

RAiO

RA8835

**Dot Matrix
LCD Controller**

Q&A

Preliminary Version 1.2

July 13, 2009

RAiO Technology Inc.

©Copyright RAiO Technology Inc. 2009

Update History		
Version	Date	Description
1.0	July 13, 2009	Preliminary Version

問題一：如何避免 RA8835 寫入或讀出資料問題？例如 RA8835 無法正確顯示，或者是 MCU 無法從 display memory 讀回一個正確的資料？

Q1: How to avoid the writing or the reading problem of RA8835, e.g. RA8835 can not correctly display on LCM or MCU can not read from display memory an un-correct data.

回覆一：LCM 的顯示系統上，通常在 PCB 上會有一個連接 MCU 以及 RA8835 的 bus line 架構，爲了避免錯誤顯示或錯誤讀寫，我們會建議在 bus line 上加一 1K~3.3KΩ 的 pull-up resistor 來穩定 bus 信號。或者，MCU interface 的 cable 儘量勿長於 30 公分。

Ans. 1: LCM display system ordinary have a bus line configuration on printed circuit boards for data communication between MCU and RA8835. To avoid the display/ the access error, we suggest that bus signals might be better stabilized by pulling up to VCC using the resistors around 1k-3.3 kΩ or keeping the length of MCU interface less then 30CM.

請參考下列相關圖示說明：

Please refer to the following picture.

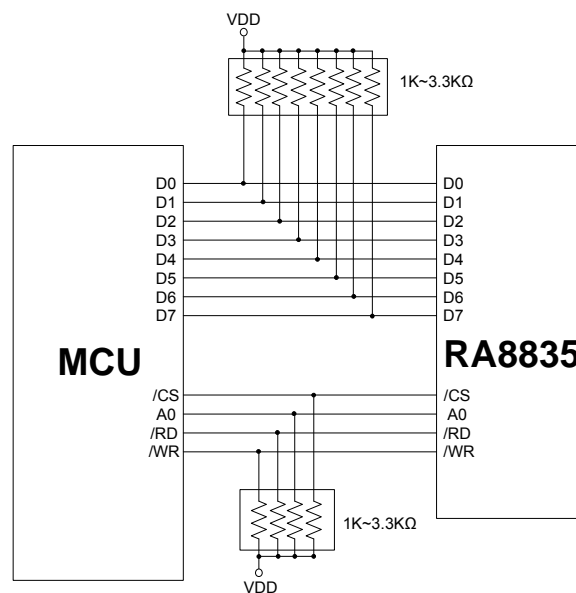


Figure 1 Pull-up resistor

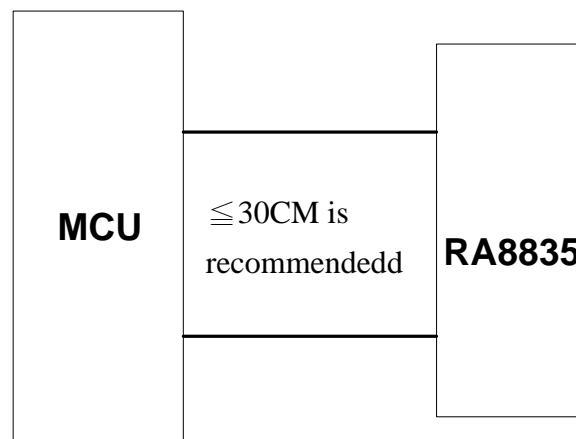


Figure 2 MCU interface wire length

問題二. RA8835/RA8835A 暫存器的初值為何?

Q2: What are the register default values of the RA8835/RA8835A?

回覆二：請參考下列圖表說明：

Ans2: Please refer to the following table.

