ORIENT DISPLAY (North America) Limited
145 Royal Crest Court Unit 42 Markham, ON, Canada L3R 9Z4
Tel: 905-477-1166 Fax: 905-477-1782 http://www.orientdisplay.com

SPECIFICATION
FOR
LCM MODULE

MODULE NO: **AFA480272C262K-4.3-M7**

Customer Approval:

	SiGNATURE	DATE
PREPARED BY (RD ENGiNEER)		
PREPARED BY (QA ENGiNEER)		
CHECKED BY		
APPROVED BY		

DOCUMENT REVISION HISTORY

Version	DATE	DESCRIPTION	CHANGED BY
00	9-28-2010	First issue	

CONTENTS

1. Functions & Features

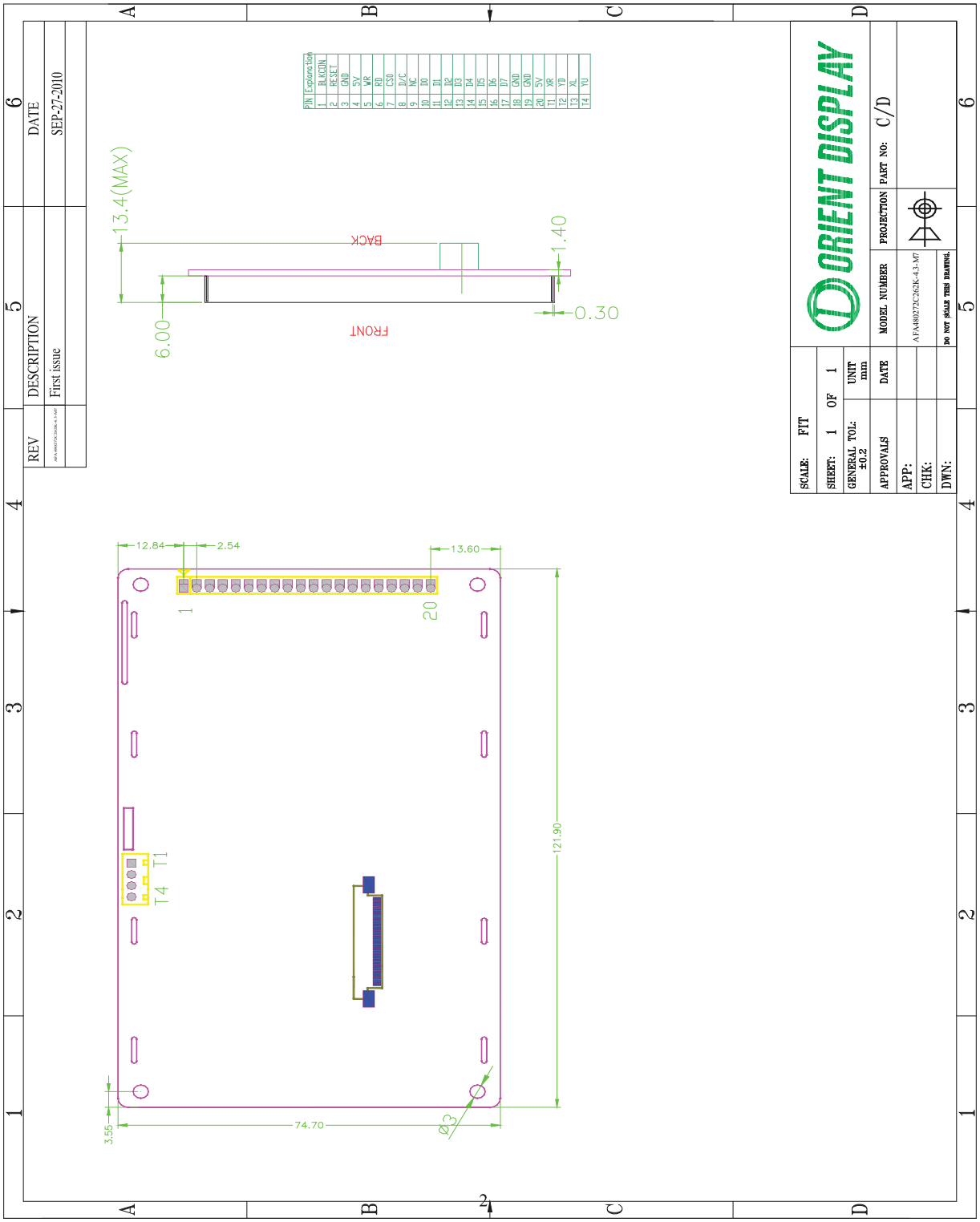
1.1. Format : 480*272 Dots
1.2. LCD mode : Normally White, Transmissive
1.3. Viewing direction : 6 O'clock
1.4. Display color : Up to 262K color
1.5. Operation temp : -20=70
1.6. Storage temp : -30=80
1.7. Power supply voltage (VDD) : 5.0V
1.8. LED power voltage : 23 V
1.9. Backlight color : White(LED)
1.10 LCM Contrast ratio : 500:1
1.11 LCM Birghtness : 200 nit(tye)
1.12.RoHS standard
1.13.LCM work in 8080 Mode

2. MECHANICAL SPECIFICATIONS

2.1. Module size : 122.0mm(L)*75.0mm(W)*14.5 max mm (H)
2.2. Viewing area : 95.04mm(L)*53.86mm(W)
2.3. Dot pitch : 0.198(W)*0.198(H) mm
2.4. Weight : Approx.

