晶采光電科技股份有限公司
AMPIRE CO., LTD.

SPECIFICATIONS FOR LCD MODULE

CUSTOMER	
CUSTOMER PART NO.	
AMPIRE PART NO.	AM-480272H3TMQW-TW7H
APPROVED BY	
DATE	

☐ Approved For Specifications
☐ Approved For Specifications & Sample

AMPIRE CO., LTD.
2F., No.88, Sec. 1, Sintai 5th Rd., Sijhih City, Taipei County 221, Taiwan (R.O.C.)台北縣汐止市新台五路一段88號2樓(東方科學園區D棟)
TEL:886-2-26967269 , FAX:886-2-26967196 or 26967270

APPROVED BY	CHECKED BY	ORGANIZED BY

RECORD OF REVISION

Revision Date	Page	Contents	Editor
2010/10/21	--	New Release	John
2011/03/31	8	Add PWM Frequency and Voltage	Kevin
2011/03/31	5	Add TP life time	Kevin
2011/06/22	5-15	Add TP sample code	Kevin
2012/01/17	19	Add LED life time	Kevin
		Modify life time	Kain
2014/1/14	21	Add LCD Driver Timing and LCD Controller initial code	Rober
	43		
2014/2/6	3	Correct the Physical specifications	Rober
	17	Correct the Absolute max. ratings	
	21	Correct the TFT LCD Driver input timing requirement	
2015/09/04		Remove IIS	Lawlite

1 Features

4.3 inch Amorphous-TFT-LCD (Thin Film Transistor Liquid Crystal Display) module. This module is composed of a 4.3” TFT-LCD panel, LCD controller, power driver circuit,

touch panel and backlight unit.

1.1 TFT Panel Feature :

(1) Construction: 4.3” a-Si color TFT-LCD, White LED Backlight and PCB.
(2) Resolution (pixel): 480(R.G.B) X 272
(3) Number of the Colors : 262K colors (R , G , B 6 bit digital each)
(4) LCD type : Transmissive Color TFT LCD (normally White)
(5) Interface: 40 pin pitch 0.5
(6) Power Supply Voltage: 3.3V single power input. Built-in power supply circuit.
(7) Viewing Direction: 6 O’clock (The direction it’s hard to be discolored):

1.2 LCD Controller Feature:

(1) MCU interface 8/9/16/18 bit 80&68 series MCU interface.
(2) Display RAM size : 640x320x3x6 bits. Ex : 320x240 two frame buffer with 262K colors.
(3) Arbitrary display memory start position selection.
(4) MCU interface : 8 bit / 9 bit / 16bit / 18 bits 80/68 MPU interface. **Default : i80-16Bit.**
(5) 8 bit / 16 bit interface support 65K (R5G6B5) /262K(R6G6B6) colors data format.
(6) 9 bit / 18 bit interface support 262K(R6G6B6) colors data format only.

2 Physical specifications

Item	Specifications	Unit
Display resolution(dot)	480(R.G.B.) (W) x 272(H)	mm
Active area	95.04 (W) x 53.856 (H)	mm
Screen size	4.3 (Diagonal)	mm
Pixel size	0.198 (W) x 0.198 (H)	um
Color configuration	R.G.B stripe	
Overall dimension	105.5(W) x 67.2(H) x 7.61(D)	mm
Weight	95±3	g
Backlight unit	LED	

3 Default Setting & Option

- Interface :

The user can select the MCU interface by change the Jumper & Resister Array.

Setting	JP1	RA1	RA2	RA3	RA4	Remark

Interface Type						
80-18Bit interface	1,2 short 2,3 open	2K ohm	OPEN	OPEN	OPEN	
80-16Bit interface	**1,2 short 2,3 open**	**OPEN**	**2K ohm**	**OPEN**	**OPEN**	**Default**
80-9Bit interface	1,2 short 2,3 open	OPEN	OPEN	2K ohm	OPEN	
80-8Bit interface	1,2 short 2,3 open	OPEN	OPEN	OPEN	2K ohm	
68-18Bit interface	1,2 open 2,3 short	2K ohm	OPEN	OPEN	OPEN	
68-16Bit interface	1,2 open 2,3 short	OPEN	2K ohm	OPEN	OPEN	
68-9Bit interface	1,2 open 2,3 short	OPEN	OPEN	2K ohm	OPEN	
68-8Bit interface	1,2 open 2,3 short	OPEN	OPEN	OPEN	2K ohm	

- LED Driver:

The user can select the LED driver built-in or not.

Pin Define / Interface Type	PIN3 LEDA/PWM	PIN4 LEDK	Remark
Without LED Driver	LED Anode	LED Cathode	
With LED Driver	**PWM** The PWM pin combined enable and brightness adjust function. When PWM=High constantly, the LED back-light is turn on. When PWM=GND constantly, the LED back-light is turn off. When PWM signal (100Hz to 1KHz) input, the LED Back-light brightness is relative to duty cycle of the PWM signal.	**NC** This pin must be open	**Default**

- Touch panel and Touch panel controller:

The user can select the with TP controller or without TP controller.

Pin Define / Option	SK/X1	DO/X2	DI/Y1	TPCS/Y2	IRQ	Remark
Without TP	NC	NC	NC	NC	NC	

With TP / Without TP controller	X1	X2	Y1	Y2	NC	
With TP / With TP controller	**SK**	**DO**	**DI**	**TPCS**	**IRQ**	**Default**

If user want to change the default setting for mass production, please contact with Ampire. We'll apply a new P/N for you.

