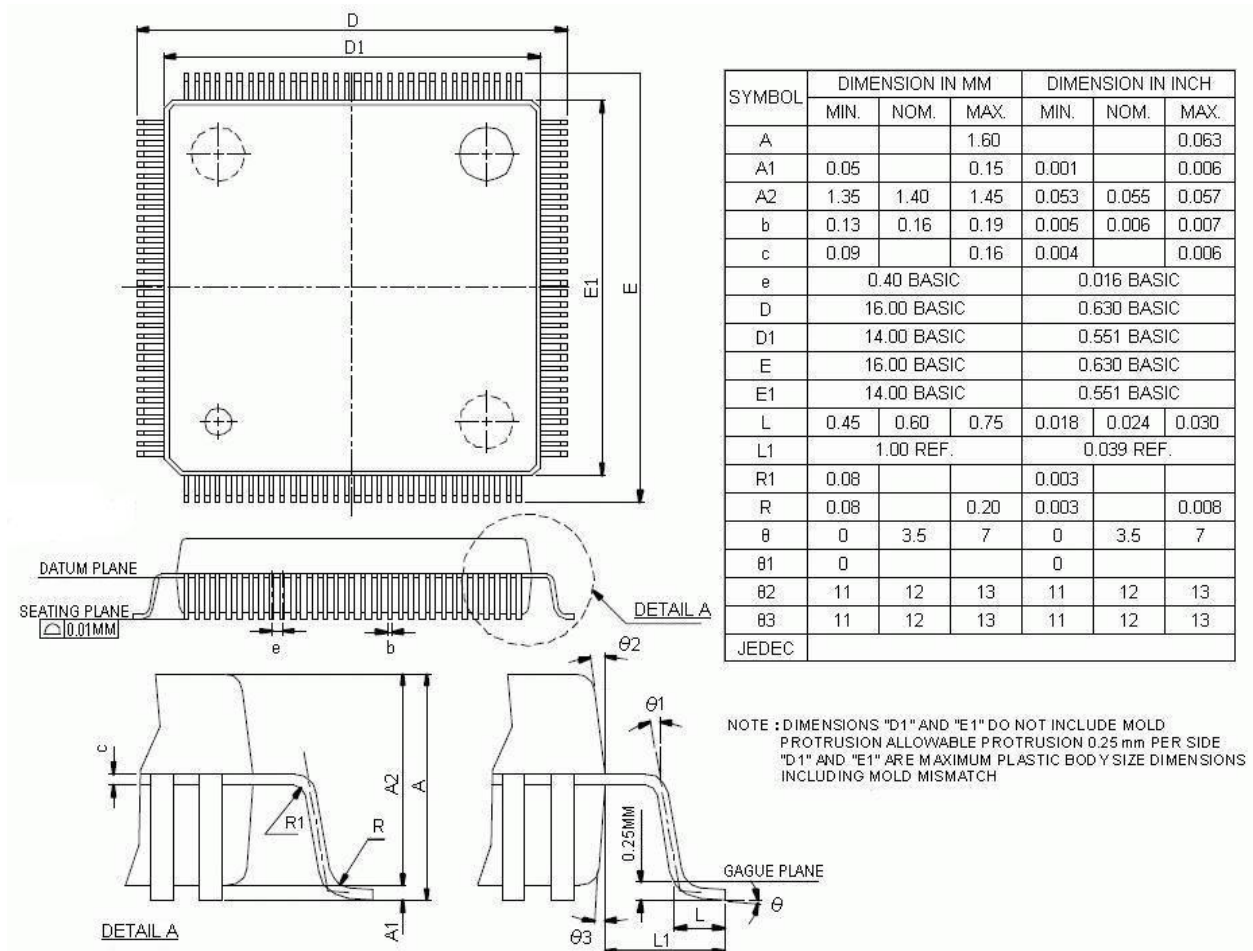


圖 3-4 : RA8860 Outline Dimensions

4. 腳位定義

腳位名稱	I/O	說明
LCD 驅動輸出		
SEG[79:0]	O	LCD Segment 驅動輸出。
COM[1:0]	O	LCD Common 驅動輸出。
LED 驅動輸出		
LED-R	O	Red LED 驅動輸出。
LED-G	O	Green LED 驅動輸出。
LED-B	O	Blue LED 驅動輸出。
MCU 界面		
/RST	I	重置輸入 (Reset)。
IF_SEL[1:0]	I	串聯 MCU 界面選擇。 00 : I ² C 模式。 01 : 3 線 SPI。 10 : 4 線 SPI Type-A。 11 : 4 線 SPI Type-B。
SCL, /CS	I	串聯數據界面。 在串聯模式，相關連接信號定義如下：
SDA_SDO	I/O	SCL : 串聯時脈 (Clock)。
RS_SDI	I	CS : 晶片選擇。 SDA : 雙向串聯 Data。 SDO : 串聯 Data 輸出。 RS : 記憶體/暫存器 週期選擇。 SDI : 串聯 Data 輸入。
CLK_SEL	I	時脈 (Clock) 選擇輸入 0 : 時脈由內部 RC 振盪產生。 1 : 時脈由外部 “CL” 接腳輸入。
CL	I	外部時脈 (Clock) 輸入。 當 CLK_SEL=1 或 Slave 模式時，此腳為外部時脈 (Clock) 輸入。 如未使用，請連接到 VDDP 或 GND。
MS	I	串接模式的主從設定 (Master / Slave)。 0 : 此 RA8860 設定為 Slave 模式。 1 : 此 RA8860 設定為 Master 模式。
SA[2:0]	I	I ² C 界面的元件選擇設定 (Device Address)。 如不使用必須連接到 VDDP 或 GND。
SYNC	I/O	用於串接模式的同步信號。 當 MS=1，此為同步信號輸出。 當 MS=0，此為同步信號輸入。

腳位名稱	I/O	說明
CLK_OUT	O	系統時脈 (Clock) 輸出。
電源界面		
VDD GND	P	內部核心電路的電源。
AVDD AGND	P	LCD 相關的類比電路電源。
VDDP GNDP	P	I/O 電源。
C[3:1]P C[2:1]N	O	倍壓電源的外部電容接腳。
VOUT	P	倍壓電源輸出。
VREF	P	V0 的參考電壓。
VR	I	外部的分壓電路輸入。
V0 VCM_IN VCM	P	LCD 驅動電壓。
TP[6:0]	I	測試輸入。 這些腳位為測試訊號輸入，必須保持浮接 (NC)。

5. 電氣特性

5-1 最大範圍限制

表 5-1

Characteristic	Symbol	Value	Unit	Note
工作電壓	VDD AVDD VDDP	-0.3 - +7.0	V	--
LCD 驅動電壓	V0 VOUT	-0.3 – 15.0	V	--
工作溫度	VOPR	-35 - +90	°C	--
儲存溫度	VSTG	-55 - +125	°C	--

表 5-1 的數據表示在超過 "最大範圍限制" 的環境下，可能對RA8860 造成某種固定程度上的傷害。而長時間接觸 "可能最大範圍"，可能將影響RA8860 的可靠度或某局部性的功能損壞。

5-2 DC 電氣特性

表 5-2

(VDD = +5V / +3.3V ± 10% , VSS = 0V , Ta = -30 to +80°C)

Characteristic	Symbol	Condition	Min	Typ	Max	Unit	Note
Input high voltage	V _{IH1}	—	0.55V _{DDP}	—	V _{DDP}	V	(1)
	V _{IH2}	—	0.8V _{DDP}	—	V _{DDP}	V	(2)
Input low voltage	V _{IL1}	—	0	—	0.3V _{DDP}	V	(1)
	V _{IL2}	—	0	—	0.2V _{DDP}	V	(2)
Output high voltage	V _{OH}	I _{OH} / I _{OL} = 8mA	0.8V _{DDP}	—	—	V	(3)
Output low voltage	V _{OL1}	I _{OH} / I _{OL} = 8mA	—	—	0.2V _{DDP}	V	(3)
	V _{OL2}	I _{OL} = 80mA	—	—	0.2V _{DDP}	V	(4)
Input leakage current	I _{LKG}	V _{IN} = V _{SS} ~ V _{DDP}	-1.0	—	1.0	μA	(1, 2)
Operating current	I _{DD1}	Display On	—	—	950	μA	(5)
	I _{DD2}	Display Off	—	—	450	μA	(5)

註：

1. 此條件適用於輸入腳 /RST , SCL , /CS , RS_SDI , SDA_SDO 和雙向腳位 SYNC 。
2. 此條件適用於輸入腳 IF_SEL[1:0] , CLK_SEL , CL , MS 和 SA[2:0] 。
3. 此條件適用於雙向腳位 SDA_SDO , SYNC 和輸出腳位 CLK_OUT 。
4. 此條件適用於 LED 驅動腳位 LED-R , LED-G , LED-B 。
5. 此條件為當 Frame 頻率 = 65Hz , LED 輸出 (LED-R , LED-G , LED-B) 無負載時。

5-3 AC 電氣特性

表 5-3 : I²C 界面

Item	Signal	Symbol	Rating		Unit
			Min.	Max.	
SCL Clock Frequency	SCL	f_{SCL}	100	400	KHz
Bus Free Time Between STOP and START	SCL/SDA_SDO	t_{BUF}	1	--	μs
Low Period of SCL Clock	SCL	t_{LOW}	200	--	ns
High Period of SCL Clock	SCL	t_{HIGH}	200	--	ns
Data Setup Time	SCL/SDA_SDO	t_{DSIC}	100	--	ns
Data Hold Time	SCL/SDA_SDO	t_{DHIC}	100	--	ns

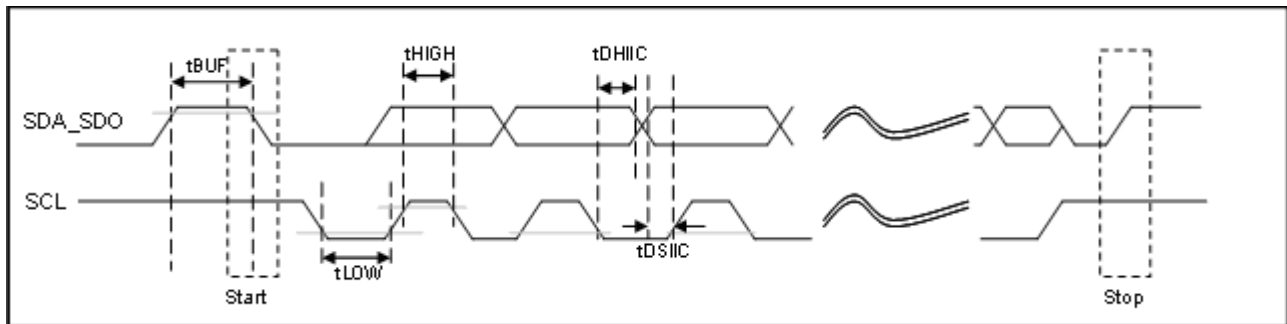


圖 5-1 : I²C 時序

表 5-4 : 3 線 SPI 界面

Item	Signal	Symbol	Rating		Unit
			Min.	Max.	
Access Time	/CS	t_{CYC3}	200	--	ns
/CS Setup Time		t_{CSH3}	20	--	
Clock Low Pulse Width	SCL	t_{CKL3}	100	--	
Clock High Pulse Width		t_{CKH3}	100	--	
Data Setup Time	SDA_SDO	t_{DS3}	20	--	
Data Hold Time		t_{DH3}	10	--	

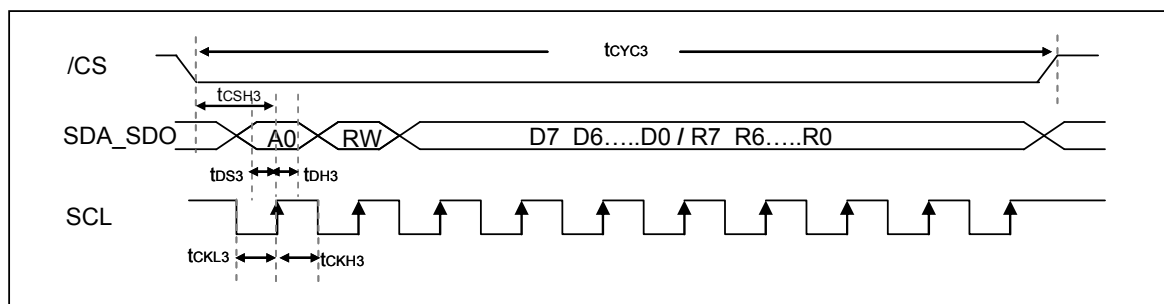


圖 5-2 : 3 線 SPI 時序

表 5-5 : 4 線 SPI (Type-A) 界面

Item	Signal	Symbol	Condition	Rating		Unit
				Min.	Max.	
Access Time	$\overline{\text{CS}}$	t_{CYC4A}	--	200	--	ns
$\overline{\text{CS}}$ Setup Time		t_{CSH4}	--	20	--	
Clock Low Pulse Width	SCK	t_{CKL4}	--	100	--	
Clock High Pulse Width		t_{CKH4}	--	100	--	
Data Setup Time	SDA , RS	t_{DS4}	--	20	--	
Data Hold Time		t_{DH4}	--	10	--	

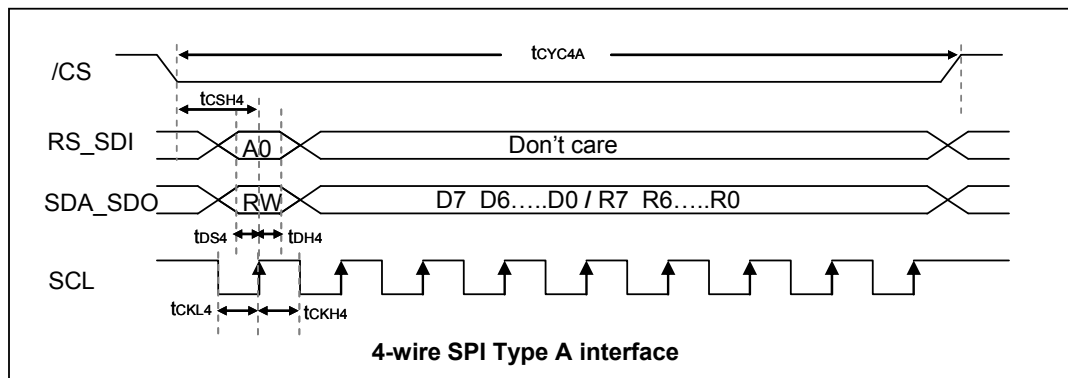


圖 5-3 : 4 線 SPI (Type-A) 時序

表 5-6 : 4 線 SPI (Type-B) 界面

Item	Signal	Symbol	Condition	Rating		Unit
				Min.	Max.	
Access Time	$\overline{\text{CS}}$	t_{CYC4B}	--	200	--	ns
$\overline{\text{CS}}$ Setup Time		t_{CSH4}	--	20	--	
Clock Low Pulse Width	SCK	t_{CKL4}	--	100	--	
Clock High Pulse Width		t_{CKH4}	--	100	--	
Data Write Setup Time	SDI	t_{DS4}	--	20	--	
Data Write Hold Time		t_{DH4}	--	10	--	
Data Read Setup Time	SDO	t_{OS4B}	--	20	--	
Data Read Hold Time		t_{OH4B}	--	10	--	