Item	STANDARD Value	Unit
LCD type	16.7M a-Si TFT-LCD TRANSMISSVIE	
Dot arrangement	480(R.G.B)*272	Dot
Module size	105.5(W)*67.2(H)*2.89(T)	mm
Active area	95.04(W)*53.86(H)	mm
Pixel size	198(W)*198(H)	um
Diagonal length	4.3	inch
Viewing direction	6 O'clock	-
Backlight	LED(white 7*LED)	-
Top & Tst	-20˚C - +70˚C & -30˚C - +80˚C	˚C
LCM: All of LCM of material and process measure up to ROHS Europe		

3. DIMENSIONAL OUTLINE



4. PIN DESCRIPTIO

No.	Symbol	F
1	BLKCON	BackLight Control.(H: BackLight On, L: BackLight Off)
2	RESET	RESET
3,18,19	GND	GND
4,20	5V	POWER Supply
6	/RD	TFT Read Signal(Low Active)
5	/WR	TFT Write Signal(Low Active)
7	/CS0	TFT Control Chip enable signal
8	D/C	TFT Data Or Command selection(H:Data L: Command)
9	NC	No USE
10-17	DB0-DB7	Data bus line (TFT)

5. MAXIMUM ABSOUTE LIMIT

Item	Symbol	MIN	MAX	Unit
Supply Voltage for Logic	VDD	-0.3	5.0	V
Input Voltage	Vin	-0.3	VDD+0.3	V
Supply Current Without Backllight	IDD(Ta = 25°C)		180	mA
Supply Current for Backlight	IF(Ta = 25°C)	---	280	mA
Reverse Voltage for Backlight	VR(Ta = 25°C)	---	5.5	V
Operating Temperature	Top	-20	70	
Storage Temperature	Tst	-30	80	

6. ELECTRICAL CHARACTERISTICS

Item	Symbol	Condition	Min	Typ	Max	Unit
Supply Voltage for Logic	V_{DD}-V_{SS}	Ta = 25°C	3.3.	5	5.5	V
Input High Voltage	V_{IH}	Ta = 25°C	08V $_{DD}$	---	V_{DD}	V
Input Low Voltage	V_{IL}	Ta = 25°C	0	---	02V $_{DD}$	V
Output High Voltage	V_{OH}	Ta = 25°C	08V $_{DD}$	---	V_{DD}	V
Output Low Voltage	V_{OL}		0	---	02V $_{DD}$	V
Supply Current Without Backllight	I_{DD}	Ta = 25°C	--	160	200	mA

7. BACKLIGHT CHARACTERISTICS

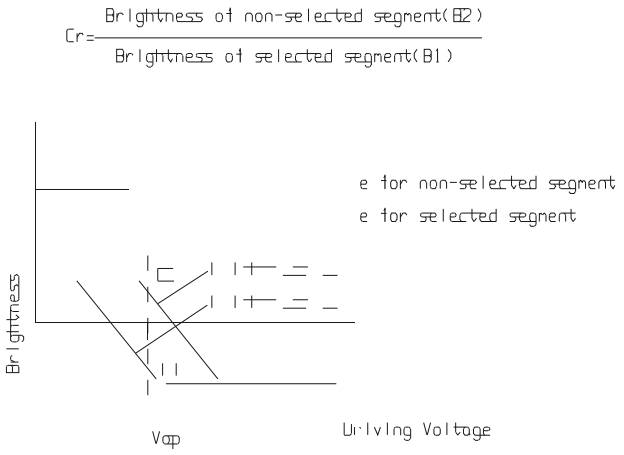
Ta = 250C

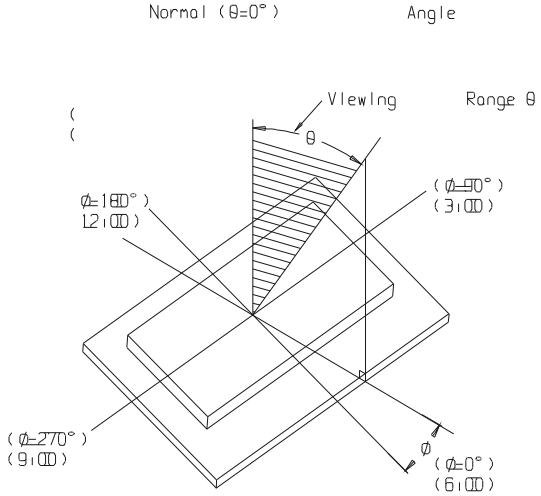
Item	Symbol	Condition	Min	Typ	Max	Unit
Led Control	VF	Led =5V	3.3	5	5.5	V
Peverse Current	IP	Vled=5V	---	160	200	mA
Luminous Intensity (With LCD dots off)	IV	Vled=5.0V	150	200	---	Cd/m2
LED Backlight Color	White					