- Touch panel Mechanical characteristics

Mechanical characteristics	Min	Note
Notes Life	10^5	Within guaranteed active area
Input Life	10^6	Within guaranteed active area

- Touch controller sample code

```
TSC 2046 T/P SAMPLE CODE
#include "FSA506.h" // Type define
#include "TSC2046.h"
#define ShowTP_XY 0 //0: don't show
static uint8 Get_Start,Get_END,Dat16bit;
static uint32 TP_X,TP_Y;
static uint32 TP_X_Start,TP_Y_Start,TP_X_End,TP_Y_End,TP_X_All,TP_Y_All=0;
static uint8 TP_flag=0;
void Delay_NOP( uint32 C)
{
uint32 i,j;
for(j=0;j<10;j++)
{
for (i=0;i<C;i++)
{
}
}
}
void Start_Arrow(int Dat16bit)
{
uint32 i,j;
for (j=0;j<16;j++)
{
for (i=0;i<16;i++)
{
if(j==0 & (i==0 | i==1))
```

```
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==1 & (i==0 | i==1 | i==2 | i==3 | i==4))
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==2 & (i==1 | i==2 | i==5 | i==6))
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==3 & (i==1 | i==3 | i==7 | i==8))
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==4 & (i==2|i==4))
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==5 & (i==2 | i==5))
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==6 & (i==3|i==6))
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==7 & (i==3| i==7))
{
LCD_Pixel(i,j,Dat16bit);
}
if(i==j)
{
LCD_Pixel(i,j,Dat16bit);
}
}
}
}
}
void END_Arrow(int Dat16bit)
```

```
{
uint32 i,j;
uint16 X,Y;
X=Resolution_X;Y=Resolution_Y;
for (j=(Y-16);j<Y;j++)
{
for (i=(X-16);i<X;i++)
{
if((j==(Y-1)) & (i==(X-2) | i==(X-1)))
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==(Y-2) & (i==(X-1) | i==(X-2) | i==(X-3) | i==(X-4)) )
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==(Y-3) & (i==(X-2) | i==(X-3) | i==(X-5) | i==(X-6)))
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==(Y-4) & (i==(X-2) | i==(X-4) | i==(X-7) | i==(X-8)))
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==(Y-5) & (i==(X-2)|i==(X-5)))
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==(Y-6) & i==(X-3) )
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==(Y-7) & i==(X-3))
{
LCD_Pixel(i,j,Dat16bit);
}
if(j==(Y-8) & (i==(X-4)| i==(X-4)))
{
LCD_Pixel(i,j,Dat16bit);
```

```
}
if((i-(X-20))==(j-(Y-20)))
{
LCD_Pixel(i,j,Dat16bit);
}
}
}
}
void SET_TC2064_IO(void)
{
GPIO_InitTypeDef GPIO_InitStructure;
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_12 ;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOB, &GPIO_InitStructure);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13 |GPIO_Pin_15 ;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_OD;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOB, &GPIO_InitStructure);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_14;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;
//GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOB, &GPIO_InitStructure);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;
//GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOA, &GPIO_InitStructure);
}
uint16 Init2046(char SPI_ID) //input: SPI_ID; output: Dat16bit
{
uint16 SPIData,i,j,Data,Dat16bit;
uint16 ReadBit;
CLR_DCLK;
CLR_TPCS;
Data=0x0000;
```

```
for(i=0;i<8;i++) //clock 1~8
{
if(SPI_ID & 0x80)
{
SET_DI;
}
else
{
CLR_DI;
}
CLR_DCLK;
Delay_NOP(1);
SET_DCLK;
Delay_NOP(1);
SPI_ID<<=1;
}
CLR_DCLK; //clock 9
Delay_NOP(1);
SET_DCLK;
Delay_NOP(1);;
SPIData=0;
for(j=0;j<12;j++) //clock10~21
{
CLR_DCLK;
Delay_NOP(1);
SET_DCLK;
Delay_NOP(1);
if((ReadDI(ReadBit))>1)
{
SPIData=(SPIData<<1)+1;}
else
{SPIData=(SPIData<<1);}
}
for(i=0;i<3;i++) //clock 22~24
{
CLR_DCLK;
Delay_NOP(1);
SET_DCLK;
Delay_NOP(1);
```

```
}
SET_DCLK;
SET_TPCS;
Dat16bit = SPIData;
return Dat16bit;
}
uint16 MeasureTemp(void)
{
char x,X_Add_8bit;
uint16 TempX,Dat16bit;
for (x=0;x<1;x++)
{
TempX=0xffff;
X_Add_8bit=0x00;
//while(TempX!=X_Add_8bit)
while(TempX!=Dat16bit)
{
// TempX=X_Add_8bit;
TempX=Dat16bit;
TP_X=Init2046(TSC2046_Temp);
if(TPX_Mirro)
{
TP_X=0x0FFF-TP_X;
}
Dat16bit=TP_X;
X_Add_8bit=(Dat16bit&0x0ff0)>>4;
}
//ShowChar16bit(0x20,0,X_Add_8bit);
Print_DAT(0xE0,0,0,Dat16bit);
}
return Dat16bit;
}
uint16 MeasureX(void)
{
uint8 x,X_Add_8bit;
uint16 TempX,Dat16bit;