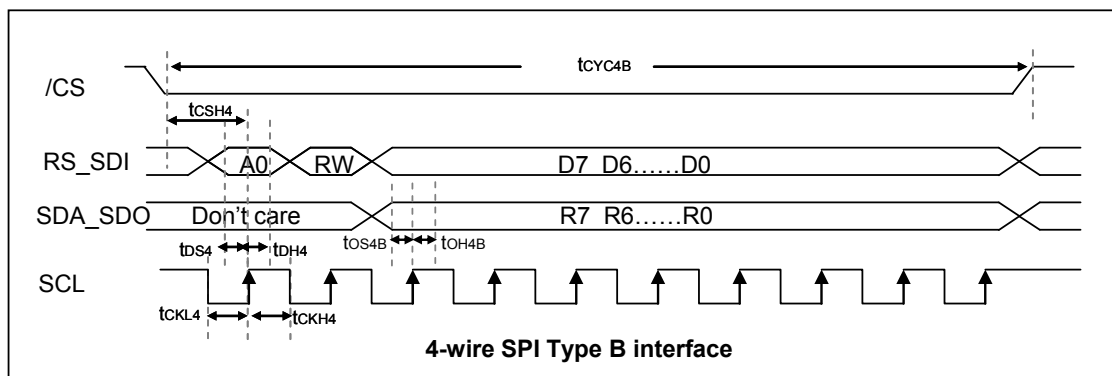


圖 5-4 : 4 線 SPI (Type-B) 時序

6. 暫存器與指令

表 6-1：暫存器表

Reg No#	COMMAND	CODE									
		HEX	A0	D7	D6	D5	D4	D3	D2	D1	D0
00	NOP	00	0	0	0	0	0	0	0	0	0
01	Software Reset	01	0	0	0	0	0	0	0	0	1
10	Sleep In	10	0	0	0	0	1	0	0	0	0
11	Sleep Out	11	0	0	0	0	1	0	0	0	1
22	Exit all Point On	22	0	0	0	1	0	0	0	1	0
23	Enter all Point On	23	0	0	0	1	0	0	0	1	1
24	Blinking Setting	24	0	0	0	1	0	0	1	0	0
		-	1	0	0	0	0	0	0	0	EN
25	Blinking Address	25	0	0	0	1	0	0	1	0	1
		-	1	RBA0	CBA6	CBA5	CBA4	CBA3	CBA2	CBA1	CBA0
26	Blinking Interval	26	0	0	0	1	0	0	1	1	0
		-	1	0	0	1	0	0	BLK2	BLK1	BLK0
28	Display On	28	0	0	0	1	0	1	0	0	0
29	Display Off	29	0	0	0	1	0	1	0	0	1
2A	Set Column	2A	0	0	0	1	0	1	0	1	0
		-	1	0	0	C5	C4	C3	C2	C1	C0
2B	Set Duty	2B	0	0	0	1	0	1	0	1	1
		-	1	0	0	0	0	0	0	0	DT
2C	Memory Write	2C	0	0	0	1	0	1	1	0	0
2E	Memory Read	2E	0	0	0	1	0	1	1	1	0
2F	Address Point	2F	0	0	0	1	0	1	1	1	1
		-	1	0	RA0	CA5	CA4	CA3	CA2	CA1	CA0
A1	Red LED Start	A1	0	1	0	1	0	0	0	0	1
		-	1	LRS7	LRS6	LRS5	LRS4	LRS3	LRS2	LRS1	LRS0
A2	Green LED Start	A2	0	1	0	1	0	0	0	1	0
		-	1	LGS7	LGS6	LGS5	LGS4	LGS3	LGS2	LGS1	LGS0
A3	Blue LED Start	A3	0	1	0	1	0	0	0	1	1
		-	1	LBS7	LBS6	LBS5	LBS4	LBS3	LBS2	LBS1	LBS0
A4	Red LED Width	A4	0	1	0	1	0	0	1	0	0
		-	1	LRW7	LRW6	LRW5	LRW4	LRW3	LRW2	LRW1	LRW0
A5	Green LED Width	A5	0	1	0	1	0	0	1	0	1
		-	1	LGW7	LGW6	LGW5	LGW4	LGW3	LGW2	LGW1	LGW0
A6	Blue LED Width	A6	0	1	0	1	0	0	1	1	0
		-	1	LBW7	LBW6	LBW5	LBW4	LBW3	LBW2	LBW1	LBW0

表 6-1：暫存器表 (接續上頁)

Reg No#	COMMAND	CODE									
		HEX	A0	D7	D6	D5	D4	D3	D2	D1	D0
B1	LED Mode	B1	0	1	0	1	1	0	0	0	1
		-	1	1	0	0	0	0	0	0	LEDP
B2	Frame Frequency	B2	0	1	0	1	1	0	0	1	0
		-	1	0	0	0	0	0	FR2	FR1	FR0
B7	LCD Scan Set	B7	0	1	0	1	1	0	1	1	1
		-	1	MY	0	0	0	0	0	0	0
B8	Enter Read Modify	B8	0	1	0	1	1	1	0	0	0
B9	Exit Read Modify	B9	0	1	0	1	1	1	0	0	1
BA	16 Color Set	BA	0	1	0	1	1	1	0	1	0
BB	8 Color Set	BB	0	1	0	1	1	1	0	1	1
C0	Vop Set	C0	0	1	1	0	0	0	0	0	0
		-	1	0	0	Vop5	Vop4	Vop3	Vop2	Vop1	Vop0
D2	PWR Control	D2	0	1	1	0	1	0	0	1	0
		-	1	0	0	CK_OEN	IRS	BST	FOL	V0	VREF
D4	RGB LED Control	D4	0	1	1	0	1	0	1	0	0
		-	1	0	0	0	0	BK	LEDR	LEDG	LEDB

[00h] NOP

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0

此為一無動作指令。

[01h] Software Reset

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	1

此為軟體重置指令，執行後暫存器除了 REG [B2h] 之外，其它都會變成初始值。

[10h / 11h] Sleep Mode

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	0	0	0	SLEEP

這個指令用來進入睡眠模式 (Sleep)，會讓 RA8860 的 RC 振盪器、LCD 驅動與 LED 驅動等電路，同時進入 Display Off 模式，LCD 驅動信號會輸出低準位，當由睡眠模式回到正常模式時需要一小段時間，以逐步開起 RC 振盪器、倍壓電路、LCD 驅動電路等等，才能回到正常顯示畫面。注意，進入睡眠模式時倍壓電路不會自動關閉，使用者需要先自行關閉。

SLEEP : 0 → 進入睡眠模式；1 → 正常模式 (初始值)。

[22h / 23h] All Points On/Off Mode

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	0	0	1	ALL

這個指令用來進入全部 Pixel 開啓 (Turn On) 的顯示模式，不過不會改變顯示記憶體的內容。

ALL : 0 → 離開全部顯示模式 (初始值)；1 → 進入全部顯示模式。

[24h] Blinking Setting

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	0	1	0	0
1	0	0	0	0	0	0	0	EN

EN : 1 → 閃爍模式開啓；0 → 閃爍模式關閉 (初始值)。

[25h] Blinking Address

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	0	1	0	1
1	RBA0	CBA6	CBA5	CBA4	CBA3	CBA2	CBA1	CBA0

此暫存器用來設定要閃爍的點 (Pixel)。

RBA0：設定閃爍的 Common 地址。(初始值 = 0)

CBA[6:0]：設定閃爍的 Segment 地址。(初始值 = 00h)

[26h] Blinking Interval

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	0	1	1	0
1	0	0	0	0	0	BLK2	BLK1	BLK0

此暫存器用來設定閃爍的間隔時間，請參考表 6-2 (初始值 = 00h)，表 6-2 是假設Frame Rate為 75Hz的條件下所得到的閃爍間隔時間。

表 6-2：閃爍的間隔時間

BLK[3:0]			閃爍的間隔時間 (Frame)	Frame Rate = 75Hz (FR[2:0]= 001)
0	0	0	4* 1/Frame	~50ms
0	0	1	8* 1/Frame	~100ms
0	1	0	16* 1/Frame	~200ms
0	1	1	32* 1/Frame	~430ms
1	0	0	64* 1/Frame	~850ms
1	0	1	128* 1/Frame	~1.7Sec
1	1	0	256* 1/Frame	~3.4Sec
1	1	1	512* 1/Frame	~6.8Sec

[28h / 29h] Display Off Mode

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	1	0	0	DOFF

此指令用來設定 RA8860 是否進入顯示關閉 (Display Off) 狀態，當進入顯示關閉 LCD 驅動輸出會拉到 GND 準位，且 LED 驅動信號也進入 Tri-state 狀態。

DOFF：1 → 顯示關閉 (初始值)；0 → 顯示開啓。

[2Ah] Set Column

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	1	0	1	0
1	0	0	C5	C4	C3	C2	C1	C0

此指令用來設定 LCD 面板最大的 Segment 數，設定範圍由 0h 到 27h，初始值為 27h。

[2Bh] Set Duty

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	1	0	1	1
1	0	0	0	0	0	0	0	DT

此指令用來設定 LCD 面板的 Duty。

DT : 0 → Static ; 1 → 1/2 Duty。(初始值)

[2Ch / 2Eh] Memory Write/Read

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	1	1	W/R	0

此指令用來設定是由記憶體讀取資料或是寫入資料到記憶體。

W/R : 0 → 寫入資料 ; 1 → 讀取資料。

[2Fh] Address Point

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	1	1	1	1
1	0	RA0	CA5	CA4	CA3	CA2	CA1	CA0

此暫存器用來設定要點亮的 Pixel。(初始值 = 0)

RBA0 : 設定要點亮的 Pixel 的 Common 地址。

CBA[6:0] : 設定要點亮的 Pixel 的 Segment 地址。

[A1h] Red LED Start Waveform Position Setting

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	0	0	0	1
1	LRS7	LRS6	LRS5	LRS4	LRS3	LRS2	LRS1	LRS0

紅色LED驅動信號 (LED-R) 動作時的起始位置，初始值 = 00h，請參考第 7-9 節與 圖 7-25 的說明。

[A2h] Green LED Start Waveform Position Setting

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	0	0	1	0
1	LGS7	LGS6	LGS5	LGS4	LGS3	LGS2	LGS1	LGS0

綠色 LED 驅動信號 (LED-G) 動作時的起始位置。(初始值 = 00h)

[A3h] Blue LED Start Waveform Position Setting

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	0	0	1	1
1	LBS7	LBS6	LBS5	LBS4	LBS3	LBS2	LBS1	LBS0

藍色 LED 驅動信號 (LED-B) 動作時的起始位置。(初始值 = 00h)

[A4h] Red LED Width Waveform Setting

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	0	1	0	0
1	LRW7	LRW6	LRW5	LRW4	LRW3	LRW2	LRW1	LRW0

紅色 LED 驅動信號 (LED-R) 動作時的寬度設定 (Active Time)，初始值 = FFh，請參考第 7-9 節與圖 7-25 的說明。

[A5h] Green LED Width Waveform Setting

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	0	1	0	1
1	LGW7	LGW6	LGW5	LGW4	LGW3	LGW2	LGW1	LGW0

綠色 LED 驅動信號 (LED-G) 動作時的寬度設定 (Active Time)。(初始值 = FFh)

[A6h] Blue LED Width Waveform Setting

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	0	0	1	1	0
1	LBW7	LBW6	LBW5	LBW4	LBW3	LBW2	LBW1	LBW0

藍色 LED 驅動信號 (LED-B) 動作時的寬度設定 (Active Time)。(初始值 = FFh)

[B1h] LED Mode

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	1	0	0	0	1
1	0	0	0	0	0	0	0	LEDP

此指令用來設定 LED 驅動信號的動作準位。

LEDP : 0 → LED-R、LED-G 和 LED-B 信號是 Low 動作。(初始值)

1 → LED-R、LED-G 和 LED-B 信號是 High 動作。

[B2h] Frame Frequency

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	1	0	0	1	0
1	0	0	0	0	0	FR2	FR1	FR0

此指令用來設定 Frame Rate 的頻率。(初始值 = 00h)

表 6-3 : Frame Rate 設定

FR[2:0]			Frame Rate (VDD=5V)
0	0	0	65Hz (初始值)
0	0	1	75Hz
0	1	0	85Hz
0	1	1	95Hz
1	0	0	105Hz
1	0	1	115Hz
1	1	0	125Hz
1	1	1	135Hz

[B7h] LCD Scan Set

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	1	0	1	1	1
1	MY	0	0	0	0	0	0	0

此指令用來設定 LCD 掃描的方向。

MY : 0 → 掃描的方向由 COM0 到 COM1 (初始值) ; 1 → 由 COM1 到 COM0。

[B8h / B9h] Read Modify Write Select

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	1	1	0	0	RMW

此指令用來決定記憶體讀取時 (0x2E)，位址是否要自動累加。在沒有進入此模式下時，當執行每筆記憶體讀取完後，內部位址位自動加一。

但若進入此模式時，執行每筆記憶體讀取完後，內部位址則不會自動加一。

RMW: 0 → 記憶體讀取時，位址不自動加一模式。

1 → 記憶體讀取時，位址自動加一模式。(初始值)

[BAh / BBh] Color Set

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	0	1	1	1	0	1	COLOR

此指令用來設定是 16 色或 8 色模式。

COLOR : 0 → 16 色模式 (此功能不支援 Normally Black 的 panel) ; 1 → 8 色模式 (初始值)。

[C0h] Vop Set

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	1	0	0	0	0	0	0
1	0	0	Vop5	Vop4	Vop3	Vop2	Vop1	Vop0

此指令用來設定LCD的V0 電壓，初始值 = 20h。請參考第 7-7 節說明。

[D2h] PWR Control

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	1	0	1	0	0	1	0
1	0	0	CK_OEN	IRS	BST	FOL	V0	VREF

此指令用來設定 RA8860 內部一些電路是開啓還是關閉狀態。(初始值 = 10h)

CK_OEN : 0 → 關閉 “CLK_OUT” 和 ”SYNC”，維持輸出 High ; 1 → 開啓 “CLK_OUT” 和 ”SYNC” 為正常輸出狀態。

IRS : 0 → 選擇外部 R-string 電路 ; 1 → 選擇內部 R-string 電路。

BST : 0 → 關閉倍壓電路 ; 1 → 開啓倍壓電路。

FOL : 0 → 關閉 LCD Bias 電路 ; 1 → 開啓 LCD Bias 電路。

V0 : 0 → 關閉電壓調整電路 (Regulator) ; 1 → 開啓電壓調整電路。

VREF : 0 → 選擇外部參考電壓 (Reference Voltage) ; 1 → 選擇內部參考電壓。

[D4h] RGB LED Control

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	1	1	0	1	0	1	0	0
1	0	0	0	0	BK	LEDR	LEDG	LEDB

此指令用來設定控制 LED 驅動信號 (LED-R、LED-G 和 LED-B) 全部關閉或各別開啓。

BK : 0 → 全部 LED 驅動關閉 (Turn Off)(初始值) ; 1 → RGB LED 驅動關閉或開啓由 LEDR、LEDG 和 LEDB bit 決定。

表 6-4

BK	LEDR	LEDG	LEDB	Description	
0	X	X	X	RGB LED Control off	
1	0	0	0	Backlight off	
1	0	0	1	Blue	
1	0	1	0	Green	
1	0	1	1	Cyan	
1	1	0	0	Red	
1	1	0	1	Pink	
1	1	1	0	Yellow	
1	1	1	1	White	