Class	Command	Hex	Parameter	Default value	
				RA8835	RA8835A
System control	System Set	40	P1	00H	00H
			P2	00H	00H
			P3	00H	00H
			P4	00H	00H
			P5	40H	40H
			P6	F0H	F0H
			P7	00H	00H
			P8	00H	00H
Display control	Display ON/OFF	58/59	P1	OFF(0)	ON(1)
	SCROLL	44	P1	00H	00H
			P2	00H	00H
			P3	00H	00H
			P4	00H	00H
			P5	00H	00H
			P6	00H	00H
			P7	00H	00H
			P8	00H	00H
			P9	00H	00H
			P10	00H	00H
	CSRFORM	5D	P1	00H	00H
			P2	00H	00H
	CGRAM ADR	5C	P1	00H	00H
			P2	00H	00H
CSRDIR	4C to 4F	NONE	4CH	4CH	
HDOT SCR	5A	P1	00H	00H	
OVLAY	5B	P1	00H	1CH	
Drawing control	CSRW	46	P1	00H	00H
			P2	00H	00H
	CSRR	47	NONE	00H	00H

問題三. RA8835/RA8835A 的 ESD/Latch up 特性為何?

Q3: What are the ESD and Latch up features of the RA8835/RA8835A?

回覆三：請參考下列圖表說明：

Ans3: Please refer to the following table.

	RA8835	RA8835A
ESD Sensitivity	+4KV/-5KV,	+/-6.5KV,
Pass(HBM)	V Class-3	V Class-3
Latch-up	+/-200mA, IT Class-3	+/-600mA, IT Class-3

問題四.在 RA8835 Data sheet 版本 2.0 的 82 和 83 頁，這裡有兩個不同的 8080 時序圖，何時為其各別的使用時機。

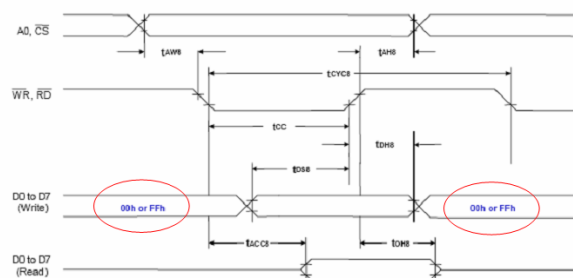
Q4: There are two different 8080 timing diagrams in the page 82 and 83 of the Datasheet Rev. 2.0. of RA8835. When does each one apply?

回覆四：就 8080 系列的介面而言，在 datasheet 2.0 的 timing-1 以及 timing-2 是有些許的不同。以 timing-2 來說，資料存取的週期必須要大於 write pulse(/WR 保持在低準位的期間)。在 timing-1 則不需要如同 timing-2 的動作，但是在存取 RA8835 之前，data bus 必須要先預設為 00H 或者是 FFH。相關的圖示說明，請參照下方的時序圖。

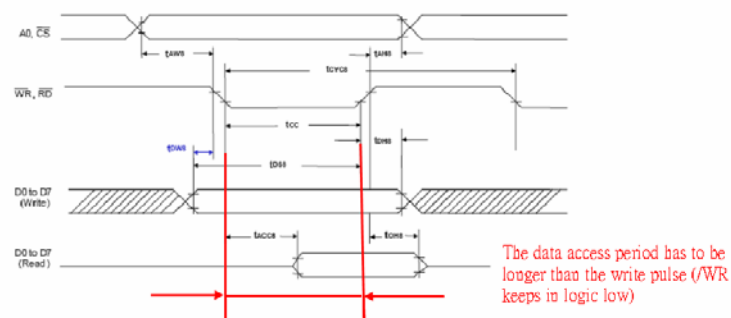
Ans4: In 8080 family interface, there is a little bit difference between timing-1 and timing-2 of the Datasheet Rev. 2.0.

In the case of timing-2, the data access period has to be longer than the write pulse (/WR keeps in logic low). The timing-1 need not to be like that, but the data bus should be set to 00H/FFH before accessing RA8835. The related timing charts are shown below, please have a look.

10-3-1a 8080 Family Interface Timing-1



10-3-1b 8080 Family Interface Timing-2



問題五.使用 RA8835 如何減少顯示畫面上的雪花現象？

Q5: How to reduce the flicker/snow effect with RA8835?