for (x=0;x<1;x++)
{
TempX=0xffff;
```

```
X_Add_8bit=0x00;
while(TempX!=Dat16bit)
{
TempX=Dat16bit;
TP_X=Init2046(TSC2046_X);
if(TPX_Mirro)
{
TP_X=0x0FFF-TP_X;
}
Dat16bit=TP_X;
X_Add_8bit=(Dat16bit&0x0ff0)>>4;
}
Print_DAT(0x20,0,0,Dat16bit);
}
return Dat16bit;
}
uint16 MeasureY(void)
{
uint8 x,Y_Add_8bit;
uint16 TempY,Dat16bit;
for (x=0;x<1;x++)
{
TempY=0xffff;
Y_Add_8bit=0x0000;
while(TempY!=Dat16bit)
{
TempY=Dat16bit;
TP_Y=Init2046(TSC2046_Y);
if(TPY_Mirro)
{
TP_Y=0x0FFF-TP_Y;
}
Dat16bit=TP_Y;
Y_Add_8bit=(Dat16bit&0x0ff0)>>4;
}
Print_DAT(0x80,0,0,Dat16bit);
}
return Dat16bit;
}
```

```
void Touch_IRQ_Enable ()
{
EXTI_InitTypeDef EXTI_InitStructure;
GPIO_InitTypeDef GPIO_InitStructure;
NVIC_InitTypeDef NVIC_InitStructure;
/* Configure PA.03 as input floating */
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
GPIO_Init(GPIOA, &GPIO_InitStructure);
//GPIO_EXTILineConfig(GPIO_PortSourceGPIOA, GPIO_PinSource3);
/* Configure one bit for preemption priority */
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);
/* Enable the EXTI9_5 Interrupt */
NVIC_InitStructure.NVIC_IRQChannel = EXTI3_IRQChannel;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);
/* Connect EXTI Line9 to PB.09 */
GPIO_EXTILineConfig(GPIO_PortSourceGPIOA, GPIO_PinSource3);
/* Configure EXTI Line9 to generate an interrupt on falling edge */
EXTI_InitStructure.EXTI_Line = EXTI_Line3;
EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling;
EXTI_InitStructure.EXTI_LineCmd = ENABLE;
EXTI_Init(&EXTI_InitStructure);
/* Generate software interrupt: simulate a falling edge applied on EXTI line 9 */
//EXTI_GenerateSWInterrupt(EXTI_Line3);
}
/*******************************************************************************
***********
* Function Name : EXTI3_IRQHandler
* Description : This function handles External interrupt Line 3 request.
* Input : None
* Output : None
* Return : None
*******************************************************************************
**********/
void EXTI3_IRQHandler(void)
```

```
{
EXTI_InitTypeDef EXTI_InitStructure;
GPIO_InitTypeDef GPIO_InitStructure;
NVIC_InitTypeDef NVIC_InitStructure;
uint32 LLCD_X,LLCD_Y;
uint16 color;
EXTI_DeInit(); // Stop INT
TP_X=MeasureX();
TP_Y=MeasureY();
// Delay_mS(100);
while((ReadINT())==0)
{
color=(rand() % 0xffff);
//FontColor=color;
TP_X=MeasureX();
TP_Y=MeasureY();
//Delay_uS(200);
//Full_LCD(0x007e0);
LLCD_X=TP_X-TP_X_Start;
LLCD_X*=Resolution_X;
LLCD_X/=TP_X_All;
LLCD_Y=TP_Y-TP_Y_Start;
LLCD_Y*=Resolution_Y;
LLCD_Y/=TP_Y_All;
LCD_Pixel(LLCD_X,LLCD_Y,color);
}
/* */
// Enable INT
Touch_IRQ_Enable();
}
void Touch_initial(void)
{
uint16 TP_X_Min,TP_X_Max,TP_Y_Min,TP_Y_Max;
char Touchflag;
SET_TC2064_IO();
// Init2046(0xd1);
Full_LCD(0xffff);
Print_String(0,Resolution_Y/2,0,"Touch panel calibration");
Print_String(0,16+(Resolution_Y/2),0,"STEP1: Touch the LCD(0,0)");
```

```
Get_Start=0;
Get_END=0;
while (Get_Start==0)
{
Touchflag=1;
TP_X_Min=0x0020;
TP_X_Max=0x01f0;
TP_Y_Min=0x0020;
TP_Y_Max=0x01f0;
Start_Arrow(0xf800);
Init2046(0xd0);
while(Touchflag==1)
{
Start_Arrow(0xf800);
while((ReadINT())==0)
{
TP_X=MeasureX();
TP_Y=MeasureY();
Delay(0);
Start_Arrow(0xf800);
if (TP_X<TP_X_Max & TP_X>TP_X_Min & TP_Y < TP_Y_Max &
TP_Y > TP_Y_Min)
{
TP_X_Start=TP_X;
TP_Y_Start=TP_Y;
Touchflag=0;
Start_Arrow(0xffff);
}
else Start_Arrow(0x007f);
}
}
Start_Arrow(0xffff);
Get_Start=1;
}
Print_String(0,16+(Resolution_Y/2),0,"STEP2: Touch the
LCD(End_X,End_Y)");
while (Get_END==0)
{
Touchflag=1;
```

```
TP_X_Min=0x0b00;
TP_X_Max=0x0fe0;
TP_Y_Min=0x0b00;
TP_Y_Max=0x0fe0;
END_Arrow(0xf800);
Init2046(0xd0);
while(Touchflag==1)
{
END_Arrow(0xf800);
while((ReadINT())==0)
{
TP_X=MeasureX();
TP_Y=MeasureY();
Delay(0);
END_Arrow(0xf800);
if (TP_X<TP_X_Max & TP_X>TP_X_Min & TP_Y < TP_Y_Max &
TP_Y > TP_Y_Min)
{
TP_X_End=TP_X;
TP_Y_End=TP_Y;
Touchflag=0;
END_Arrow(0xffff);
}
else END_Arrow(0x007f);
}
}
END_Arrow(0xffff);
Get_END=1;
TP_X_All=TP_X_End-TP_X_Start;
TP_Y_All=TP_Y_End-TP_Y_Start;
}
Touch_IRQ_Enable();
Print_String(0,32+(Resolution_Y/2),0,"Touch panel calibration OK");
TP_X=0x0001;
TP_Y=0x0001;
TP_flag=0xff;
}
```