7. 功能描述

7-1 MCU 界面

RA8860 支援四種串聯界面，一種是 3 線 SPI，一種是 I²C 界面，另兩種是 4 線 SPI (Type-A，Type-B) 界面，四種串聯界面由輸入信號 IF_SEL[1:0] 來控制，請參考表 7-1。

表 7-1：MCU 界面的選擇

IF_SEL[1:0]	MCU 界面
00	I ² C
01	3 線 SPI
10	4 線 SPI，Type-A
11	4 線 SPI，Type-B

表 7-2：四種串聯界面所對應的接腳定義

Mode 信號	I ² C	3 線 SPI	4 線 SPI， Type-A	4 線 SPI， Type-B
SCL	串聯時脈	串聯時脈	串聯時脈	串聯時脈
/CS	註*	晶片選擇 (Chip Select)	晶片選擇	晶片選擇
SDA_SDO	Bi-direction Data	Bi-direction Data	Bi-direction Data	Data Out
RS_SDI	註*	註*	A0	Data In
SA[2:0]	元件選擇 (Device Address)	註*	註*	註*

註：請連接到 VDDP 或 GND。

7-1-1 I²C 界面

如果使用 I²C 界面，MCU 只需要 2 個信號與 RA8860 的 "SCL" 和 "SDA_SDO" 連接，"SDA_SDO" 的傳輸內容包括了 Device Address、A0、Read/Write State (RW)、ACK 和 Data，同時傳輸這些內容時必須與 "SCL" 同步，使用 I²C 界面時輸入信號 "SA[2:0]" 必須接到 High 或 Low，並且與 MCU 透過 "SDA_SDO" 傳輸內容的 Device Address 一致，這樣此 RA8860 才會被選到 (Active)，由此也可看出 RA8860 透過 SA[2:0] 可選擇 8 個不同的 Device Address。

例如下 圖 7-2 的 SA2 連接到 VDDP，SA1 與 SA0 連接到 GND，所以 Device Address SA[2:0] = 2'100b，因此完整的 I²C bus Device Address 是：

$$\begin{aligned} AD &= \{ 2'b10, SA[2:0], 1'b0 \} \\ &= 6'b101000 \end{aligned}$$

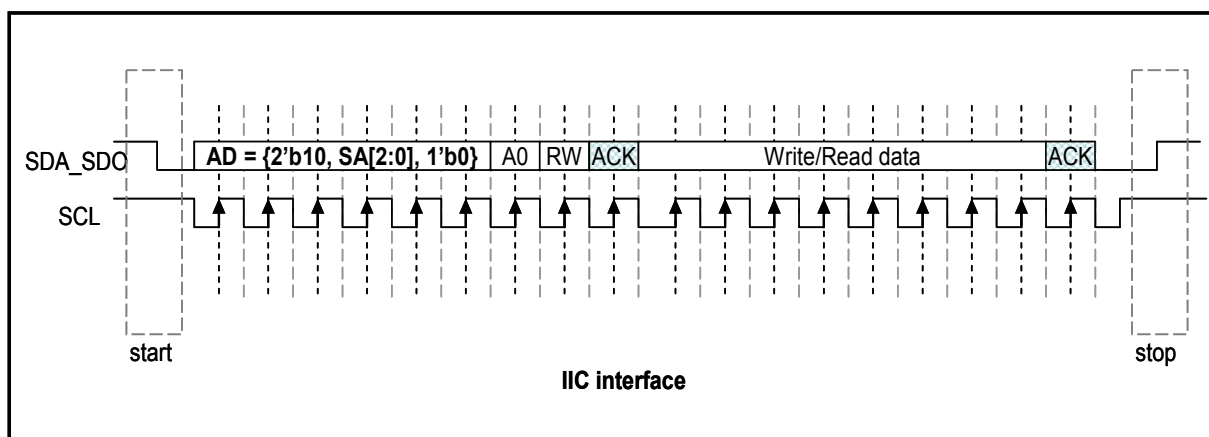


圖 7-1 : I²C 界面時序

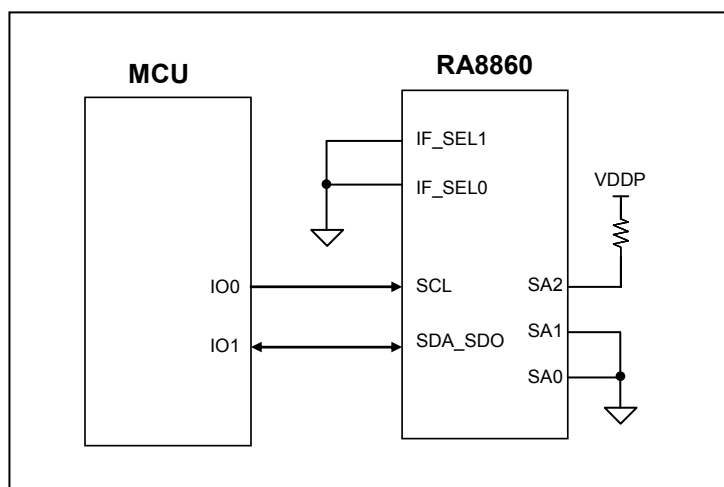


圖 7-2 : I²C 界面連接 MCU

7-1-2 3線SPI

在 3 線 SPI 界面，“SDA_SDO” 的傳輸內容包括了 A0、Read/Write State (RW) 和 Data，一樣在傳輸這些內容時必須與“SCL”同步，此界面的“SDA_SDO”是雙向傳輸的，對 RA8860 來說通常為輸入，但當 RW=1，也就是 MCU Read 時，這根腳變成 Data 輸出信號，讓 MCU 讀取數據。

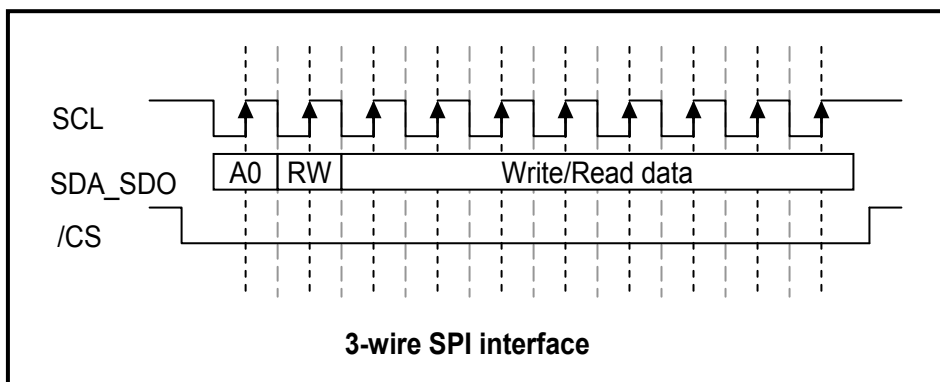


圖 7-3：3 線 SPI 時序

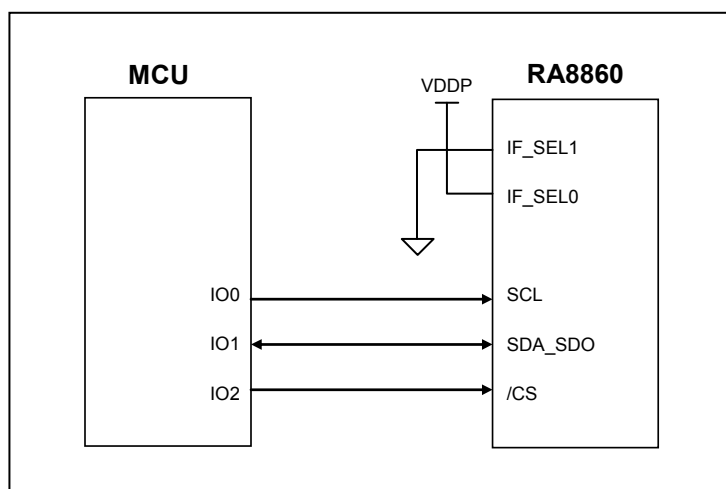


圖 7-4：3 線 SPI 界面連接 MCU

7-1-3 4線SPI (Type-A)

4 線 SPI (Type-A) 界面與 3 線 SPI 很相似，不同點是 A0 狀態由信號“RS_SDI”來表示，同樣“SDA_SDO”是雙向傳輸的腳位，對 RA8860 來說通常為輸入，但是當 RW=1，也就是 MCU Read 時，這根腳變成 Data 輸出信號，讓 MCU 讀取數據。

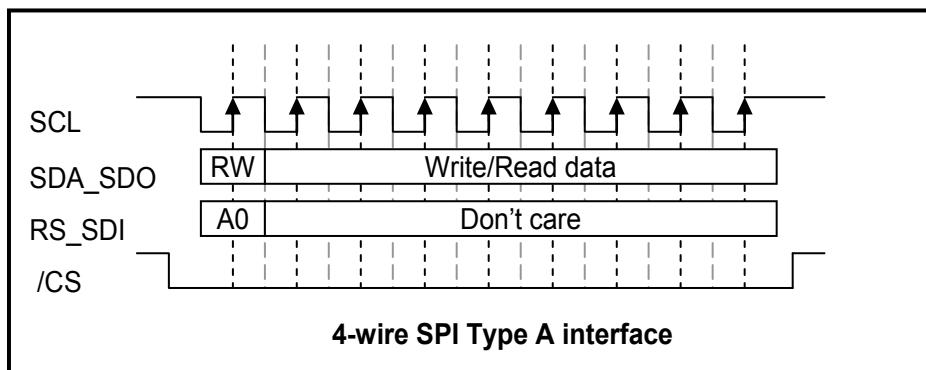


圖 7-5 : 4 線 SPI (Type-A) 時序

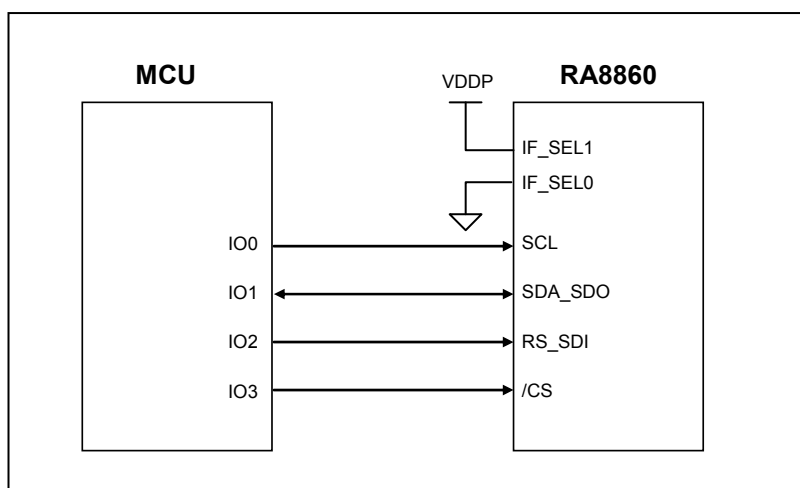


圖 7-6 : 4 線 SPI (Type-A) 界面連接 MCU

7-1-4 4線SPI (Type-B)

4 線 SPI (Type-B) 界面也和 3 線 SPI 很相似，“RS_SDI” 是輸入腳，MCU 透過它傳輸的內容包括了 A0、Read Write State (RW) 和 Write Data，同時傳輸這些內容時必須與“SCL” 同步，“SDA_SDO” 則專門當作 Data 輸出腳，當 RW=1，也就是 MCU Read 時，MCU 透過“SDA_SDO” 讀取 RA8860 的串聯數據。

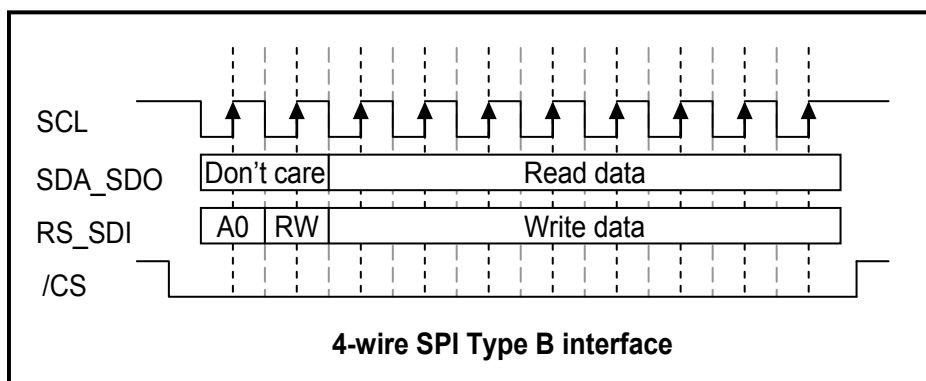


圖 7-7 : 4 線 SPI (Type-B) 時序

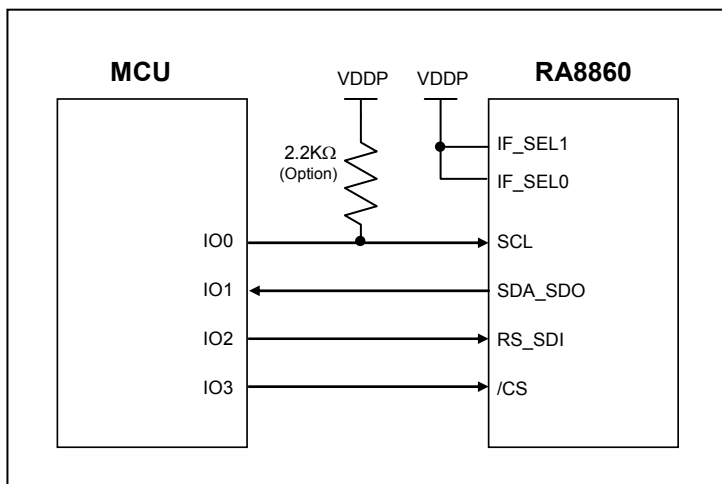
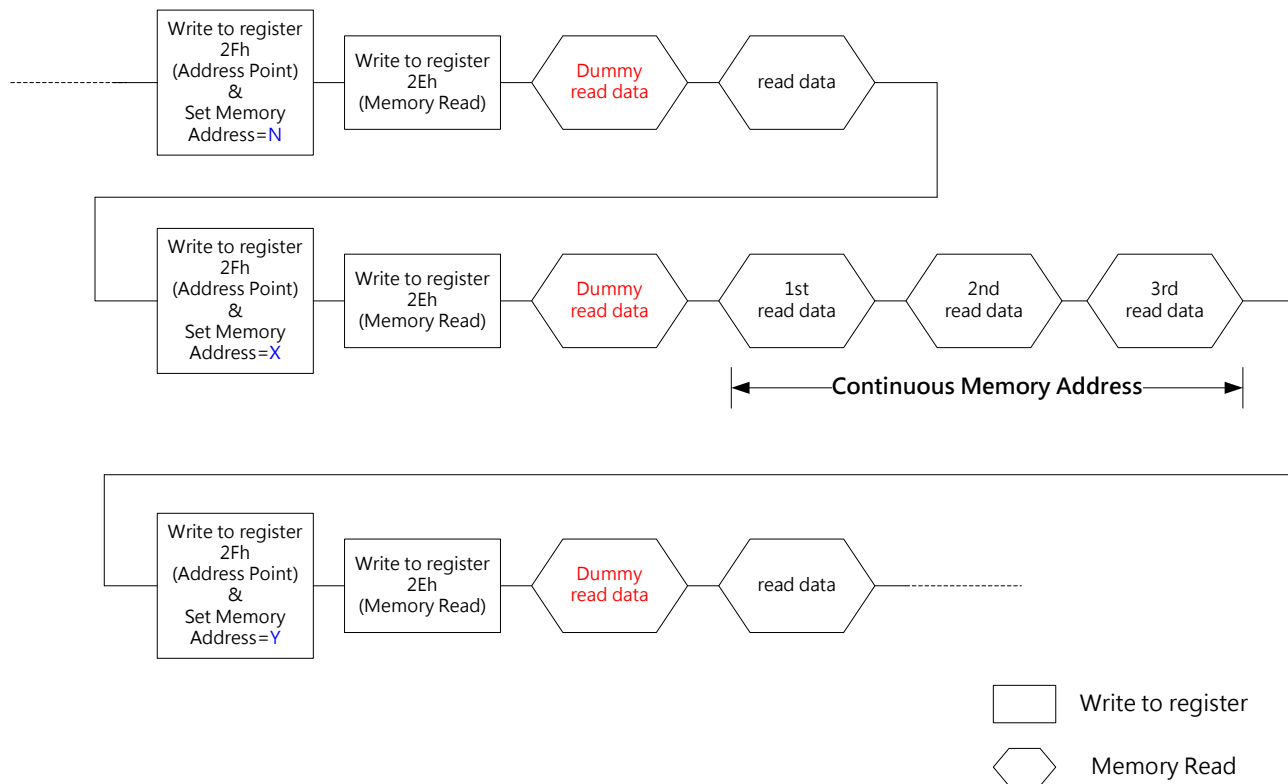