回覆五：當 RA8835 內部的掃瞄線路在處理掃瞄任務時，若同時經由 MPU 來做存取 DDRAM 的動作，那麼掃瞄資料將被強迫的到一個錯誤的值而造成閃爍或雪花現象。當閃爍或雪花現象太嚴重時，其將使影響整體的顯示畫面。我們會建議使用使用者在寫入資料到 DDRAM 之前，做一個 check busy 的動作來減少閃爍或雪花的現象。相關的程式說明如下：

Ans5: When the scan logic of RA8835 is processing scan task at the same time the DDRAM is accessed by the MPU, the scan data will be forced to get a wrong one and it might cause the flicker or the snow effect. When the flicker or snow is too much, it will infect the display appearance. To reduce the flick/snow effect, it is suggested that user can use a check busy subroutine before writing data into DDRAM. The related demo program is described as below.

```
void lcd_datawrite(unsigned char datax)
{
    lcd_chkbusy();           //It is recommended that execute check busy subroutine before accessing display
memory
    lcd_SEL1 = 0;
    lcd_A0 = 0;
    lcd_DATA = datax;
    lcd_CS = 0;
    lcd_WR = 0;
    lcd_WR = 1;
    lcd_CS = 1;
    lcd_A0 = 1;
}
```

```
unsigned char lcd_cmdread(void)
{
    unsigned char address_cmd;
    lcd_DATA=0xff;
    lcd_SEL1 = 0;
    lcd_A0 = 0;
    lcd_CS = 0;
    lcd_RD = 0;
    address_cmd = lcd_DATA;
    lcd_RD = 1;
    lcd_CS = 1;
    lcd_A0 = 0;
    return(address_cmd);
}
```

```
void lcd_chkbusy(void)
{
    unsigned char busy;
    do {
        busy =(lcd_cmdread() & 0x40);
    }while(busy);
}
```

問題六.如何在 RA8835 外部的 CG-ROM 上自建一個新的字型？

Q6: How to create a new character (symbol/pattern) on the CG-ROM of RA8835?