4 Electrical specification

4.1 Absolute max. ratings

4.1.1 Electrical Absolute max. ratings

Item	Symbol	Condition	Min.	Max.	Unit	Remark
Power voltage	VDD	VSS=0	-0.3	3.6	V	

Input voltege	V_{in}		-0.3	VDD+0.3	V	Note 1

Note1: /CS,/WR,/RD,RS,DB0~DB17

4.1.2 Environmental Absolute max. ratings

Item	OPERATING		STORAGE		Remark
	MIN	MAX	MIN	MAX	
Temperature	-20	70	-30	80	Note2,3,4,5,6,7,8
Humidity	Note1		Note1		
Corrosive Gas	Not Acceptable		Not Acceptable		

Note1 : Ta <= 40℃ : 85% RH max

Ta > 40℃ : Absolute humidity must be lower than the humidity of 85%RH at 40℃

Note2 : For storage condition Ta at -30℃ < 48h , at 80℃ < 100h

For operating condition Ta at -20℃ < 100h

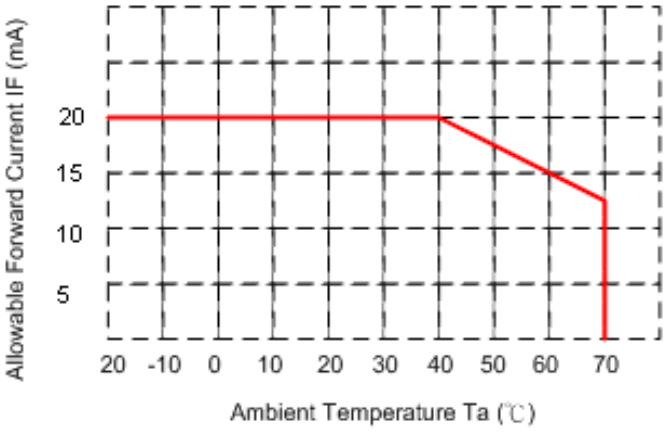
Note3 : Background color changes slightly depending on ambient temperature. This phenomenon is reversible.

Note4 : The response time will be slower at low temperature.

Note5 : Only operation is guarantied at operating temperature. Contrast , response time, another display quality are evaluated at +25℃

Note6 :

- LED BL : When LCM is operated over 40℃ ambient temperature, the I_{LED} of the LED back-light should be follow :



Note7 : This is panel surface temperature, not ambient temperature.

Note8 :

- LED BL:When LCM be operated over than 40℃ , the life time of the LED back-light will be reduced.

.

4.1.3 LED back-light Unit Absolute max. ratings

Item	Symbol	Ratings	Unit	Remark
Pulse Forward Current	IF	100	mA	
Forward Current	IF	30	mA	
Reverse Voltage	VR	35	V	
Power Dissipation	Po	0.84	W	

4.2 Electrical characteristics

4.2.1 DC Electrical characteristic of the LCD

Typical operating conditions (VSS=0V)

Item		Symbol	Min.	Typ.	Max.	Unit	Remark
Power supply		VDD	3.0	3.3	5.0	V	
Input Voltage	H Level	V_{IH}	2.0	-	5.5	V	Note 1

for logic	L Level	V_{IL}	VSS	-	0.8	V	
Output Voltage for Logic	H Level	V_{OH}	2.4	-	VDD	V	Note 2
	L Level	V_{OL}	VSS		0.4	V	
Power Supply current		IDD	-	450	-	mA	Note 3
PWM Frequency		PWM_F	100	-	1K	Hz	
PWM Voltage		PWM_V	3	5	6	V	

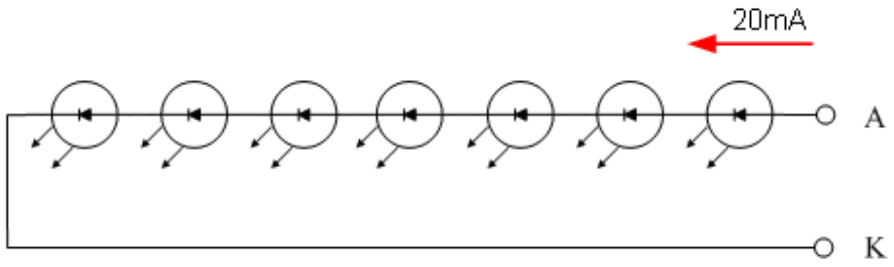
Note1: With 5V Tolerance Input , /CS, /WR,/RD,RS,DB0~DB17