8. ELECTRO-OPTICAL CHARACTERISTICS

(VDD=3.3V,Vled=5.0V, Ta = 250C)

Item	Symbol	Condition	Min	Typ	Max	Unit
Viewing angle (CP 10)	0L	=1800(9 o'clock)	60	70	---	degree
	0P	=00(3 o'clock)	60	70	---	
	0T	=900(12 o'clock)	40	50	---	
	0B	=2700(6 o'clock)	60	70	---	
Pesponse time	Ton	Normal 0= =00	---	10	20	ms
	Toff		---	15	30	ms
Contrast ratio	Cr		450	500	---	---
Luminance	L1		150	200		Cd/m2





9. CONTROL INSTRUCTION(Simple Description)

9.1 Instruction

Table 9-1: Instruction

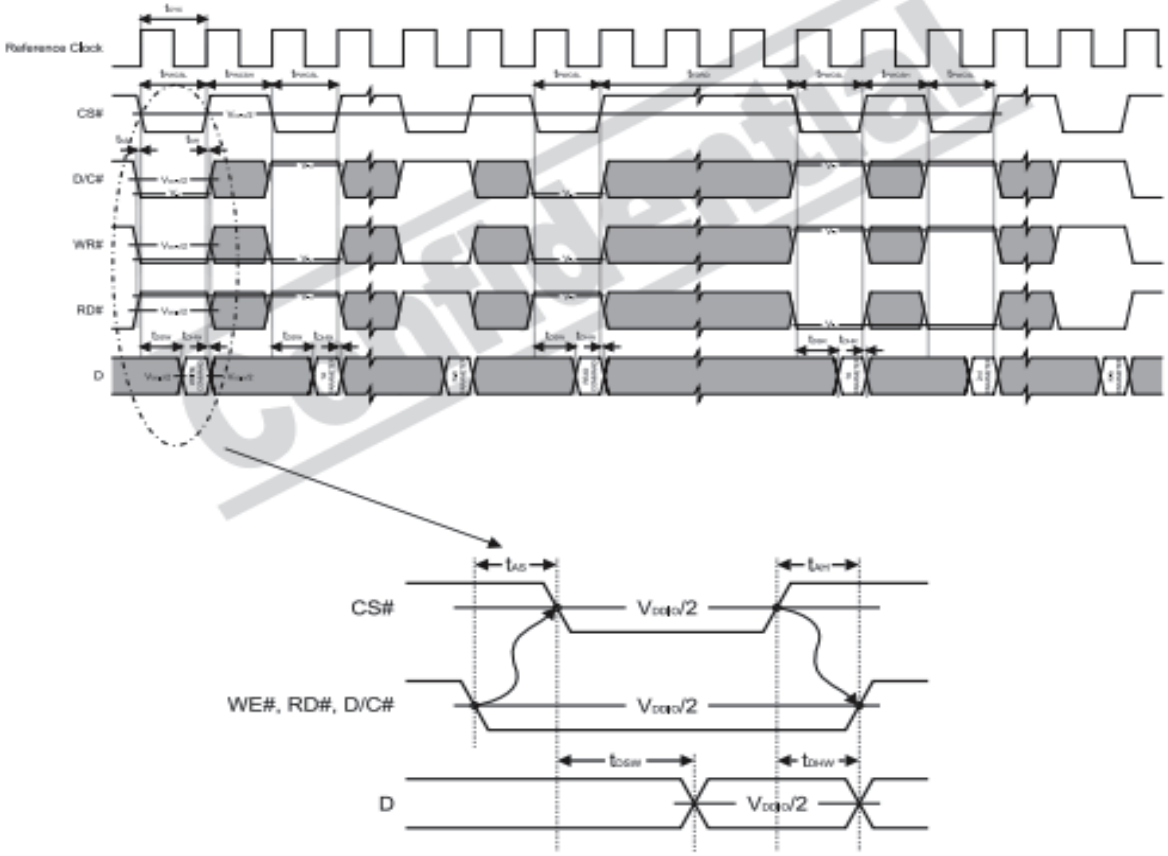
/CS0	D/C	/WR	/RD	DB0=DB7
H	H/L	H/L	H/L	Command non-active
L	L	L	H	Write Data
L	H	L	H	Write Command (Used DB0=DB7)
L	L	H	L	Command non-active
L	H	H	L	Read Data