圖 7-8 : 4 線 SPI (Type-B) 界面連接 MCU

7-1-5 記憶體讀取



Example:

```

LCD_CmdWrite(0x2F);    //Set Memory Address=N
LCD_DataWrite(N);

LCD_CmdWrite(0x2E);    //Memory Read

Temp=LCD_DataRead();   //Dummy Read
Temp=LCD_DataRead();   //Data

LCD_CmdWrite(0x2F);    //Set Memory Address=X
LCD_DataWrite(X);

LCD_CmdWrite(0x2E);    //Memory Read

Temp=LCD_DataRead();   //Dummy Read
Temp=LCD_DataRead();   //1st Data
Temp=LCD_DataRead();   //2nd Data
Temp=LCD_DataRead();   //3rd Data

LCD_CmdWrite(0x2F);    //Set Memory Address=Y
LCD_DataWrite(Y);

LCD_CmdWrite(0x2E);    //Memory Read

Temp=LCD_DataRead();   //Dummy Read
Temp=LCD_DataRead();   //Data
  
```

圖 7-9：記憶體讀取

7-2 串接模式

RA8860 支援多顆的串接模式，當“MS”連接到 High，此 RA8860 則當成 Master，同時“SYNC”和“CLK_OUT”輸出信號到 Slave。當“MS”連接到 Low，此 RA8860 則當成 Slave，“CL”是時脈 (Clock) 輸入，接收由 Master “CLK_OUT”送出的時脈信號，而“SYNC”也變成輸入，接收由 Master “SYNC”送出的同步信號。RA8860 最多可串聯 8 個，也就是支援 80x2x8 個點的 LCD 面板，而且串接模式支援前面所提的 4 種 MCU 串聯界面。

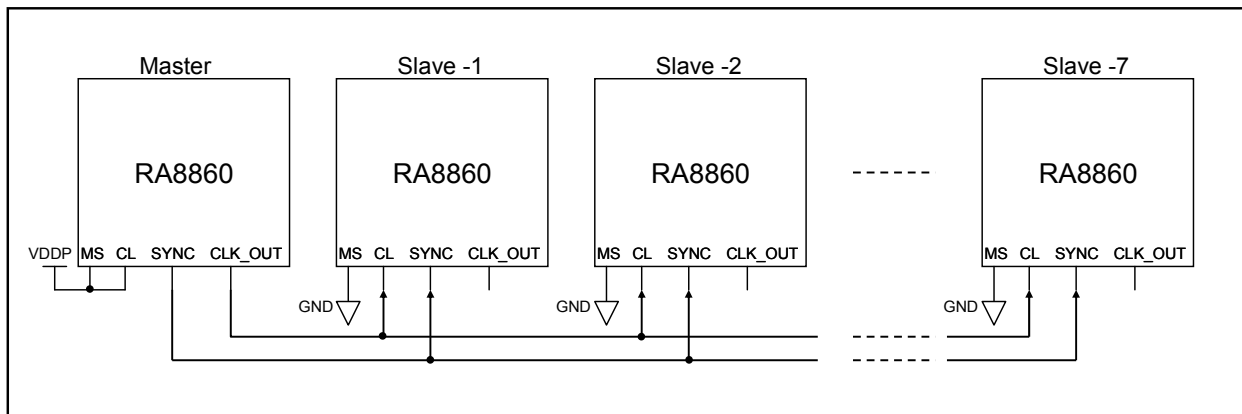


圖 7-10 : RA8860 的串接模式

圖 7-11 是一個 3 線 SPI 界面的串接模式，2 個 RA8860 的“SCL”、SDA_SDO 和“/RST”分別接在一起，並由 MCU 來控制，而“/CS”則各別由 MCU 單獨來控制。如果想要用 I²C 界面的串接模式，可參考圖 8-3 的應用電路。

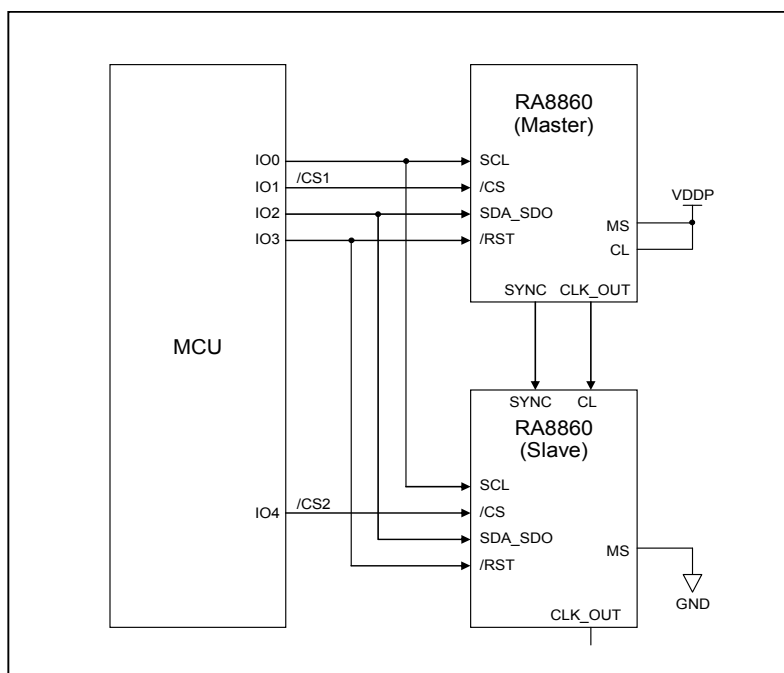


圖 7-11 : 使用 3 線 SPI 界面的串接模式

7-3 時脈與內部 RC 振盪器

RA8860 的工作時脈可由內部 RC 振盪器或外部輸入，當 “CLK_SEL” 為 Low，工作時脈來自內部 RC 振盪器，當 “CLK_SEL” 為 High，工作時脈來自接腳 “CL”，同時 RC 振盪器會被關閉。

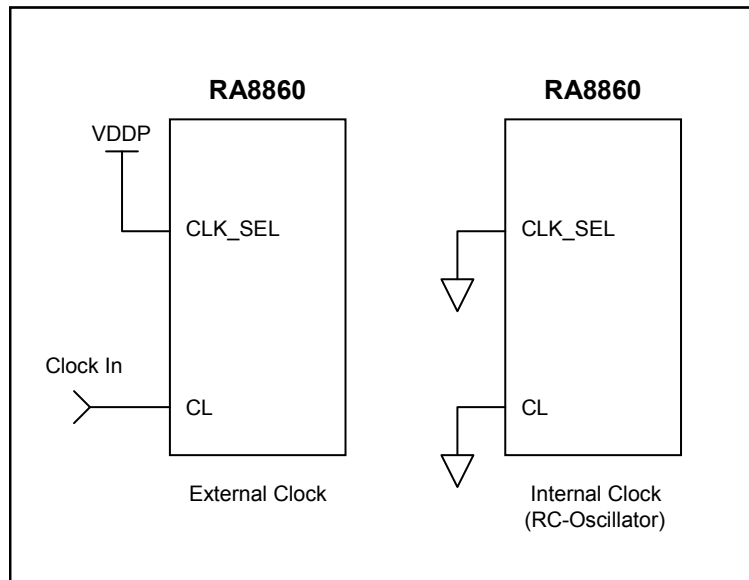


圖 7-12：工作時脈的來源

如上所述，當 “CLK_SEL” 為 Low，工作時脈來自內部 RC 振盪器，而透過暫存器 REG [B2h] 的 FR[2:0] 設定可以控制 RC 振盪器頻率，共有 8 個不同頻率選擇，當 RC 振盪器頻率改變時，LCD 掃描的 Frame Rate 也跟著變動，如下表所示：

表 7-3

FR[2:0]			Frame Rate	System Clock
0	0	0	65Hz	200KHz
0	0	1	75Hz	230KHz
0	1	0	85Hz	260KHz
0	1	1	95Hz	290KHz
1	0	0	105Hz	325KHz
1	0	1	115Hz	350KHz
1	1	0	125Hz	380KHz
1	1	1	135Hz	415KHz

7-4 顯示資料 (Display Data)

RA8860 每個Segment輸出是以 4-bit資料量來達成 16 種顏色所組成，故以 8-bit MCU的介面可以一次傳輸二筆SEG資料。此 4-bit資料可以依照 表 7-5 或 表 7-7 來對應出 8 或 16 種顏色的寫入資料，而依照暫存器 [BAh] 的設定來決定是 8 色或是 16 色模式。當在 8 色模式下時，4-bit中的最高位元可以忽略。在顯示記憶體中的所有資料 (80x2x4) 可以透過記憶體讀取來讀回，而內部電路會把這些資料轉換成相對應的Common和Segment訊號來達成彩色的顯示。

7-4-1 顯示記憶體 (Display Data RAM)

RA8860 的顯示記憶體是以 80x2x4 bits所組成來儲存每點 (Pixel) 的顯示資料，此顯示記憶體資料和位址相對應的組成關係請參考 表 7-4。

表 7-4 : Display Data RAM

RAM Alignment																															
Column																															
				39								38								----		0									
																															
R o w	MY=0		MY=1		D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0	-----	D7	D6	D5	D4	D3	D2	D1	D0	COM out	
	0		1		Display data				Display data				Display data				Display data				-----	Display data				Display data				COM0	
	1		↓		0	↑	Display data				Display data				Display data				Display data				-----	Display data				Display data			
SEG out				SEG79				SEG78				SEG77				SEG76				-----		SEG1				SEG0					

7-4-2 8 色的資料格式

SEG	SEGn+1				SEGn			
Data	X	D6	D5	D4	X	D2	D1	D0

表 7-5 : 8 色的色表 (Normally White)

Data	Color	
000	White	
001	Yellow	
010	Pink	
011	Red	
100	Cyan	
101	Green	
110	Blue	
111	Black	

表7-6 : 8 色的色表 (Normally Black)

Data	Color	
111	White	
110	Yellow	
101	Pink	
100	Red	
011	Cyan	
010	Green	
001	Blue	
000	Black	

7-4-3 16 色的資料格式

SEG	SEGn+1				SEGn			
Data	D7	D6	D5	D4	D3	D2	D1	D0

表 7-7 : 16 色的色表 (Normally White)

Data	Color		Data	Color	
0000	White		1000	Maroon	
0001	Yellow		1001	Aqua	
0010	Pink		1010	Lime Green	
0011	Red		1011	Teal	
0100	Purplish Red		1100	Green	
0101	Light Green		1101	Blue	
0110	Orange		1110	Navy Blue	
0111	Purple		1111	Black	

7-5 閃爍功能

RA8860 支援 LCD 被點亮的 Pixel 進入自動的閃爍模式，暫存器 REG [24h] 是閃爍模式的開關設定，使用者藉由暫存器 REG [26h] 可調整閃爍的快或慢，及暫存器 REG [25h] 可設定哪些顯示點要閃爍，此功能可以增加彩色 TN 的顯示變化，並且減化使用者軟體的複雜度。

7-6 重置 (Reset)

在電源開啓時，RA8860 需要一至少 1ms 時間的重置 (Reset) 脈波去啓動內部相關電路，以便進入正常工作狀態，圖 7-13 (1) 是一 "/RST" 信號的參考電路，當 "/RST" 連接到 Low，或執行一 Software Reset 指令，內部相關電路將被初始化並引發下列動作：

- 顯示關閉 (Display Off) → SEG / COM 輸出 VSS (GND) 準位
- LED 的驅動被關閉
- LCD 的相關電源電路停止工作
- RC 振盪器也會被關閉

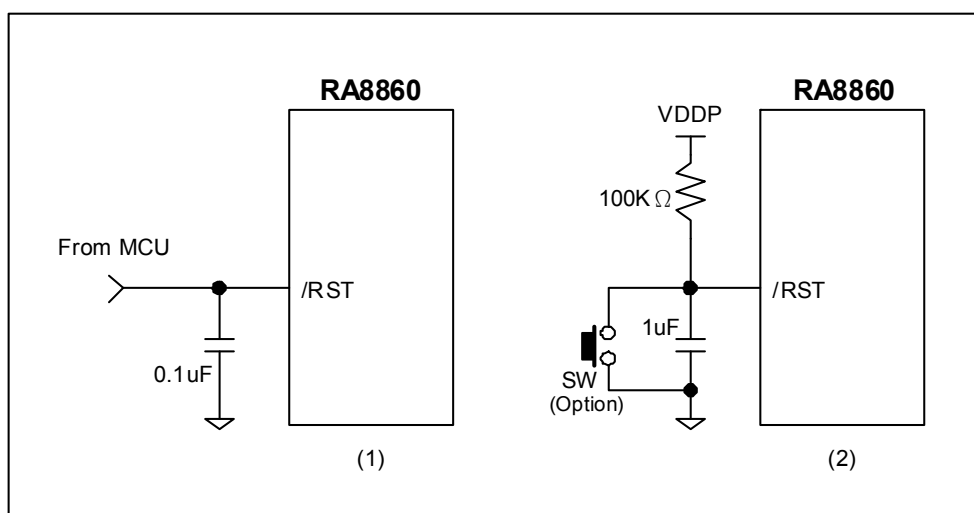


圖 7-13 : /RST 接腳的參考電路

7-7 LCD 驅動器與電壓供應電路

LCD 驅動器電源系統的操作方式描述如下 圖 7-14。電源供應電路是一個低功率、省電的設計，它包含有步階升壓電路 (Booster)、參考電壓電路 (Reference voltage)、電壓調整器 (Regulator) 和電壓隨耦器 (Voltage Follower) 可產生適當的電壓準位，供給LCD所需的電壓。