Note2: DB0~DB17

Note3: f_V =60Hz , Ta=25℃ , Display pattern : All Black

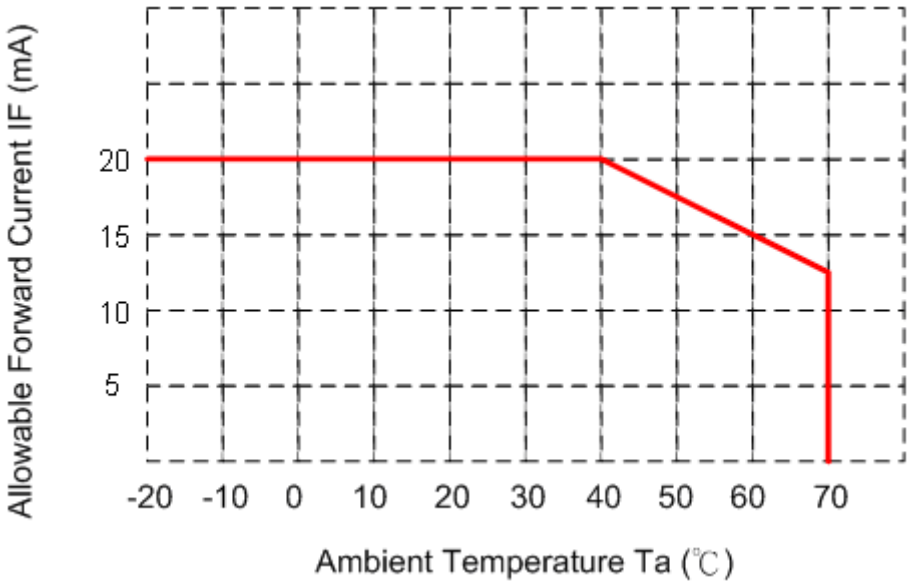
4.2.2 Electrical characteristic of LED Back-light

Paramenter	Symbol	Min.	Typ.	Max.	Unit	Condiction
LED voltage	V_{AK}	--	23.1	--	V	I_{LED} =20mA,Ta=25℃
LED forward current	I_{LED}	--	20	--	mA	Ta=25℃
	I_{LED}	--	15	--	mA	Ta=60℃
LED life time	-	20	--	--	Khr	I_{LED} =20mA,Ta=25℃



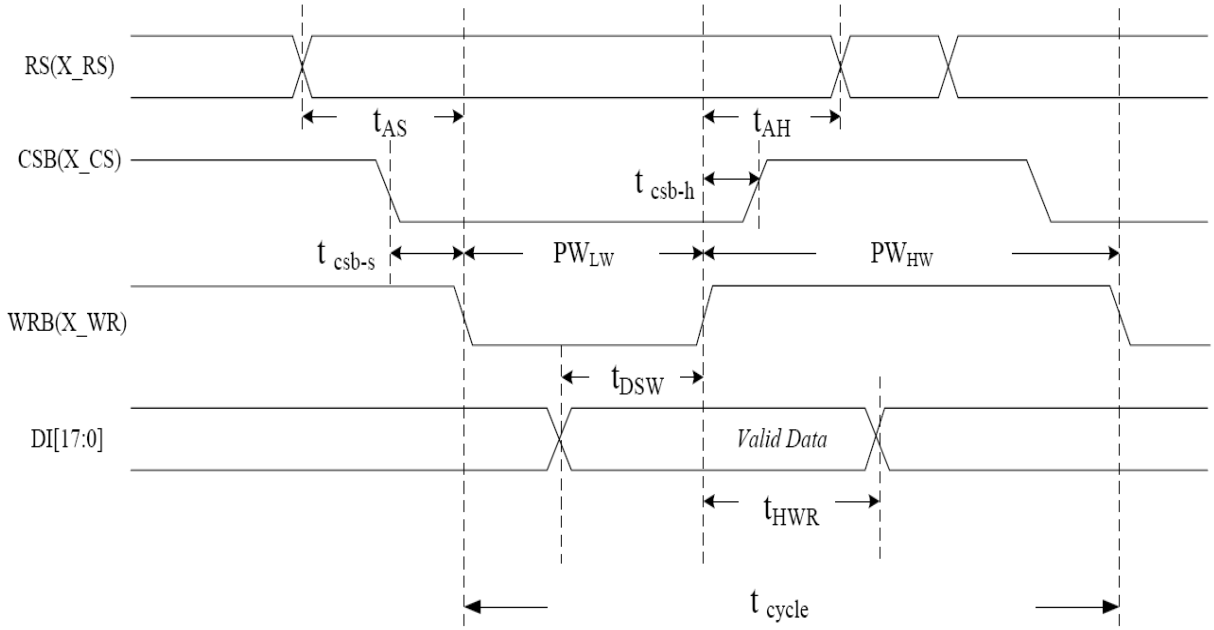
■ The constant current source is needed for white LED back-light driving.

When LCM is operated over 60℃ ambient temperature, the I_{LED} of the LED back-light should be adjusted to 15 mA.



4.3 AC Timing characteristic of the Graphic TFT LCD controller

4.3.1 80 series Timing



Symbol	Parameter	Min	Typ	Max	Unit	Remark
t_{cycle}	Enable cycle time	100	200		ns	
PW_{HW}	Enable high-level pulse width	66	70		ns	
PW_{LW}	Enable low-level pulse width	33	130		ns	

t_{AS}	RS setup time	16	25		ns	
t_{AH}	RS hold time	16	45		ns	
t_{DSW}	Write data setup time	50	50		ns	
t_{HWR}	Write data hold time	40	50		ns	
t_{csb-s}	CSB setup time	16	20		ns	
t_{csb-h}	CSB hold time	16	30		ns	

4.4 TFT LCD Driver input timing requirement

Parameter	Symbol	Spec.			Unit
		Min.	Typ.	Max.	
Clock cycle	f_{CLK} (1)	-	9	15	MHz
Hsync cycle	1/th	-	17.14	-	KHz
Vsync cycle	1/tv	-	59.94	-	Hz
Horizontal Signal					
Horizontal cycle	th	525	525	605	CLK
Horizontal display period	thd	480	480	480	CLK
Horizontal front porch	thf	2	2	82	CLK
Horizontal pulse width	thp (2)	2	41	41	CLK
Horizontal back porch	thb (2)	2	2	41	CLK
Vertical Signal					
Vertical cycle	tv	285	286	511	H (1)
Vertical display period	tvd	272	272	272	H (1)
Vertical front porch	tvf	1	2	227	H (1)
Vertical pulse width	tvp (2)	1	10	11	H (1)
Vertical back porch	tvb (2)	1	2	11	H (1)

Note: (1) Unit: CLK=1/ f_{CLK} , H=th,

(2)It is necessary to keep tvp+tvb=12 and thp+thb=43 in sync mode. DE mode is unnecessary to keep it.