9.2 8080 Mode Write Cycle

Table 9-2: 8080 Mode Timing

Symbol	Parameter	Min	Typ	Max	Unit
t_{cyc}	Reference Clock Cycle Time	9	-	-	ns
t_{PWCSL}	Pulse width CS# low	1	-	-	t_{CY}
t_{PWCSH}	Pulse width CS# high	1	-	-	t_{CY}
t_{FDRD}	First Read Data Delay	5	-	-	t_{CY}
t_{AS}	Address Setup Time	1	-	-	ns
t_{AH}	Address Hold Time	1	-	-	ns
t_{DSW}	Data Setup Time	4	-	-	ns
t_{DHW}	Data Hold Time	1	-	-	ns
t_{DSR}	Data Access Time	-	-	5	ns
t_{DHR}	Output Hold time	1	-	-	ns

Figure 9-3: 8080 Mode Timing Diagram



10. Reliability of LCM

Reliability test condition:

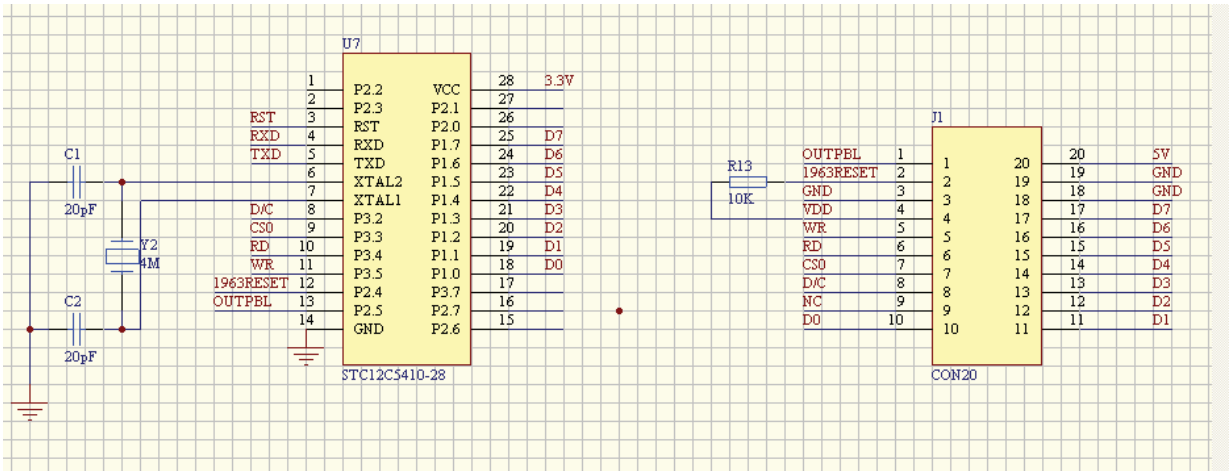
Item	Condition	Time (hrs)	Assessment
High temp. Storage	80°C	48	No abnormalities in functions and appearance
High temp. Operating	70°C	48	
Low temp. Storage	-30°C	48	
Low temp. Operating	-20°C	48	
Humidity	40°C! 90%RH	48	
Temp. Cycle	-20°C — 25°C 70°C (30 min — 5 min 30min)	**1**0cycles	

Recovery time should be 24 hours minimum. Moreover, functions, performance and appearance ,etc. shall be free from remarkable deterioration within 50,000 hours under ordinary operating and storage conditions room temperature (20 8°C), normal humidity (below 4 20% RH), and in the area not exposed to direct sun light. The life time is not content the life time of the LED (for the life time of

LED which decay only 50%,in the industry the experience value is 50000 hours, but there are not any experimentation data to support this). .