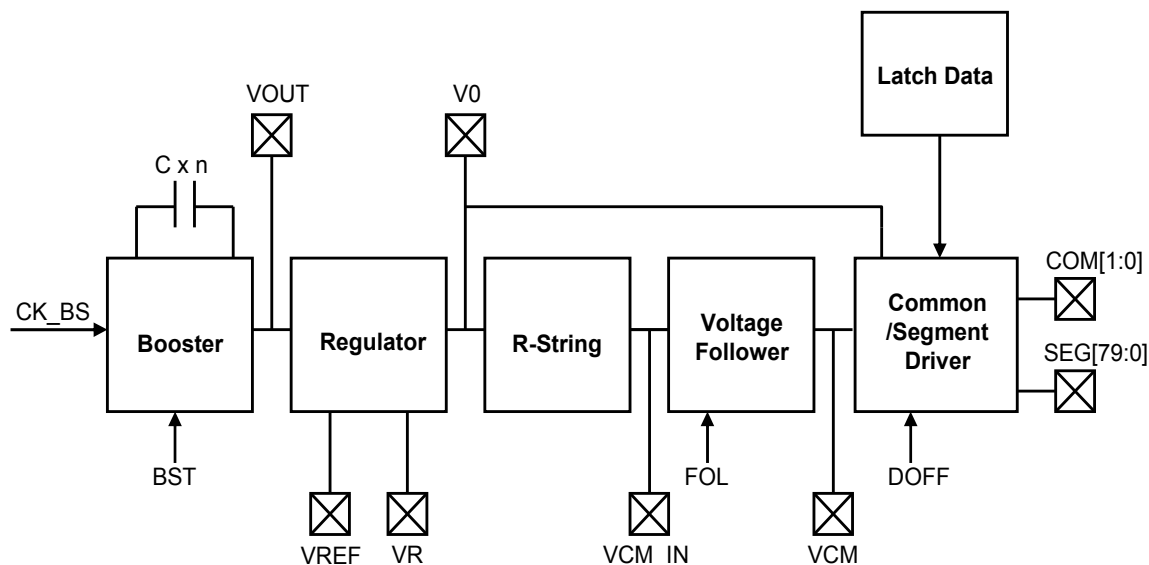


圖 7-14 : LCD 驅動器及電源供應電路方塊圖

請參閱下 表 7-8 的狀態設定去控制電源系統，使用者可設定暫存器 REG [D2h] 去選擇內部或是外部的電源供應系統來提供所需的電源。

表 7-8 : 電源電路的設定

Driver 控制 暫存器 [D2h] D3 D2 D1 D0	Booster (BST)	Voltage Follower (FOL)	Voltage Regulator (V0)	Reference Voltage (VREF)	External Power
1 1 1 1	ON	ON	ON	Internal	VDD
0 1 1 1	OFF	ON	ON	Internal	VOUT , VDD
1 1 1 0	ON	ON	ON	External	VREF , VDD
0 1 1 0	OFF	ON	ON	External	VOUT , VREF , VDD
0 1 0 0	OFF	ON	OFF	Don't Need	V0 , VDD
0 0 0 0	OFF	OFF	OFF	Don't Need	V0 , VCM , VDD

7-7-1 步階升壓電路 (Booster)

在 RA8860 的晶片內建有步階升壓電路 (Booster) 可以供應高於 VDD 的電壓給 LCD 使用，升壓電路可升壓 2 倍、3 倍、4 倍的 VDD 電壓位準作為步階升壓 VOUT，供給下一級的電壓調整器 (Regulator) 及內部 LCD 驅動電路使用。若 C1P 與 C1N 兩端跨接 1uF 的電容則會產生 2 倍 VDD 電壓，若 C2P 與 C2N 兩端也再跨接 1uF 的電容則會產生 3 倍 VDD 電壓。另外，當 C3P 與 C1N 也再跨接 1uF 的電容則產生 4 倍 VDD 電壓。

假設當電路使用 2 倍或是 3 倍壓的情況時，注意未使用到的電容正極腳位 (CxP) 需要接到 VOUT，負極腳位 (CxN) 則保持浮接，其連接方式如下圖 7-15。

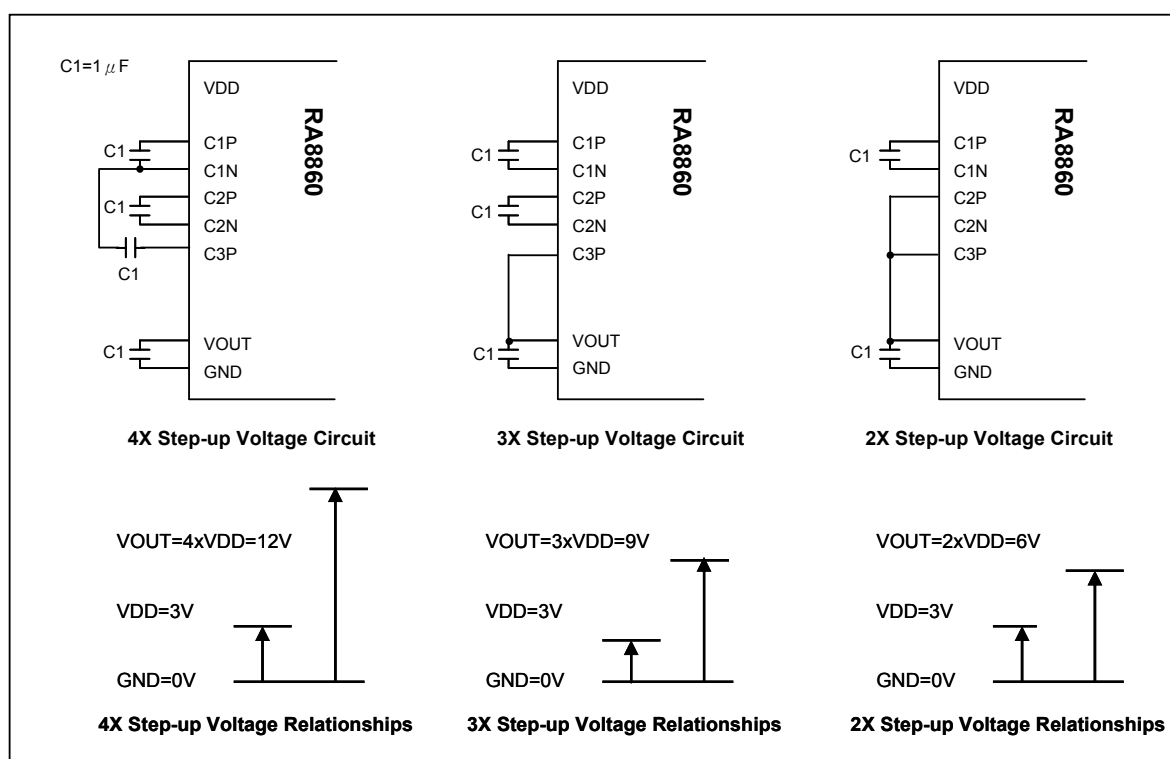


圖 7-15：步階升壓應用電路

RA8860 升壓電路的時脈是來自內部系統時脈，所以也會受到暫存器 [B2h] 的控制，而系統時脈會在升壓電路內部除頻 4 倍來做為真正升壓電路使用的頻率。另外，若使用內部的步階升壓電源供應後級的驅動器電壓，則其外部連接方式如圖 7-16 所示，如果 VOUT 電壓要由外部供應，也就是不使用內部步階升壓電路 (Booster)，其外部連接方式如圖 7-17 所示。

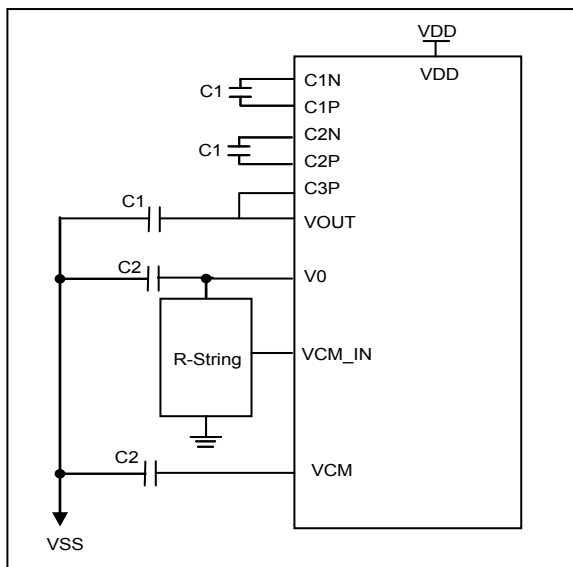


圖 7-16：使用內部升壓電源電路 (3xVDD)

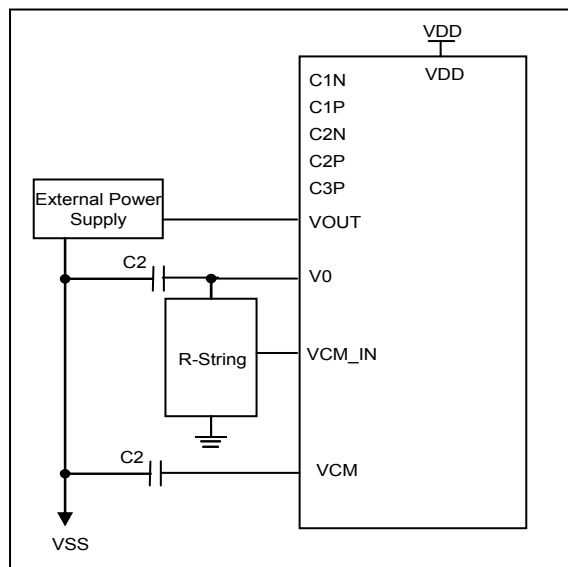


圖 7-17：使用外部 VOUT 電源電路

7-7-2 電壓調整器電路 (Regulator)

電壓調整器電路是由帶隙參考電路 (Bandgap Reference Voltage Circuit) 及非反相運算放大器所組成 (如圖 7-18 所示)，而帶隙參考電路是產生 2.1V 的參考電壓位準並供應給非反相運算放大器電路。假若暫存器 [D2h] 的第 4 個位元 (IRS) 設為 1 時，V0 的公式推導如下所示：

$$V0 = (2 + \alpha/21) * V_{REF}$$

V_{REF} 為帶隙參考電路的輸出電壓，係數 α 可由設定暫存器 [C0h] 中的 Vop[5:0] 去作 64 位階的調整。由公式得知，V0 調整的範圍可從 4.2V 至 10.5V，每一階的變動位準為 0.1V，預設值輸出為 7.4V (其中 α 為 32 時)。

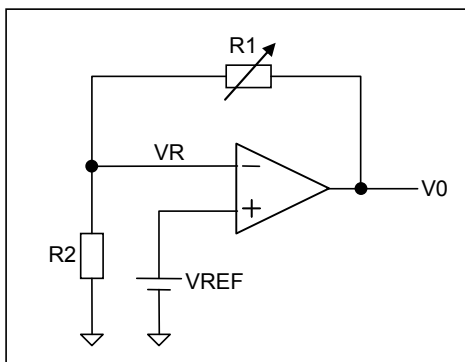


圖 7-18：電壓調整器

表 7-9 : V_{REF} 選擇

參考電壓形式	REG [D2h] D1 (V0)	REG [D2h] D0 (VREF)	V_{REF}
內部提供 V_{REF}	1	1	內部的 V_{REF}
外部提供 V_{REF}	1	0	外部的 V_{REF}

V_{REF} 可為 RA8860 內部產生或由外部輸入，請參考 表 7-9。若想對電壓調整器阻值比作調整，可使用外部的電阻比例分壓，且內部的電阻串必需對暫存器 [D2h] 的第 4 位元 (IRS) 設為不致能，另對外部電路的接法如 圖 7-19 所示。

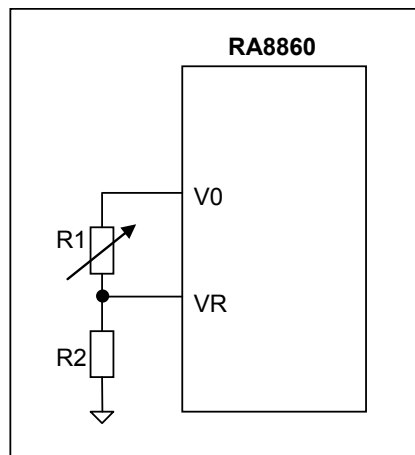


圖 7-19 : 使用外部分壓電阻

若對暫存器 [D2h] 的第 4 位元 (IRS) 設為 0 時，其 $V0$ 的公式推導如下：

$$V0 = (R1+R2) / R2 * V_{REF}$$

7-7-3 電壓隨耦器 (Voltage Follower)

內建的電壓隨耦器可提供VCM電壓準位給Common模式的驅動電路，LCD偏壓可獨立用外部的分壓電阻串輸入至電壓隨耦器的 "VCM_IN" 腳位去供應給VCM電源準位，使用者也可由內部電壓隨耦器或是外部的偏壓去供應VCM電源準位，其電路如圖 7-20 或圖 7-21 所示，且應參考表 7-8 的電源控制狀態說明。對於各種操作狀況下，電壓VOUT、V0、VCM與GND的相對關係為：

$$VOUT > V0 > VCM > GND$$

由圖 7-20 所示之電路是使用內部的電壓隨耦器，圖 7-21 連接之電路是使用外部的電壓輸入V0及VCM來供應電源電壓。

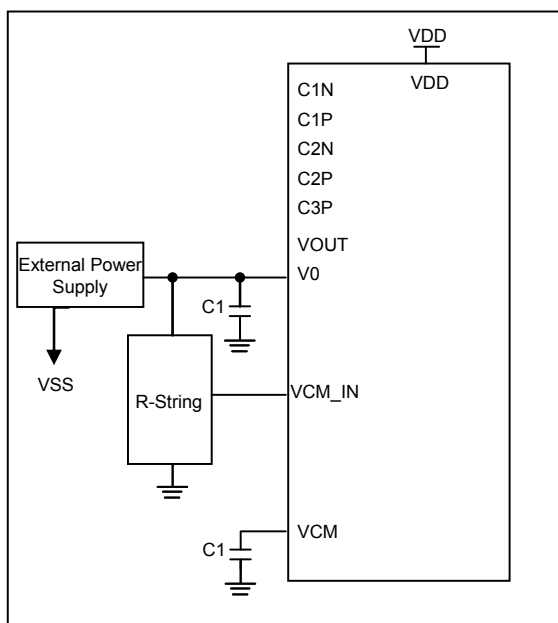


圖 7-20 : 使用內部電壓隨耦器

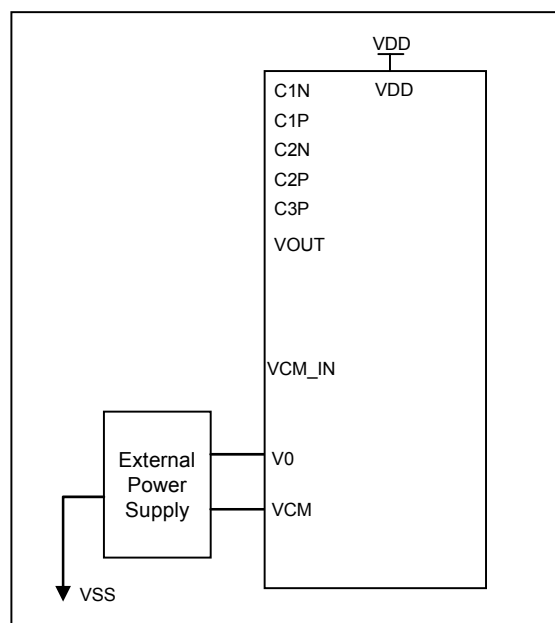


圖 7-21 : 使用外部偏壓電源

7-8 LCD 驅動輸出

RA8860 的 LCD 驅動輸出是循序的抓取內部顯示記憶體的资料，然後由時序產生器控制 LCD 的驅動信號，不停的切換成 V0、VCM 或 GND 的電壓準位，以開啓或關閉所對應的 LCD Pixel 或 Icon，RA8860 提供 80 個 Segments 與 2 個 Commons 的驅動輸出，因此總共可推動 160 個點的 LCD 面板。在顯示關閉 (Display Off) 時，RA8860 的 LCD 驅動輸出準位是 GND。