(3) Clock cycle recommended setting as 12M Hz , the MAX Clock cycle can't over 15M Hz

5 Optical specification

5.1 Optical characteristic :

Item		Symbol	Conditon	Min.	Typ.	Max.	Unit	Remark
Response Time	Rise	T_r	Θ=0°	--	15	20	ms	Note 1,2,3,5
	Fall	T_f		--	35	50		
Contrast ratio		CR	At optimized viewing angle	150	250	--		Note 1,2,4,5
Viewing Angle	Top		CR≧10	--	55	--	deg.	Note1,2, 5,6
	Bottom			--	35	--		
	Left			--	70	--		
	Right			--	70	--		
Brightness LED BL Without TP		Y_L	I_{LED}=20mA, 25℃		500	--	cd/m²	Note 7
Brightness LED BL With TP		Y_L	I_{LED}=20mA, 25℃		400	--	cd/m²	Note 7
Red chromaticity		X_R	Θ=0°	(0.585)	(0.615)	(0.645)		Note 7 For reference only. These data should be update according the prototype.
		Y_R		(0.314)	(0.344)	(0.374)		
Green chromaticity		X_G		(0.277)	(0.307)	(0.337)		
		Y_G		(0.532)	(0.562)	(0.592)		
Blue chromaticity		X_B		(0.103)	(0.133)	(0.163)		
		Y_B		(0.120)	(0.150)	(0.180)		
White chromaticity		X_W		(0.279)	(0.309)	(0.339)		
		Y_W		(0.320)	(0.350)	(0.380)		

( )For reference only. These data should be update according the prototype.

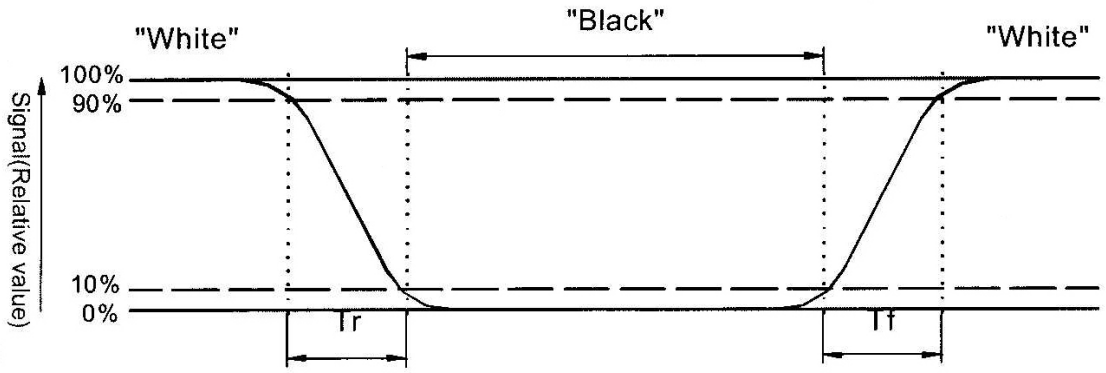
Note 1:

- LED BL : Ambient temperature=25℃,and lamp current I_{LED}=20mA.To be measured in the dark room.

Note 2:To be measured on the center area of panel with a viewing cone of 1°by Topcon luminance meter BM-7,after 10 minutes operation.

Note 3.Definition of response time:

The output signals of photo detector are measured when the input signals are changed from “black“ to “white”(falling time) and from ”white” to “black” (rising time),respectively. The response time is defined as the time interval between the 10% and 90% of amplitudes. Refer to figure as below.



Note 4.Definition of contrast ratio:

Contrast ratio is calculated with the following formula.

$$\text{Contrast ratio(CR)} = \frac{\text{Photo detector output when LCD is at "White" state}}{\text{Photo detector Output when LCD is at "Black" state}}$$

Note 5:White $V_i = V_{i50}$ +1.5V

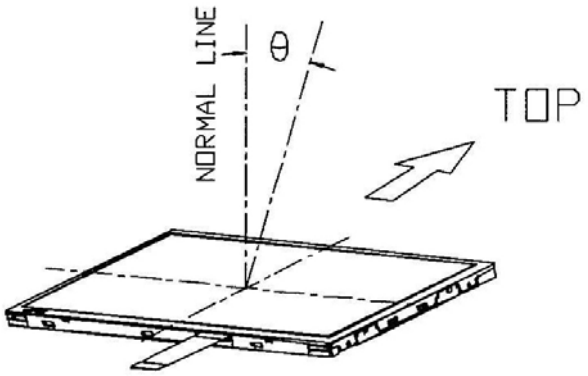
Black $V_i = V_{i50}$ +2.0V

"±"means that the analog input signal swings in phase with V_{COM} signal.