11. APPLICATION EXAMPLE

Figure 11-1: APPLICATION CIRCUIT FOR M7-BUS



12. PROGRAM EXAMPLE

(1) M7_Bus.h File

```
void ClearScreen(uint color);


void M7_BUSInit2();

void M7_BUSData_16(uchar sdataH,uchar sdataL);
void M7_BUSData1_16(uint dat);
void M7_BUSCommand_16(uint cmd);

void StartScroll(uint start);


void SetPixel(uint Col,uint Page,uint color); //显示一个相素点
uint GetPixel(uint Col,uint Page);//得到一个像素点
/*******************************************************
M7_BUS 的 128 脚，用来控制读写的方式，如果为 0 则为 6800 总线 MODE
如果位 1 ，则为 8080 总线。下面的定义是 8080 MODE
*******************************************************/

void M7_BUSCommand_8(uchar cmd);
void M7_BUSData_8(uchar sdata);

void M7_BUSInit();

uchar GetM7_BUSData_8();
uint GetM7_BUSData_16();

void ReadM7_BUSSet();

#define SetM7_BUSRD() SetBit(P3,4)
```

```
#define ClrM7_BUSRD() ClrBit(P3,4)


#define SetM7_BUSWR() SetBit(P3,5)
#define ClrM7_BUSWR() ClrBit(P3,5)

#define SetM7_BUSCS() SetBit(P3,3)
#define ClrM7_BUSCS() ClrBit(P3,3)

#define SetM7_BUSDC() SetBit(P3,2)
#define ClrM7_BUSDC() ClrBit(P3,2)

#define SetM7_BUSReset() SetBit(P2,4)
#define ClrM7_BUSReset() ClrBit(P2,4)


#define BLKOn()                 SetBit(P2,5)
#define BLKOff()    ClrBit(P2,5)

#define DataPortL  P1
```