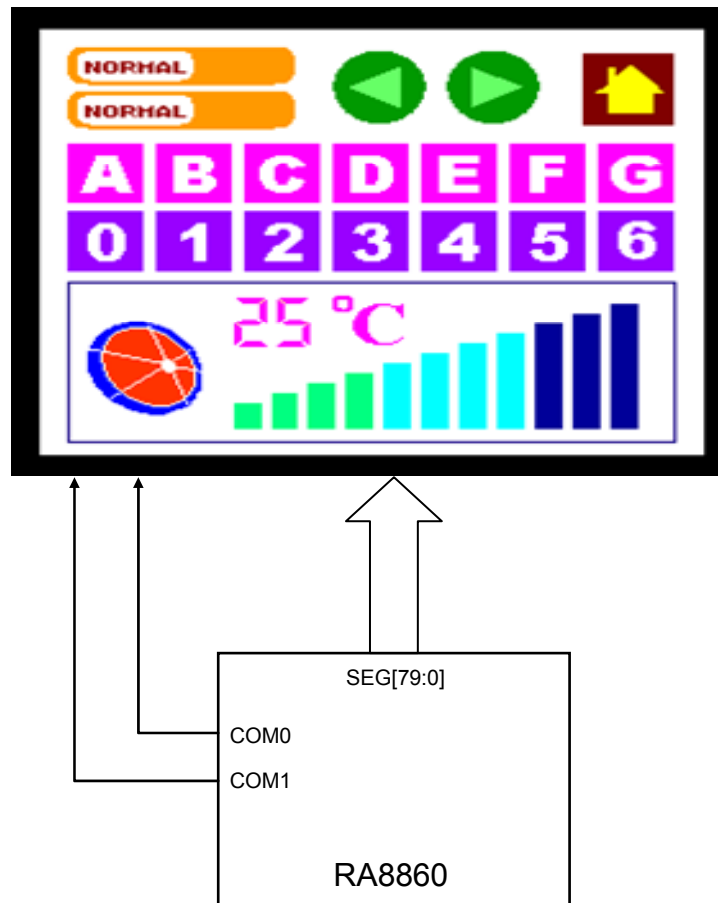


圖 7-22 : LCD 驅動界面

7-9 LED 驅動輸出

RA8860 支援 3 個 LED 驅動信號到 LED 背光面板，此 3 個 LED 驅動信號 “LED-R”、“LED-G” 和 “LED-B” 是高驅動電流的輸出，可直接推動 LED 或經由電晶體 (BJT) 來驅動，如圖 7-23 和圖 7-24 是 2 種應用電路。LED 驅動信號 “LED-R”，“LED-G” 和 “LED-B” 的工作準位 (Active Low 或 High) 可由暫存器 REG [B1h] 來設定。

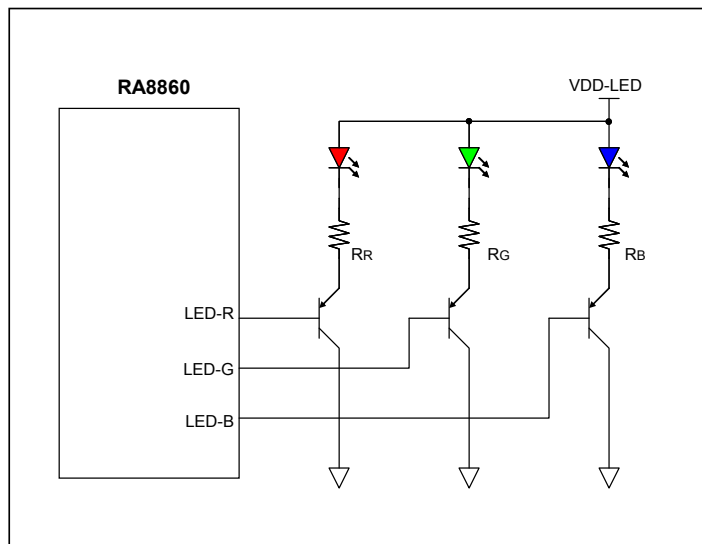


圖 7-23 : LED 驅動電路 - 1

如圖 7-24，LED 驅動信號 “LED-R”、“LED-G” 和 “LED-B” 直接推動 LED，在一般使用上 RGB 的 LED 是循序被開啓的 (Turn On)，也就是同一時間只有一種顏色 LED 開啓，每個 LED 驅動信號的驅動電流最大為 80mA，即使全部 LED 驅動信號 “LED-R”、“LED-G” 和 “LED-B” 全部開啓，最大驅動電流仍限制在 80mA，而一般使用會加上限流電阻讓電流小於 80mA，大於 80mA 或不加限流電阻都會導致不正常動作甚至損壞 IC。許多高效率的 LED 甚至 10mA 就達到很好的亮度。

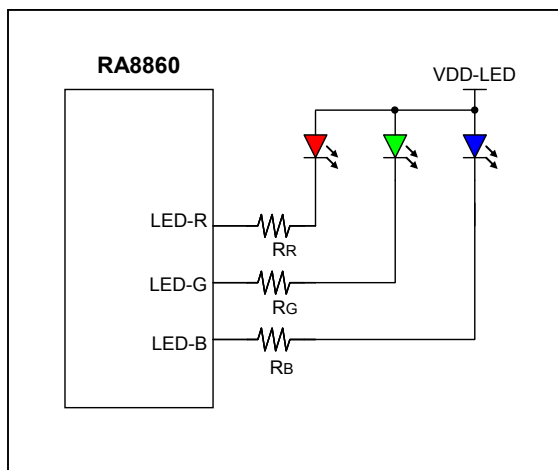


圖 7-24 : LED 驅動電路 - 2

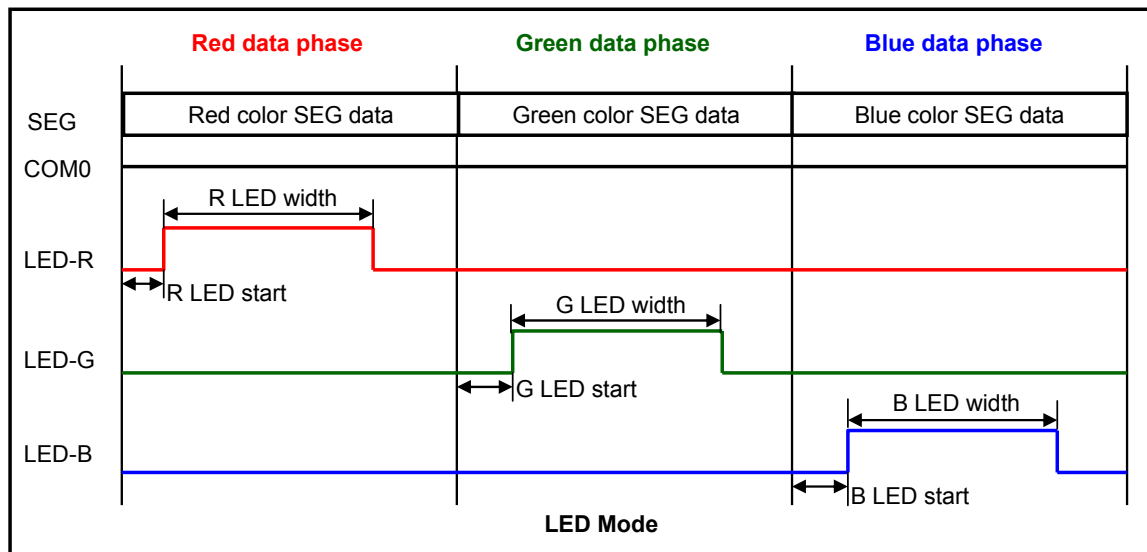


圖 7-25 : LED 時序控制

LED驅動信號“LED-R”、“LED-G”和“LED-B”的動作由暫存器REG [D4h] 控制，而這 3 個驅動信號的開啓時間 (Start) 與動作時間長短 (Active Time Width) 都由獨立的暫存器控制，請參考上圖 7-25，開啓時間由暫存器REG [A1h~A3h] 來設定，動作時間長短由暫存器 REG [A4h~A6h] 來設定。

7-10 重置 (Reset) 與休眠 (Sleep) 時的輸出信號

表 7-10：重置與休眠時的輸出信號

輸出信號	重置 (Reset)	休眠 (Sleep)
SDA_SDO	當 IF_SEL[1:0] = 11，輸出 High 電位	當 IF_SEL[1:0] = 11，保持輸出電位
COM[1:0]	輸出 Low 電位	輸出 Low 電位
SEG[79:0]	輸出 Low 電位	輸出 Low 電位
LED-R LED-G LED-B	Tri-state	Tri-state
SYNC	當 MS=1，輸出 High 電位 (註*)	當 MS=1 & CK_OEN=0，輸出 High 電位 當 MS=1 & CK_OEN=1，保持輸出電位 (註*)
CLK_OUT	輸出 High 電位	輸出 High 電位

註：當 MS=0，此腳位為輸入狀態。

8. 應用電路

下面的圖是幾個RA8860的應用電路，圖 8-1 使用 3.3V 電源及 2 倍壓產生 VOUT，並且使用 I²C 串聯 MCU 界面。LCD 電壓 (V0) 是由暫存器 REG [C0h] 來設定，同時由暫存器 REG [D2h] 來選擇內部 R-String 及參考電壓。

如果使用 5V 單一電源，那麼 圖 8-2 是一簡單的範例，此應用選擇的 MCU 界面是 3 線 SPI，因此信號線 "IF_SEL1" 必須接到 GND，"IF_SEL0" 必須接到 VDD。

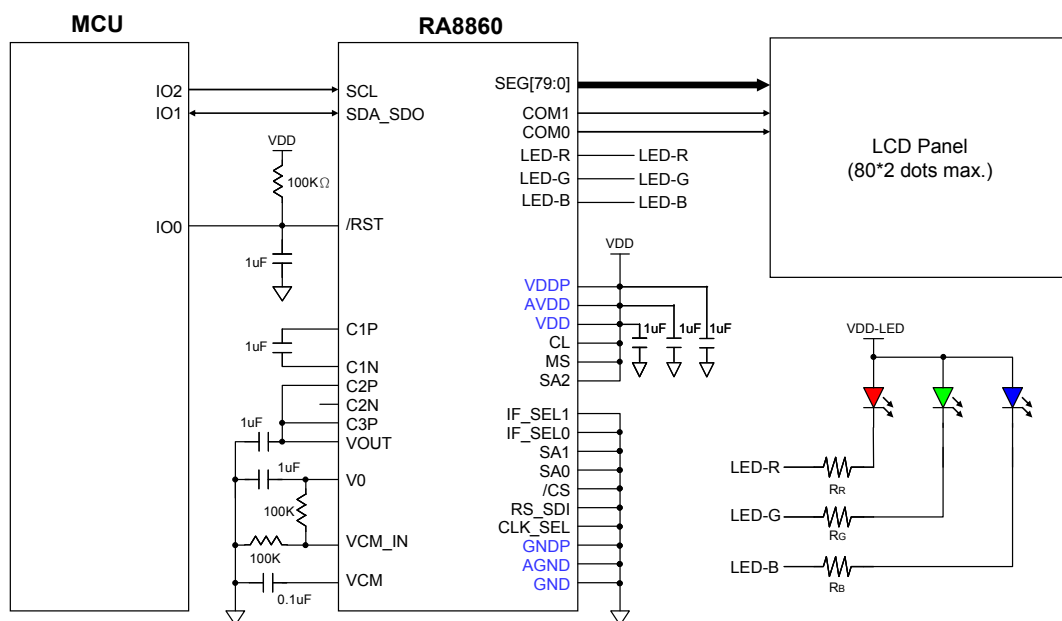


圖 8-1：應用電路 – 1，VDD=3.3V，VOUT=2xVDD，I²C 界面

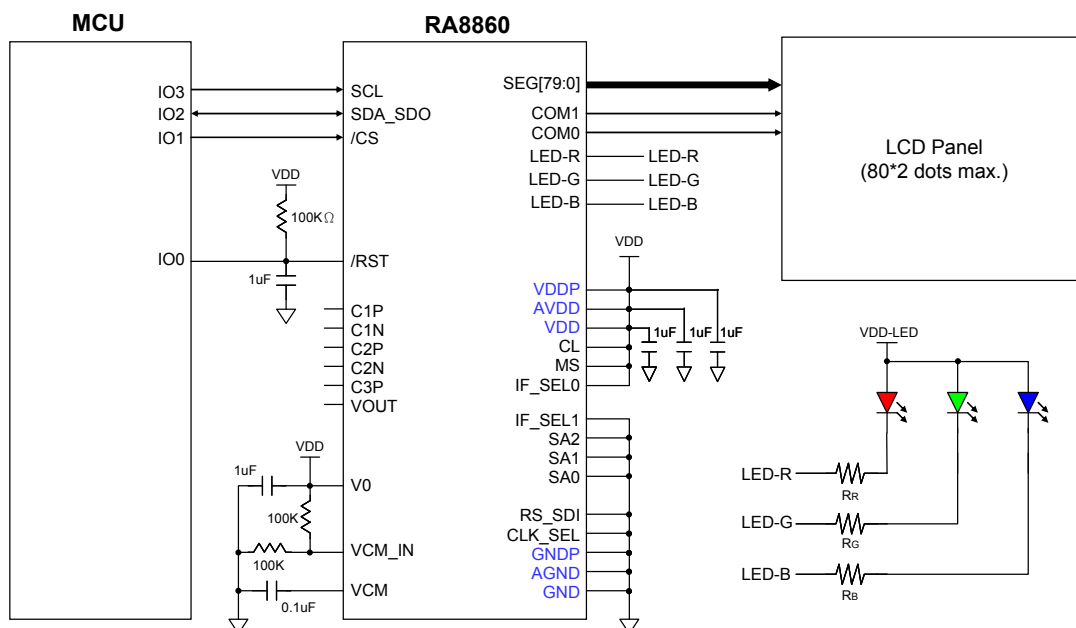


圖 8-2：應用電路 – 2，VDD=5V 單一電源，3 線 SPI 界面

圖 8-3 與上面圖 8-1 類似，但是使用串接模式用到 2 顆 RA8860，單一 3.3V 電源及 2 倍壓產生 VOUT，LCD 電壓 (V0) 由第一顆 RA8860 (Master) 產生，也因為使用 I²C 界面，因此 "SCL" 和 "SDA" 可接在一起，但必須靠 Device Address SA[2:0] 連接到不同的電壓準位。第二顆 RA8860 (Slave) 的 Booster 與 Voltage Follower 是關閉的，同時 LCD 電壓 V0 與 VCM 由第一顆 RA8860 (Master) 產生，信號 "CL" 是時脈輸入 (Clock) 來自 Master 的 "CLK_OUT" 輸出，信號 "SYNC" 輸入同樣也來自 Master 的 "SYNC"。