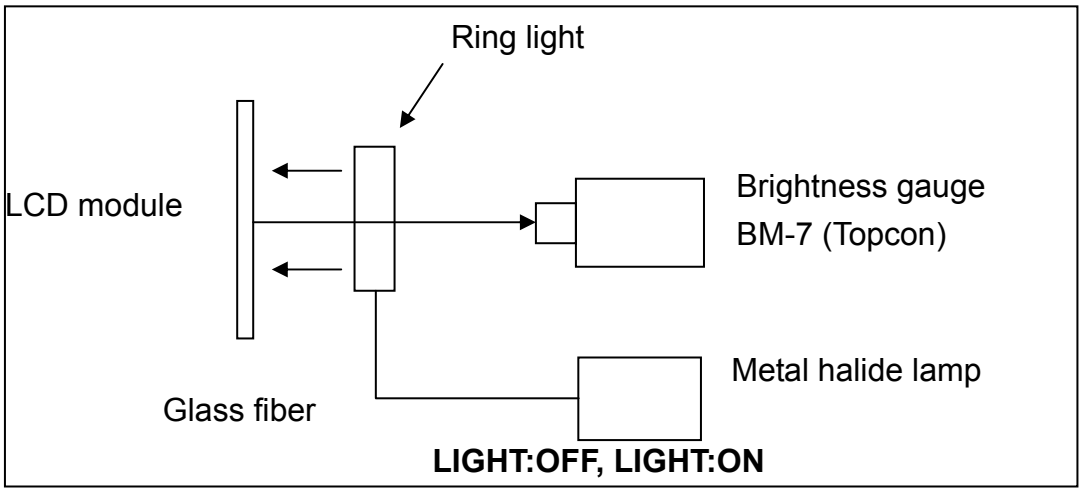
"$\frac{-}{+}$ " means that the analog input signal swings out of phase with V_{COM} signal.

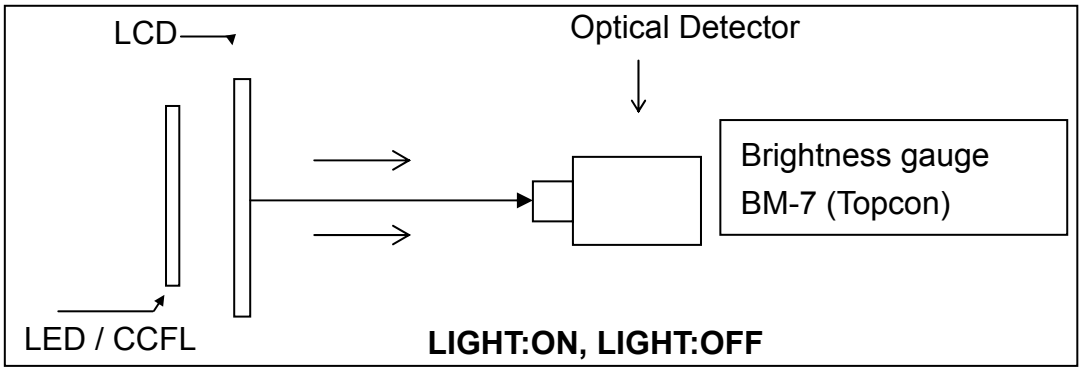
V_{i50} : The analog input voltage when transmission is 50%.The 100% Transmission is defined as the transmission of LCD panel when all the Input terminals of module are electrically opened.

Note 6.Definition of viewing angle, Refer to figure as below.



Note 7.Measured at the center area of the panel when all the input terminals of LCD panel are electrically opened.


Ring light
LCD module
Brightness gauge
BM-7 (Topcon)
Metal halide lamp
Glass fiber
LIGHT:OFF, LIGHT:ON


LCD
Optical Detector
Brightness gauge
BM-7 (Topcon)
LED / CCFL
LIGHT:ON, LIGHT:OFF

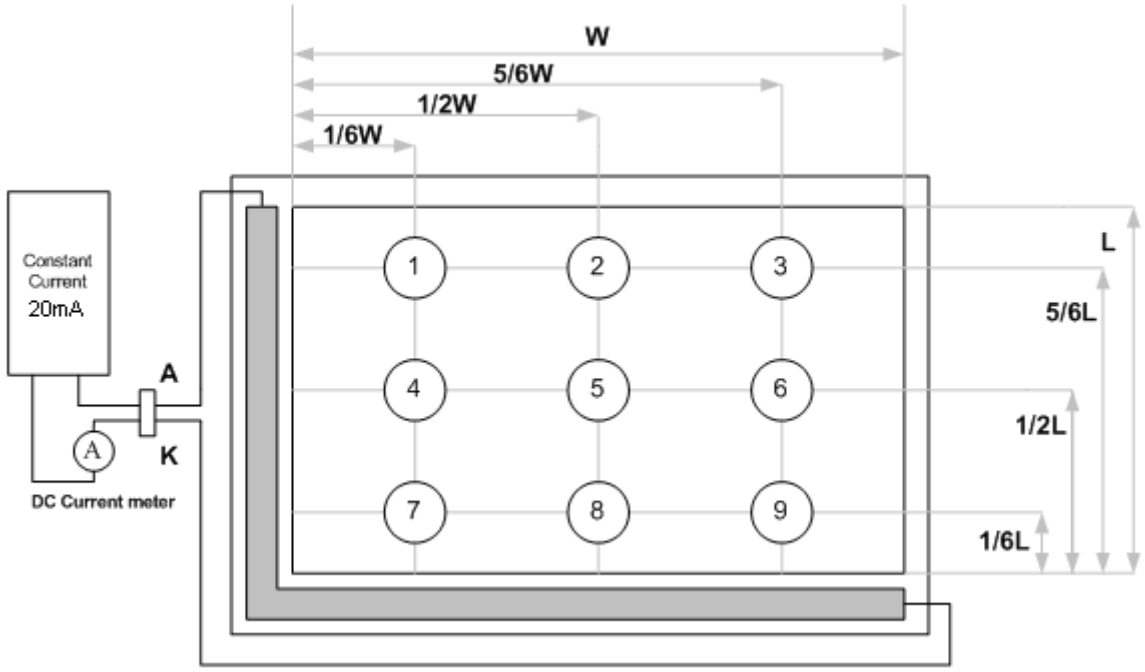
5.2 Optical characteristic of the LED Back-light