(2) M7_Bus.C File

```
#include "includes.h"


//UWORD 是一个结构体
//注意在给 1963 写命令时，只能用数据线的低 8 位来传输数据。
void M7_BUSCommand_16(uint cmd)
{
        UWORD temp;
        temp.UInt=cmd;

        ClrM7_BUSDC();
        SetM7_BUSRD();

        ClrM7_BUSCS();
        ClrM7_BUSWR();
        DataPortL=temp.UByte[1];
    //  DataPortH=temp.UByte[0];

        SetM7_BUSWR();
        SetM7_BUSCS();


}

void M7_BUSData1_16(uint color)
{

        SetM7_BUSDC();
        SetM7_BUSRD();

        ClrM7_BUSCS();
        ClrM7_BUSWR();
```

```
                DataPortL=(uchar)((color>>8)&0x00f1);


                SetM7_BUSWR();
                ClrM7_BUSWR();
                DataPortL=(uchar)((color>>3)&0x00fc);


                SetM7_BUSWR();

                ClrM7_BUSWR();

                DataPortL=(uchar)((color<<3) & 0x00f1);


                SetM7_BUSWR();

                SetM7_BUSCS();



}

void M7_BUSCommand_8(uchar cmd)
{

                ClrM7_BUSDC();
                SetM7_BUSRD();

                ClrM7_BUSCS();
                ClrM7_BUSWR();
                DataPortL=cmd;
        ///     DataPortH=0X00;

                SetM7_BUSWR();
                SetM7_BUSCS();


}
void M7_BUSData_8(uchar sdata)
{
                SetM7_BUSDC();

                SetM7_BUSRD();

                ClrM7_BUSCS();
                ClrM7_BUSWR();
                DataPortL=sdata;
        //      DataPortH=0X00;

                SetM7_BUSWR();
                SetM7_BUSCS();


}
```

```
/*void M7_BUSData_16(uchar sdataL,uchar sdataH)
{
                    SetM7_BUSDC();
                    SetM7_BUSRD();
                    ClrM7_BUSWR();
                    ClrM7_BUSCS();

                    DataPortL=sdataL;
                    DataPortH=sdataH;
                    nop();nop();nop(); nop();nop();nop();
                    SetM7_BUSWR();
                    SetM7_BUSCS();

}*/
uchar GetM7_BUSData_8()
{
                    uchar sdata;

                    ClrM7_BUSRD();
                    SetM7_BUSWR();
                    SetM7_BUSDC();
                    ClrM7_BUSCS();
                    sdata=DataPortL;

                    nop();nop();nop();nop();nop();nop();
                    SetM7_BUSRD();
                    SetM7_BUSCS();
          return sdata;

}



uint GetM7_BUSData_16()
{
                    uint sdata;


                    sdata=0;
                    SetM7_BUSWR();
                    SetM7_BUSDC();
                    ClrM7_BUSCS();
                    ClrM7_BUSRD();
                    sdata|=((DataPortL&0xf1)<<8);
                    SetM7_BUSRD();

                    ClrM7_BUSRD();
                    sdata|=((DataPortL&0xfc)<<3);
                    SetM7_BUSRD();

                    ClrM7_BUSRD();
                    sdata|=((DataPortL&0xf1)>>3);
                    SetM7_BUSRD();

                    SetM7_BUSCS();
```

```
        return sdata;

}
uint GetPixel(uint Col,uint Page)
{

    uint temp;
            uint color;
    M7_BUSCommand_8(0x2a); //设置行开始和结束的位置
            M7_BUSData_8(Col/256);
            M7_BUSData_8(Col%256);
            temp=Col;
            M7_BUSData_8(temp/256);
            M7_BUSData_8(temp%256);

            M7_BUSCommand_8(0x2b);
            M7_BUSData_8(Page/256);
            M7_BUSData_8(Page%256);
            temp=Page;
            M7_BUSData_8(temp/256);
            M7_BUSData_8(temp%256);

            M7_BUSCommand_8(0x2E);
            color=GetM7_BUSData_16();
            return color;

}
void SetPixel(uint Col,uint Page,uint color)
{

    uint temp;

    M7_BUSCommand_8(0x2a); //设置行开始和结束的位置
            temp=Col;
            M7_BUSData_8(temp/256);
            M7_BUSData_8(temp%256);
            temp=Col;
            M7_BUSData_8(temp/256);
            M7_BUSData_8(temp%256);

            M7_BUSCommand_8(0x2b);
            temp=Page;
            M7_BUSData_8(temp/256);
            M7_BUSData_8(temp%256);
            temp=Page;
            M7_BUSData_8(temp/256);
            M7_BUSData_8(temp%256);

            M7_BUSCommand_8(0x2c);
           // for(i=0;i<9;i++)
           // {
            M7_BUSData1_16(color);
           // }
```

```
}
void ReadM7_BUSSet()
{



}
void M7_BUSInit()                    //320*240
{

        M7_BUSCommand_8(0x01);
        Delay_Us(400);
        Delay_Us(400);

        M7_BUSCommand_8(0x01);
        Delay_Us(400);
        Delay_Us(400);

        //enable pll
        M7_BUSCommand_8(0xe0);
        Delay_Us(400);
        M7_BUSData_8(0x01);
        Delay_Us(400);
        M7_BUSCommand_8(0xe0);
        Delay_Us(400);
        M7_BUSData_8(0x03);
        Delay_Us(400);

        //pll: n= 20 m= 2
        M7_BUSCommand_8(0xe2);
        M7_BUSData_8(0x14);
        M7_BUSData_8(0x02);
        M7_BUSData_8(0x04);  //set pll 100Mhz

        //dotclk setting
        M7_BUSCommand_8(0xe6);
        M7_BUSData_8(0x01);
        M7_BUSData_8(0x06);
        M7_BUSData_8(0x24); //pclk=6.4mhz

        //panel setting
        M7_BUSCommand_8(0xb0);
        Delay_Us(100);
        M7_BUSData_8(0x3c);
        M7_BUSData_8(0x00);
        M7_BUSData_8(0x01);

        M7_BUSData_8(0x3f);  //320
        M7_BUSData_8(0x00);
        M7_BUSData_8(0xef);  //240
        M7_BUSData_8(0x00);

        //pixel format
        M7_BUSCommand_8(0x3a);
        M7_BUSData_8(0x50);
```

```
//hsync setting
M7_BUSCommand_8(0xb4);
M7_BUSData_8(0x01);
M7_BUSData_8(0x50);  //TV