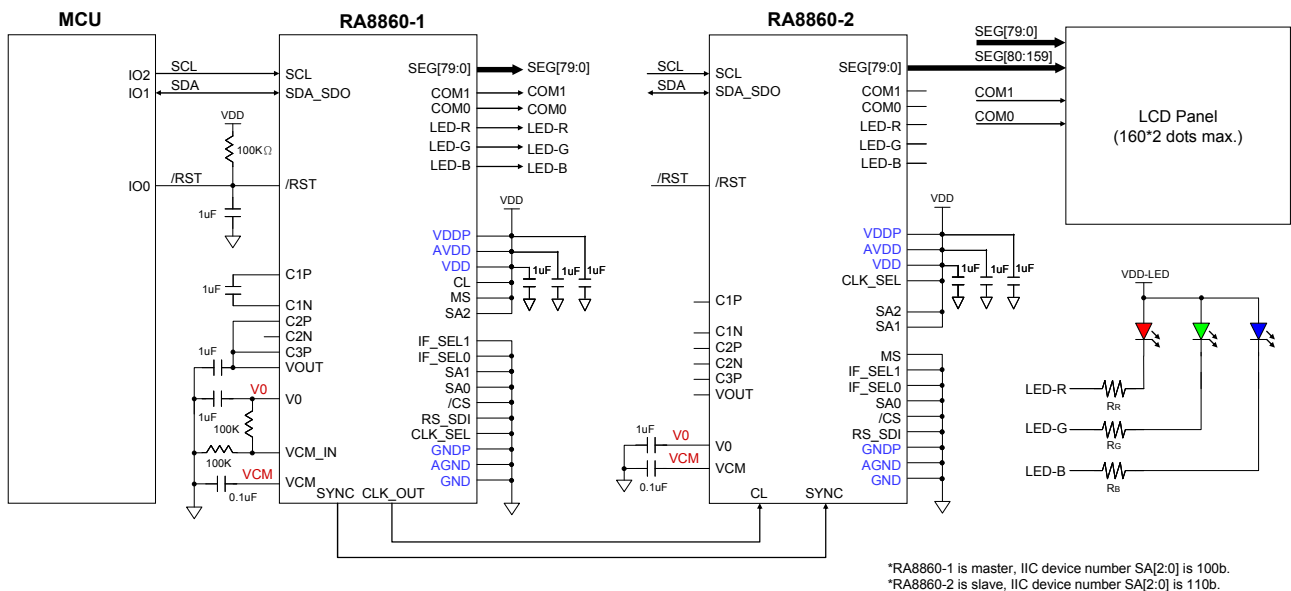


圖 8-3：應用電路 – 3，VDD=3.3V，VOUT=2xVDD，I²C 界面

9. 初始化暫存器設定程序

9-1 主模式初始化程序

RA8860 Master Mode (Single Chip)

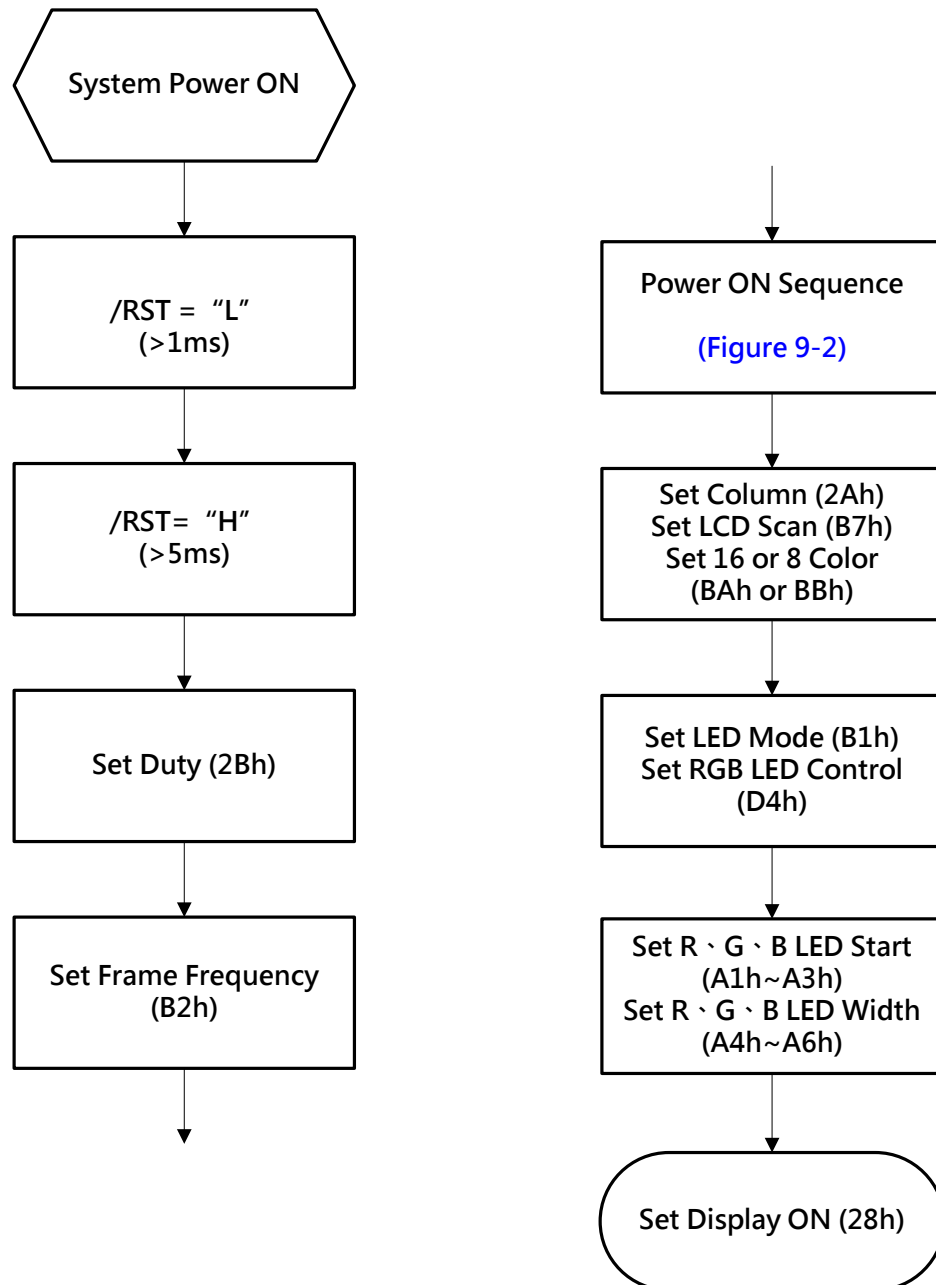


圖 9-1：主模式初始化程序

9-2 電源開啓程序

Power ON Sequence

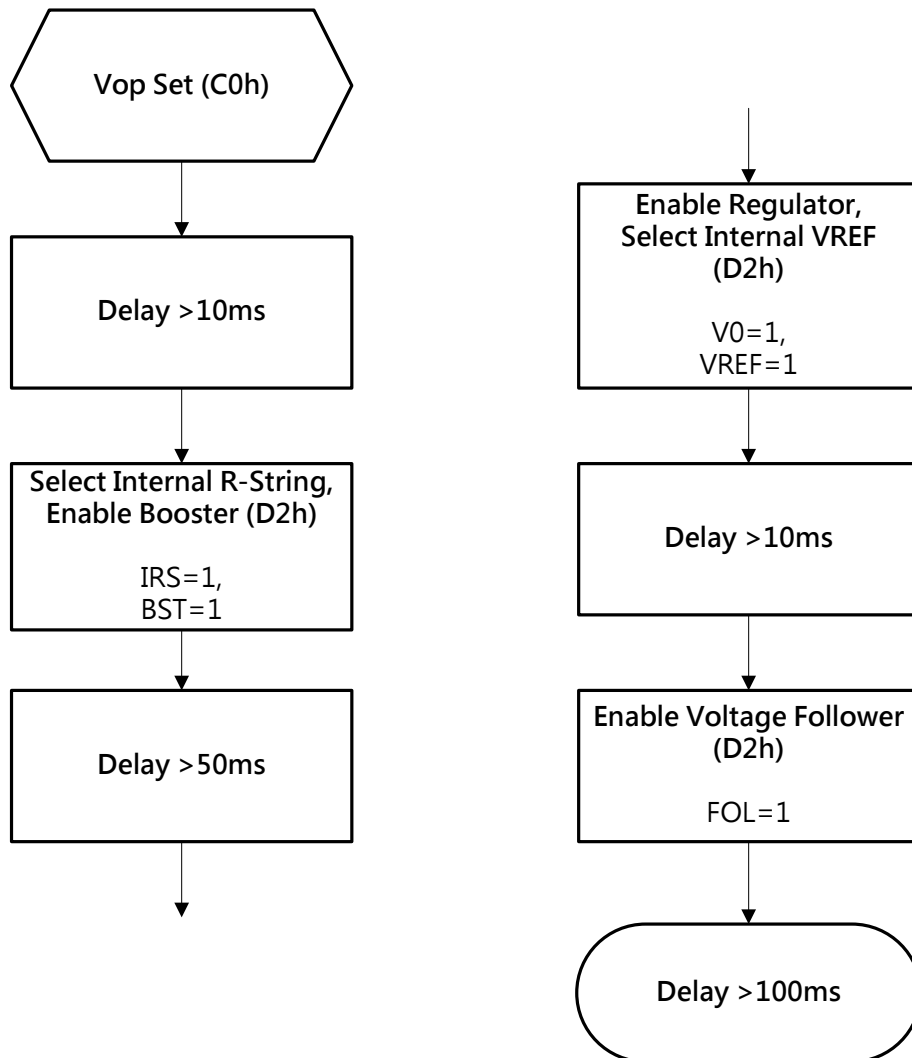
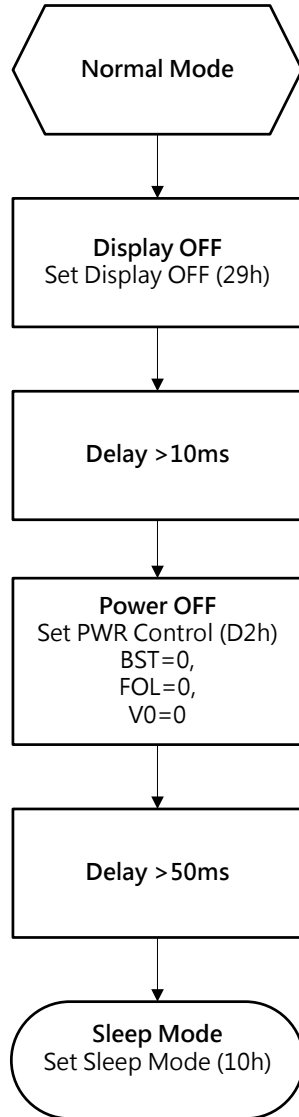