回覆六：請參考下列範例程式。

Ans6: Please refer to the following sample program.

```
void main(void)
{

    lcd_reset();
    lcd_initial();           // initial setting
    lcd_clear();           // Clear memory

    //=====
    // Display a 8x8 / 8x16 char (External CG-ROM)
    //-----
    // M0 = 1 for External CG-ROM
    // M1 = 1 for D6 correction, 0 for N0 D6 correction
    // M2 = 1 for 8x16, 0 for 8x8
    //=====
    lcd_cmdwrite(0x40);      //SYSTEM SET
    lcd_datawrite(0x35);    //p1 ==> 8x16 , No D6 Correction, External CG-ROM

    lcd_cmdwrite(0x59);    //DISP ON, NOTE ==> this command is always following
                          //SYSTEM SET CMD

    char_generate_EX_CGROM(0xF050, Ext_char1);
    char_generate_EX_CGROM(0xF060, Ext_char2);

    lcd_cursorxy(0x00,0x00);
    lcd_cmdwrite(0x42);
    lcd_datawrite(0x05);
    lcd_datawrite(0x06);

}

void lcd_reset(void)
{

    lcd_RES = 1;
    delay1ms(2);
    lcd_SEL1 = 0;
    lcd_A0 = 0;
    lcd_WR = 1;
    lcd_RD = 1;
    lcd_CS = 1;
    lcd_RES = 0;
    delay1ms(100);
    lcd_RES = 1;
    delay1ms(500);

}

void lcd_initial(void)
{

    lcd_cmdwrite(0x40);      //SYSTEM SET
    lcd_datawrite(0x30);    //p1 ==> single panel, no screen top-line
    lcd_datawrite(0x87);    //p2 ==> Font Width = 8
    lcd_datawrite(0x0F);    //p3 ==> Font Height = 15
    lcd_datawrite(0x27);    //p4 ==> Set the address range covered by one display line.
    lcd_datawrite(0x30);    //p5 ==> line length = 48 bytes = TC/R = 48 = 30h
    lcd_datawrite(0xef);    //p6 ==> L/F
    lcd_datawrite(0x28);    //p7 ==> Horizontal address range = 28h = 40
    lcd_datawrite(0x00);    //p8
```

```
    lcd_cmdwrite(0x44);        //SCROLL
    lcd_datawrite(0x00);      //sa1-L
    lcd_datawrite(0x00);      //sa1-H
    lcd_datawrite(0xf0);      //sL1    0xf0

    lcd_datawrite(0x80);      //sa2-L
    lcd_datawrite(0x25);      //sa2-H
    lcd_datawrite(0xf0);      //sL2

    lcd_datawrite(0x00);      //sa3-L
    lcd_datawrite(0x70);      //sa3-H

    lcd_datawrite(0x00);      //sa4-L
    lcd_datawrite(0x80);      //sa4-H

    lcd_cmdwrite(0x5a);       //HDOT SCR
    lcd_datawrite(0x00);      // 0x00

    lcd_cmdwrite(0x58);       //DISP OFF
    lcd_datawrite(0x54);

    lcd_cmdwrite(0x5d);       //CSR FORM
    lcd_datawrite(0x07);      //CRX
    lcd_datawrite(0x07);      //CRY

    lcd_cmdwrite(0x59);       //DISP ON
    lcd_datawrite(0x54);

    lcd_cmdwrite(0x5b);
    lcd_datawrite(0x01);

    lcd_cmdwrite(0x4c);       // Cursor direction ==> Right
}

void lcd_clear(void)
{
    unsigned int i;
    lcd_cmdwrite(0x4c);       // Direction ==> Right

    lcd_cursorxy(0x00,0x00);
    lcd_cmdwrite(0x42);

    for(i=0;i<0xffff;i++)
    {
        lcd_datawrite(0x00);
    }
}

//=====
//This subroutine is for creating a new character on external CG-ROM
//Note : The address of CGRAM_SA must be set at F000H to FFFFH )
//=====
void char_generate_EX_CGRAM(unsigned int CGRAM_SA, char code *char_data)
{
    int a, pos_H, pos_L;
```



```
pos_H = CGRAM_SA >> 8;
pos_L = CGRAM_SA & 0x00ff;

lcd_cmdwrite(0x4c);

lcd_cmdwrite(0x46);
lcd_datawrite(pos_L);
lcd_datawrite(pos_H);

lcd_cmdwrite(0x42);
for(a=0;a<16;a++)
{
    lcd_datawrite(*char_data);
    char_data++;
}
}

char code Ext_char1[]=
{
    0x00,0x03,0x3f,0x1d,0x01,0x01,0x01,0x0f,0x07,0x03,0x01,0x01,0x01,0x07,0x7f,0x38,
};

char code Ext_char2[]=
{
    0x38,0xfc,0xc0,0x00,0x80,0x80,0x80,0xe0,0xf0,0xe0,0x80,0x80,0x9c,0xfe,0xff,0x00,
};
```

問題七.如何在 RA8835 內部的 CG-RAM 上自建一個新的字型？

Q7: How to create a new character (symbol/pattern) on the internal CG-RAM of RA8835?