M7_BUSData_8(0x00);
M7_BUSData_8(0x06);  //THS+THb
M7_BUSData_8(0x04);  //Pulse Width
M7_BUSData_8(0x00);
M7_BUSData_8(0x00);
M7_BUSData_8(0x00);

//vsync setting
M7_BUSCommand_8(0xb6);
M7_BUSData_8(0x00);
M7_BUSData_8(0xF4);
M7_BUSData_8(0x00);
M7_BUSData_8(0x02); //TVS+TVB
M7_BUSData_8(0x02); //Pulse Width
M7_BUSData_8(0x00);
M7_BUSData_8(0x00);
M7_BUSData_8(0x00);


//col start_end
M7_BUSCommand_8(0x2a);
M7_BUSData_8(0x00);
M7_BUSData_8(0x00);
M7_BUSData_8(0x01);
M7_BUSData_8(0x3f);//320

//row start_end
M7_BUSCommand_8(0x2b);
M7_BUSData_8(0x00);
M7_BUSData_8(0x00);
M7_BUSData_8(0x00);
M7_BUSData_8(0xef);//240


 M7_BUSCommand_8(0x36);
 M7_BUSData_8(0x00);
//MCU IF setting

//M7_BUSCommand_8(13);


M7_BUSCommand_8(0xf0);
M7_BUSData_8(0x00);   //02


//M7_BUSCommand_8(0x35);

//display on
M7_BUSCommand_8(0x29);
```

```
	M7_BUSCommand_8(0xbe);//设置 PWM
	M7_BUSData_8(0X0E);
	M7_BUSData_8(0XFF);
	M7_BUSData_8(0X09);
	M7_BUSData_8(0XFF);
	M7_BUSData_8(0X00);
	M7_BUSData_8(0X00);


	M7_BUSCommand_8(0xD4);//(Define the threshold value)
	M7_BUSData_8(0X00);
	M7_BUSData_8(0X09);
	M7_BUSData_8(0X90);
	M7_BUSData_8(0X00);
	M7_BUSData_8(0X17);
	M7_BUSData_8(0XE8);
	M7_BUSData_8(0X00);
	M7_BUSData_8(0X39);
	M7_BUSData_8(0X60);

	M7_BUSCommand_8(0xd0);
	M7_BUSData_8(0x0d);
}
void M7_BUSInit2()                                    //480*272
{

	//software reset
	M7_BUSCommand_8(0x01);
	Delay_Us(400);
	Delay_Us(400);

	//enable pll
	M7_BUSCommand_8(0xe0);
	Delay_Us(400);
	M7_BUSData_8(0x01);
	Delay_Us(400);
	M7_BUSCommand_8(0xe0);
	Delay_Us(400);
	M7_BUSData_8(0x03);
	Delay_Us(400);

	//pll: n= 34 m= 3
	M7_BUSCommand_8(0xe2);
	M7_BUSData_8(0x14);
	M7_BUSData_8(0x03);
	M7_BUSData_8(0x54);

	//dotclk setting
	M7_BUSCommand_8(0xe6);
	M7_BUSData_8(0x03);
	M7_BUSData_8(0xa0);
	M7_BUSData_8(0x00);

	//panel setting
	M7_BUSCommand_8(0xb0);
```

```
Delay_Us(100);
M7_BUSData_8(0x38);
M7_BUSData_8(0x20);
M7_BUSData_8(0x01);

M7_BUSData_8(0xdf);
M7_BUSData_8(0x01);
M7_BUSData_8(0x0f);
M7_BUSData_8(0x00);

//pixel format
M7_BUSCommand_8(0x3a);
M7_BUSData_8(0x50);

//hsync setting
M7_BUSCommand_8(0xb4);
M7_BUSData_8(0x02);
M7_BUSData_8(0x0D);
M7_BUSData_8(0x00);
M7_BUSData_8(0x25);
M7_BUSData_8(0x02);
M7_BUSData_8(0x00);
M7_BUSData_8(0x00);

M7_BUSData_8(0x00);

//vsync setting
M7_BUSCommand_8(0xb6);
M7_BUSData_8(0x01);
M7_BUSData_8(0x1E);
M7_BUSData_8(0x00);
M7_BUSData_8(0x08);
M7_BUSData_8(0x01);
M7_BUSData_8(0x00);
M7_BUSData_8(0x00);

//col start_end
M7_BUSCommand_8(0x2a);
M7_BUSData_8(0x00);
M7_BUSData_8(0x00);
M7_BUSData_8(0x01);
M7_BUSData_8(0xdf);//480

//row start_end
M7_BUSCommand_8(0x2b);
M7_BUSData_8(0x00);
M7_BUSData_8(0x00);
M7_BUSData_8(0x01);
M7_BUSData_8(0x0f);//272


 M7_BUSCommand_8(0x36);
 M7_BUSData_8(0x00);
//MCU IF setting

//M7_BUSCommand_8(13);
```

```
        M7_BUSCommand_8(0xf0);
        M7_BUSData_8(0x03);   //02


        //M7_BUSCommand_8(0x35);

        //display on
        M7_BUSCommand_8(0x29);


        M7_BUSCommand_8(0xbe);//设置 PWM
        M7_BUSData_8(0X0E);
        M7_BUSData_8(0XFF);
        M7_BUSData_8(0X09);
        M7_BUSData_8(0XFF);
        M7_BUSData_8(0X00);
        M7_BUSData_8(0X00);


        M7_BUSCommand_8(0xD4);//(Define the threshold value)
        M7_BUSData_8(0X00);
        M7_BUSData_8(0X09);
        M7_BUSData_8(0X90);
        M7_BUSData_8(0X00);
        M7_BUSData_8(0X17);
        M7_BUSData_8(0XE8);
        M7_BUSData_8(0X00);
        M7_BUSData_8(0X39);
        M7_BUSData_8(0X60);

        M7_BUSCommand_8(0xd0);
        M7_BUSData_8(0x0d);
 //  ReadM7_BUSSet();

}
void StartScroll(uint start)
{
        M7_BUSCommand_8(0x37);
        M7_BUSData_8(start>>8);
        M7_BUSData_8(start);
}
void ClearScreen(uint color)
{
         long temp;
         UWORD temp1;


         //col start_end
        M7_BUSCommand_8(0x2a);
        M7_BUSData_8(0x00);
        M7_BUSData_8(0x00);
        M7_BUSData_8(0x01);
        M7_BUSData_8(0x3f);//320
```

```
        //row start_end
        M7_BUSCommand_8(0x2b);
        M7_BUSData_8(0x00);
        M7_BUSData_8(0x00);
        M7_BUSData_8(0x00);
        M7_BUSData_8(0xef);//240

         M7_BUSCommand_8(0x2c);


         temp1.UInt=color;
         SetM7_BUSDC();
         SetM7_BUSRD();
         ClrM7_BUSCS();
         for (temp=0;temp<76800;temp++)
         {
                        DataPortL=(uchar)((color>>8)&0x00f1);
                //      DataPortH=temp1.UByte[0];
                        ClrM7_BUSWR();
                        nop();
                        nop();
                        SetM7_BUSWR();

                        DataPortL=(uchar)((color>>3)&0x00fc);
                //      DataPortH=temp1.UByte[0];
                        ClrM7_BUSWR();
                        nop();
                        nop();
                        SetM7_BUSWR();
                        DataPortL=(uchar)((color<<3) & 0x00f1);

                //      DataPortH=temp1.UByte[0];
                        ClrM7_BUSWR();
                        nop();
                        nop();
                        SetM7_BUSWR();
                /*      if(ADCount(false))
                        {
                                break;
                        }                       */

                        if(ModeFlag==ReceiveMode)
                        {
                        break;
                        }

         }
         SetM7_BUSCS();
}
```