圖 9-2 : 電源開啓程序

9-3 進入休眠模式與喚醒程序

Sleep Mode Sequence



Wake Up Sequence

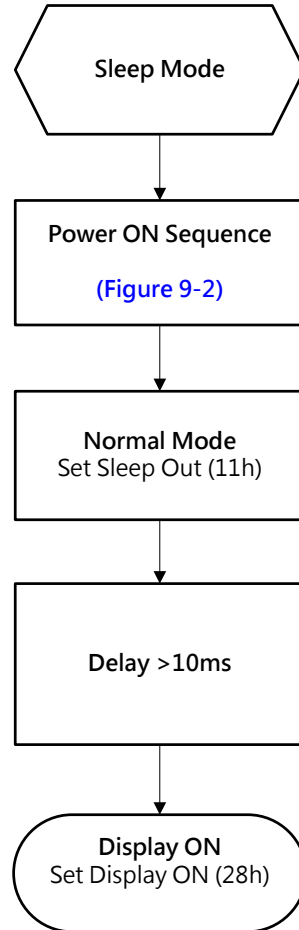


圖 9-3 : 進入休眠模式與喚醒程序

9-4 主從模式(串接)初始化程序

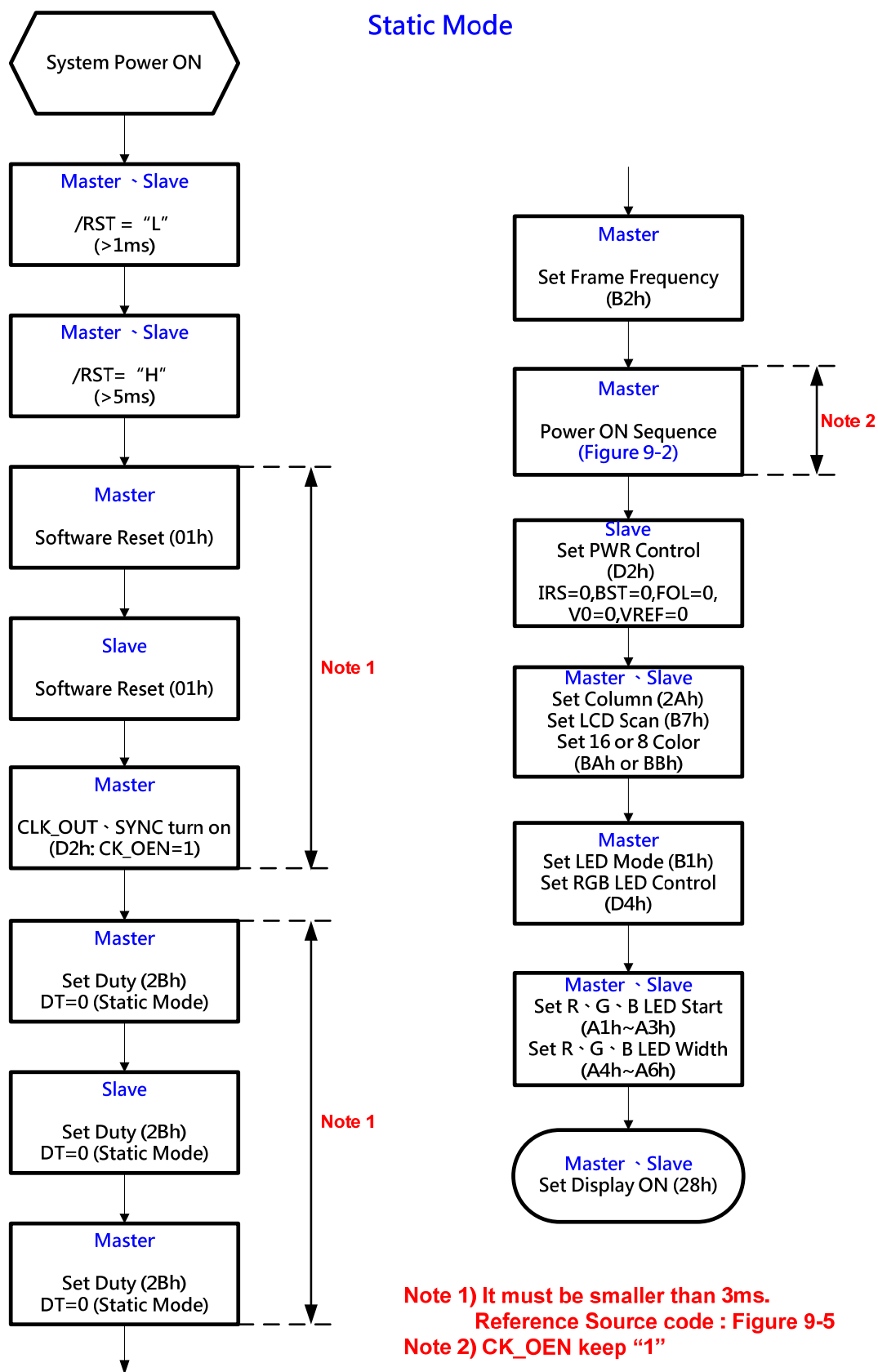


圖 9-4 : 主從模式初始化程序

9-5 Note 1) 示範程式

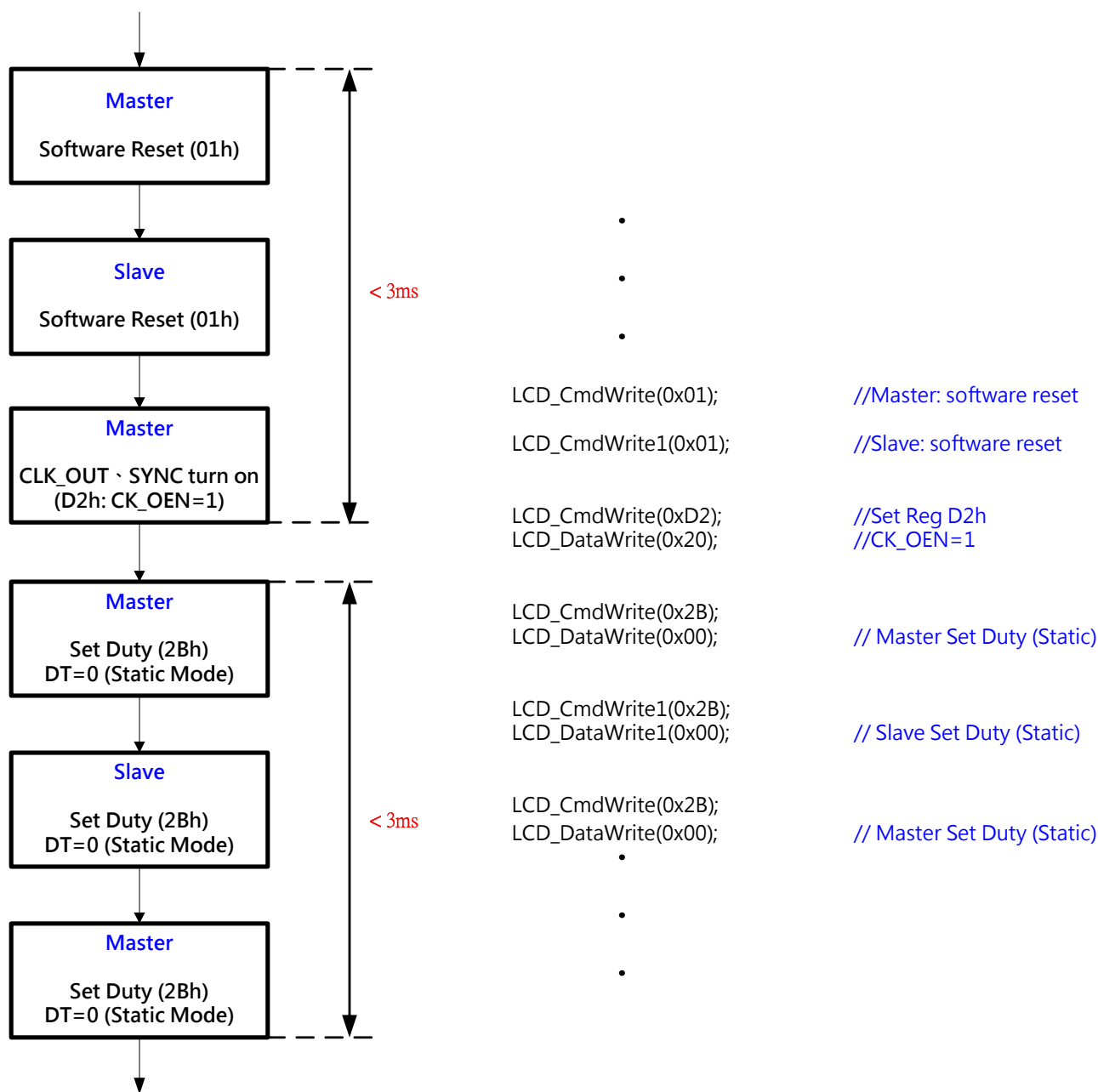


圖 9-5 : Note 1) 示範程式

10. 示範程式

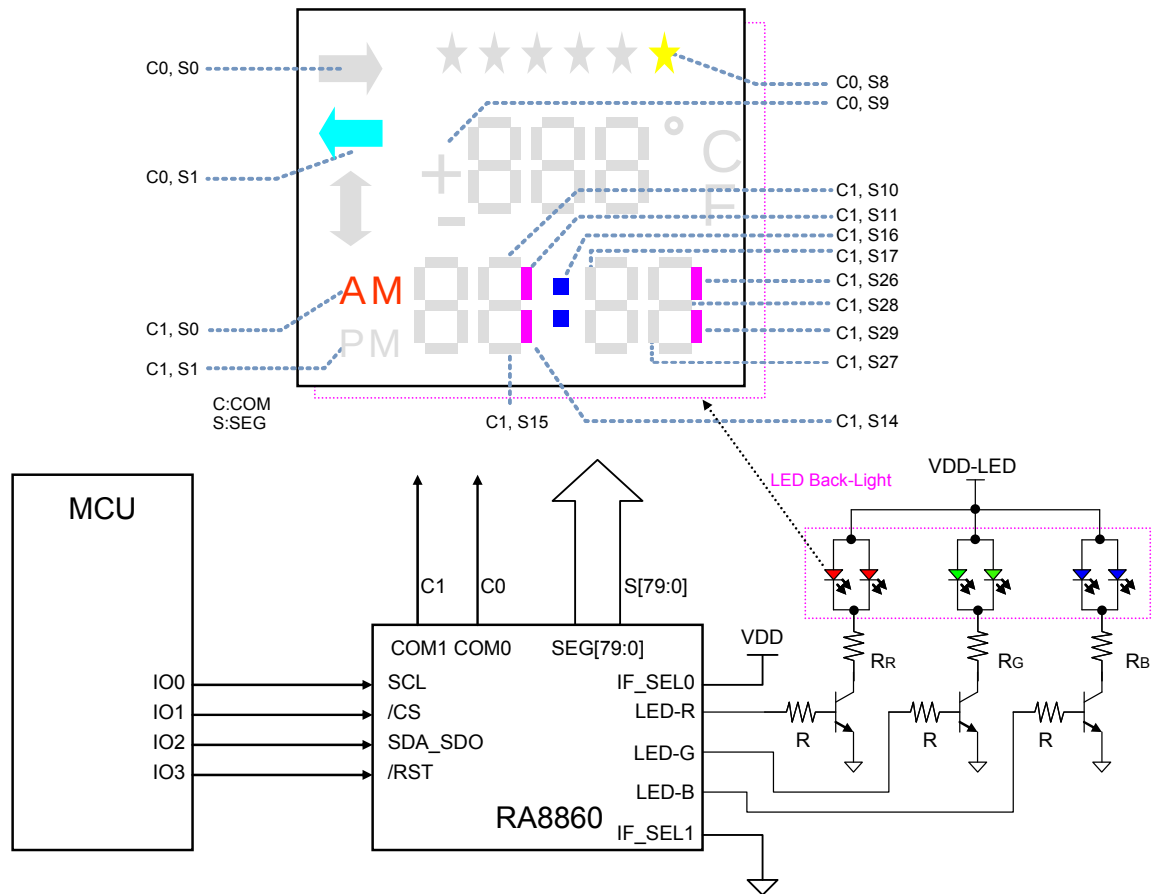


圖 10-1：示範程式的硬體架構

主程式：

```
//===== Main Process Start =====//
void main(void)
{  unsigned char i;

    P0 = 0xff;
    P1 = 0xff;
    P2 = 0xff;
    P3 = 0xff;

    LCD_Reset();           // Reset RA8860
    LCD_Initial();         // Initial RA8860
    LCD_On();              // Display on

    Address_Point(0x00);
    Memory_Write();
    for(i=0; i<=79; i++)
    {
        LCD_DataWrite(0x00);  //(C0 , S0~S79)&(C1 , S0~S79): White
    }

    Address_Point(0x00);
    Memory_Write();
    LCD_DataWrite(0x90);      //(C0,S1): Aqua , (C0,S0): White

    Address_Point(0x04);
    Memory_Write();
    LCD_DataWrite(0x01);      //(C0,S9): White , (C0,S8): Yellow

    Address_Point(0x40);
    Memory_Write();
    LCD_DataWrite(0x03);      //(C1,S1): White , (C1,S0): Red

    Address_Point(0x45);
    Memory_Write();
    LCD_DataWrite(0x20);      //(C1,S11): Pink , (C1,S10): White

    Address_Point(0x47);
    Memory_Write();
    LCD_DataWrite(0x02);      //(C1,S15): White , (C1,S14): Pink

    Address_Point(0x48);
    Memory_Write();
    LCD_DataWrite(0x0E);      //(C1,S17): White , (C1,S16): Navy Blue

    Address_Point(0x4D);
    Memory_Write();
    LCD_DataWrite(0x02);      //(C1,S27): White , (C1,S26): Pink

    Address_Point(0x4E);
    Memory_Write();
    LCD_DataWrite(0x20);      //(C1,S29): Pink , (C1,S28): White

    Blink_Speed(0x04);        // Blinking Interval : ~850ms
    Blink_Address(0x90);      //(C1,S16)
    Blink_On();               // Blinking On

    while(1);
}
```

```
// 相關副程式 :
// ===== Hardware Reset =====
void LCD_Reset(void)
{
    bRST = 0;
    Delay100ms(1);
    bRST = 1;
    Delay100ms(1);
}

// ===== 3 wire SPI Cycle =====
void LCD_CmdWrite(unsigned char LCD_data)
{
    //SPI initial
    LCD_ZCS = 1;
    LCD_SCK = 1;
    LCD_SDA = 1;
    LCD_ZCS = 0;

    // A0= 0
    LCD_SCK = 0;
    LCD_SDA = 0;
    LCD_SCK = 1;
    Delay2us(1);

    // RW = 0
    LCD_SCK = 0;
    LCD_SDA = 0;
    LCD_SCK = 1;
    LCD_SPI_SendData(LCD_data);
    LCD_ZCS = 1;
}

void LCD_DataWrite(unsigned char LCD_data)
{
    //SPI initial
    LCD_ZCS = 1;
    LCD_SCK = 1;
    LCD_SDA = 1;
    LCD_ZCS = 0;

    // A0 = 1
    LCD_SCK = 0;
    LCD_SDA = 1;
    Delay2us(2);
    LCD_SCK = 1;
    Delay2us(1);

    // RW = 0
    LCD_SCK = 0;
    LCD_SDA = 0;
    Delay2us(2);
    LCD_SCK = 1;
    LCD_SPI_SendData(LCD_data);
    LCD_ZCS = 1;
}
```

```
unsigned char LCD_DataRead(void)
{
    unsigned char LCD_data;
//SPI initial
    LCD_ZCS = 1;
    LCD_SCK = 1;
    LCD_SDA = 1;
    LCD_ZCS = 0;

// A0 = 1
    LCD_SCK = 0;
    LCD_SDA = 1;
    LCD_SCK = 1;

// RW = 1
    LCD_SCK = 0;
    LCD_SDA = 1;
    LCD_SCK = 1;
    LCD_data = LCD_SPI_GetData();
    LCD_ZCS = 1;
}

void LCD_SPI_SendData(unsigned char buf)
{
    unsigned char i,SPI_Data;

    for(i=0 ; i<8 ; i++)
    {
        SPI_Data = buf & 0x80;
        LCD_SCK = 0;
        if(SPI_Data==0)
            LCD_SDA = 0;
        else
            LCD_SDA = 1;
        LCD_SCK = 1;
        buf <<= 1;
    }
    LCD_SCK = 1;
    LCD_SDA = 1;
}

unsigned char LCD_SPI_GetData(void)
{
    unsigned char i , buf;

    for(i=0 ; i<8 ; i++)
    {
        LCD_SCK = 0;
        buf <<= 1;
        LCD_SCK = 1;

        if(LCD_SDA)
            buf |= 0x01;
    }
    LCD_SCK = 1;
    LCD_SDA = 1;
    return buf;
}
```

// ===== RA8860 Initialization =====

void LCD_Initial(void)

```
{
    LCD_CmdWrite(0x2B);
    LCD_DataWrite(0x01);    // Set Duty= 1/2 Duty

    LCD_CmdWrite(0xB2);
    LCD_DataWrite(0x01);    // Frame Frequency=75Hz
    Delay10ms(1);
```

//Power ON Sequence

```
    LCD_CmdWrite(0xC0);
    LCD_DataWrite(0x08);    // Vo=5V
    Delay10ms(1);

    LCD_CmdWrite(0xD2);
    LCD_DataWrite(0x18);    // Select Internal R-string, Enable Booster
    Delay10ms(5);

    LCD_CmdWrite(0xD2);
    LCD_DataWrite(0x1B);    // Enable Regulator, Select Internal VREF
    Delay10ms(1);

    LCD_CmdWrite(0xD2);
    LCD_DataWrite(0x1F);    // Enable Voltage Follower
    Delay10ms(10);

    LCD_CmdWrite(0x2A);
    LCD_DataWrite(0x27);    // Maximum Column = 27h = 39 (80/2 - 1)

    LCD_CmdWrite(0xBA);    // 16 color
                           // LCD_CmdWrite(0xBB); //8 color

    LCD_CmdWrite(0xB1);
    LCD_DataWrite(0x01);    // LED high active
                           // LCD_DataWrite(0x00); //LED low active

    LCD_CmdWrite(0xA1);
    LCD_DataWrite(6);       // Red LED Start Waveform Position Setting

    LCD_CmdWrite(0xA2);
    LCD_DataWrite(6);       //Green LED Start Waveform Position Setting

    LCD_CmdWrite(0xA3);
    LCD_DataWrite(6);       // Blue LED Start Waveform Position Setting

    LCD_CmdWrite(0xA4);
    LCD_DataWrite(250);     // Red LED Width Waveform Setting

    LCD_CmdWrite(0xA5);
    LCD_DataWrite(250);     // Green LED Width Waveform Setting

    LCD_CmdWrite(0xA6);
    LCD_DataWrite(250);     // Blue LED Width Waveform Setting
}
```



```
// ===== Subroutine for Display On =====
void LCD_On(void)
{
    LCD_CmdWrite(0x28);
}

// ===== Subroutine for Address Point =====
void Address_Point(unsigned char Data)
{
    LCD_CmdWrite(0x2F);
    LCD_DataWrite(Data);
}

// ===== Subroutine for Memory Write =====
void Memory_Write(void)
{
    LCD_CmdWrite(0x2C);
}

// ===== Subroutine for Blinking Interval =====
void Blink_Speed(unsigned char Data)
{
    LCD_CmdWrite(0x26);
    LCD_DataWrite(Data);
}

// ===== Subroutine for Blinking Address =====
void Blink_Address(unsigned char Data)
{
    LCD_CmdWrite(0x25);
    LCD_DataWrite(Data);
}

// ===== Subroutine for Blinking Enable =====
void Blink_On(void)
{
    LCD_CmdWrite(0x24);
    LCD_DataWrite(0x01);
}
```