回覆七：請參考下列範例程式。

Ans7: Please refer to the following sample program.

```
void main(void)
{

    lcd_reset();
    lcd_initial();          // initial setting
    lcd_clear();           // Clear memory

    //=====
    // Display a 8x8 / 8x16 / 16x16 char (Internal CG-RAM)
    //-----
    // M0 = 0 for Internal CG-RAM,
    // M1 = 1 for D6 correction, 0 for N0 D6 correction
    // M2 = 1 for 8x16, 0 for 8x8
    //=====
    lcd_cmdwrite(0x40);      //SYSTEM SET
    lcd_datawrite(0x34);     //p1 ==> 8x16 , No D6 Correction, Internal CG-RAM

    lcd_cmdwrite(0x59);     //DISP ON, NOTE ==> this command is always following
                           //SYSTEM SET CMD

    char_generate(0x4800, Ext_char1);
    char_generate(0x4810, Ext_char2);
```

```
    lcd_cursorxy(0x00,0x00);
    lcd_cmdwrite(0x42);
    lcd_datawrite(0x80);
    lcd_datawrite(0x81);
}

void lcd_reset(void)
{
    lcd_RES = 1;
    delay1ms(2);
    lcd_SEL1 = 0;
    lcd_A0 = 0;
    lcd_WR = 1;
    lcd_RD = 1;
    lcd_CS = 1;
    lcd_RES = 0;
    delay1ms(100);
    lcd_RES = 1;
    delay1ms(500);
}

void lcd_initial(void)
{
    lcd_cmdwrite(0x40);           //SYSTEM SET
    lcd_datawrite(0x30);         //p1 ==> single panel, no screen top-line
    lcd_datawrite(0x87);         //p2 ==> Font Width = 8
    lcd_datawrite(0x0F);         //p3 ==> Font Height = 15)
    lcd_datawrite(0x27);         //p4 ==> Set the address range covered by one display line.
    lcd_datawrite(0x30);         //p5 ==> line length = 48 bytes = TC/R = 48 = 30h
    lcd_datawrite(0xef);         //p6 ==> L/F
    lcd_datawrite(0x28);         //p7 ==> Horizontal address range = 28h = 40
    lcd_datawrite(0x00);         //p8

    lcd_cmdwrite(0x44);           //SCROLL
    lcd_datawrite(0x00);         //sa1-L
    lcd_datawrite(0x00);         //sa1-H
    lcd_datawrite(0xf0);         //sL1    0xf0

    lcd_datawrite(0x80);         //sa2-L
    lcd_datawrite(0x25);         //sa2-H
    lcd_datawrite(0xf0);         //sL2

    lcd_datawrite(0x00);         //sa3-L
    lcd_datawrite(0x70);         //sa3-H

    lcd_datawrite(0x00);         //sa4-L
    lcd_datawrite(0x80);         //sa4-H

    lcd_cmdwrite(0x5a);         //HDOT SCR
    lcd_datawrite(0x00);         // 0x00

    lcd_cmdwrite(0x58);         //DISP OFF
    lcd_datawrite(0x54);
```

```
    lcd_cmdwrite(0x5d);        //CSR FORM
    lcd_datawrite(0x07);      //CRX
    lcd_datawrite(0x07);      //CRY

    lcd_cmdwrite(0x59);        //DISP ON
    lcd_datawrite(0x54);

    lcd_cmdwrite(0x5b);
    lcd_datawrite(0x01);

    lcd_cmdwrite(0x4c);        // Direction ==> Right
}

void lcd_clear(void)
{
    unsigned int i;
    lcd_cmdwrite(0x4c);        // Direction ==> Right

    lcd_cursorxy(0x00,0x00);
    lcd_cmdwrite(0x42);

    for(i=0;i<0xffff;i++)
    {
        lcd_datawrite(0x00);
    }
}

//=====
// This subroutine is used for creating a new character on the internal CG-RAM
//=====
void char_generate(unsigned int CGRAM_SA, char code *char_data)
{
    int a, pos_H, pos_L;

    pos_H = CGRAM_SA >> 8;
    pos_L = CGRAM_SA & 0x00ff;

    lcd_cmdwrite(0x5c);
    lcd_datawrite(0x00);      // Set the start address at the 4000h
    lcd_datawrite(0x40);

    lcd_cmdwrite(0x4c);

    lcd_cmdwrite(0x46);
    lcd_datawrite(pos_L);
    lcd_datawrite(pos_H);

    lcd_cmdwrite(0x42);
    for(a=0;a<16;a++)
    {
        lcd_datawrite(*char_data);
        char_data++;
    }
}

char code Ext_char1[]=
{
```

```
    0x00,0x03,0x3f,0x1d,0x01,0x01,0x01,0x0f,0x07,0x03,0x01,0x01,0x01,0x07,0x7f,0x38,  
};  
  
char code Ext_char2[]=  
{  
    0x38,0xfc,0xc0,0x00,0x80,0x80,0x80,0xe0,0xf0,0xe0,0x80,0x80,0x9c,0xfe,0xff,0x00,  
};
```