13.Precaution for using LCD/LCM

LCD!LCM is assembled and adjusted with a high degree of precision. Do not attempt to make any alteration or modification. The followings should be noted.

General Precautions:

1. LCD panel is made of glass. Avoid excessive mechanical shock or applying strong pressure onto the surface of display area.
2. The polarizer used on the display surface is easily scratched and damaged. Extreme care should be taken when handling. To clean dust or dirt off the display surface, wipe gently with cotton, or other soft material soaked with isoproply alcohol, ethyl alcohol or trichlorotriflorothane, do not use water, ketone or aromatics and never scrub hard.
3. Do not tamper in any way with the tabs on the metal frame.
4. Do not make any modification on the PCB without consulting Orient Display.
5. When mounting a LCM, make sure that the PCB is not under any stress such as bending or twisting. Elastomer contacts are very delicate and missing pixels could result from slight dislocation of any of the elements.
6. Avoid pressing on the metal bezel, otherwise the elastomer connector could be deformed and lose contact, resulting in missing pixels and also cause rainbow on the display.
7. Be careful not to touch or swallow liquid crystal that might leak from a damaged cell. Any liquid crystal adheres to skin or clothes, wash it off immediately with soap and water.

Static Electricity Precautions:

1. CMOS-LSI is used for the module circuit; therefore operators should be grounded whenever he!she comes into contact with the module.
2. Do not touch any of the conductive parts such as the LSI pads; the copper leads on the PCB and the interface terminals with any parts of the human body.
3. Do not touch the connection terminals of the display with bare hand; it will cause disconnection or defective insulation of terminals.
4. The modules should be kept in anti- static bags or other containers resistant to static for storage.
5. Only properly grounded soldering irons should be used.
6. If an electric screwdriver is used, it should be grounded and shielded to prevent sparks.
7. The normal static prevention measures should be observed for work clothes and working benches.
8. Since dry air is inductive to static, a relative humidity of 50-60% is recommended.

Soldering Precautions:

1. Soldering should be performed only on the I!O terminals.
2. Use soldering irons with proper grounding and no leakage.
3. Soldering temperature: 280°C±10°C
4. Soldering time: 3 to 4 second.
5. Use eutectic solder with resin flux filling.
6. If flux is used, the LCD surface should be protected to avoid spattering flux.
7. Flux residue should be removed.

Operation Precautions:

1. The viewing angle can be adjusted by varying the LCD driving voltage Vo.
2. Since applied DC voltage causes electro-chemical reactions, which deteriorate the display, the applied pulse waveform should be a symmetric waveform such that no DC component remains. Be sure to use the specified operating voltage.
3. Driving voltage should be kept within specified range; excess voltage will shorten display life.
4. Response time increases with decrease in temperature.
5. Display color may be affected at temperatures above its operational range.
6. Keep the temperature within the specified range usage and storage. Excessive temperature and humidity could cause polarization degradation, polarizer peel-off or generate bubbles.
7. For long-term storage over 40°C is required, the relative humidity should be kept below 60%, and avoid direct sunlight.

Limited Warranty

Orient Display LCDs and modules are not consumer products, but may be incorporated by Orient Display's customers into consumer products or components thereof, Orient Display does not warrant that its LCDs and components are fit for any such particular purpose.

1. The liability of Orient Display is limited to repair or replacement on the terms set forth below. Orient Display will not be responsible for any subsequent or consequential events or injury or damage to any personnel or user including third party personnel and/or user. Unless otherwise agreed in writing between Orient Display and the customer. Orient Display will only replace or repair any of its LCD which is found defective electrically or visually when inspected in accordance with Orient Display general LCD inspection standard (copies available on request).
2. No warranty can be granted if any of the precautions state in handling liquid crystal display above has been disregarded. Broken glass, scratches on polarizer mechanical damages as well as defects that are caused accelerated environment tests are excluded from warranty.
3. In returning the LCD/LCM, they must be properly packaged; there should be detailed description of the failures or defect.