ITEM	MIN	TYP	MAX	UNIT	Condition
Bare Brightness	--	3300	--	Cd/m2	I_{LED} = 20 mA,Ta=25℃
AVG. X of 1931 C.I.E.	(0.26)	(0.29)	(0.32)	--	I_{LED} = 20 mA,Ta=25℃
AVG. X of 1931 C.I.E.	(0.25)	(0.28)	(0.31)	--	I_{LED} = 20 mA,Ta=25℃
Brightness Uniformity	80	--	--	%	I_{LED} = 20 mA,Ta=25℃

( )For reference only. These data should be update according the prototype.

Note1 : Measurement after 10 minutes from LED BL operating.

Note2 : Measurement of the following 9 places on the display.



Note3: The Uniformity definition

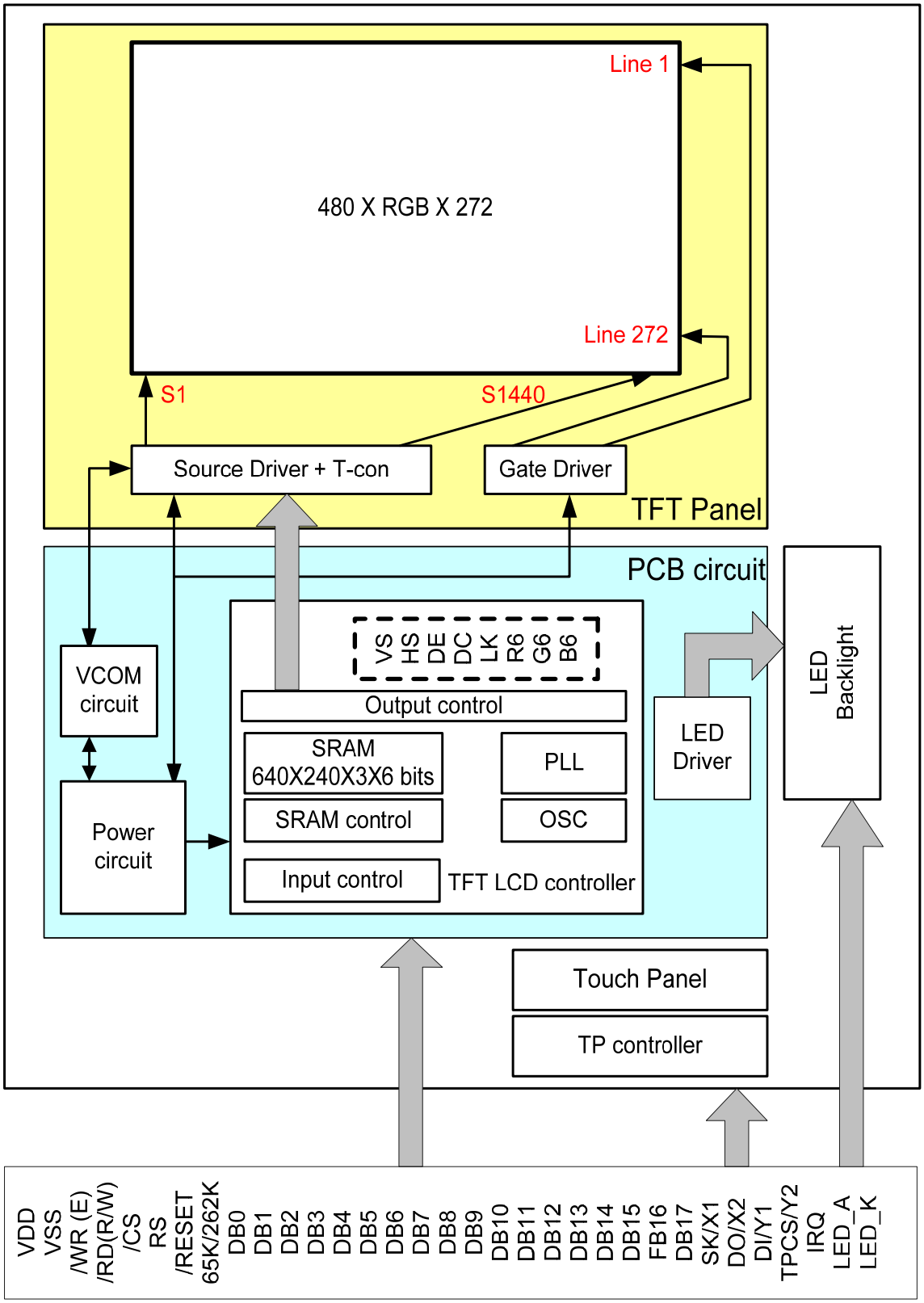
(Min Brightness / Max Brightness) x 100%

6 Interface specifications

6.1 Driving signals for the TFT panel

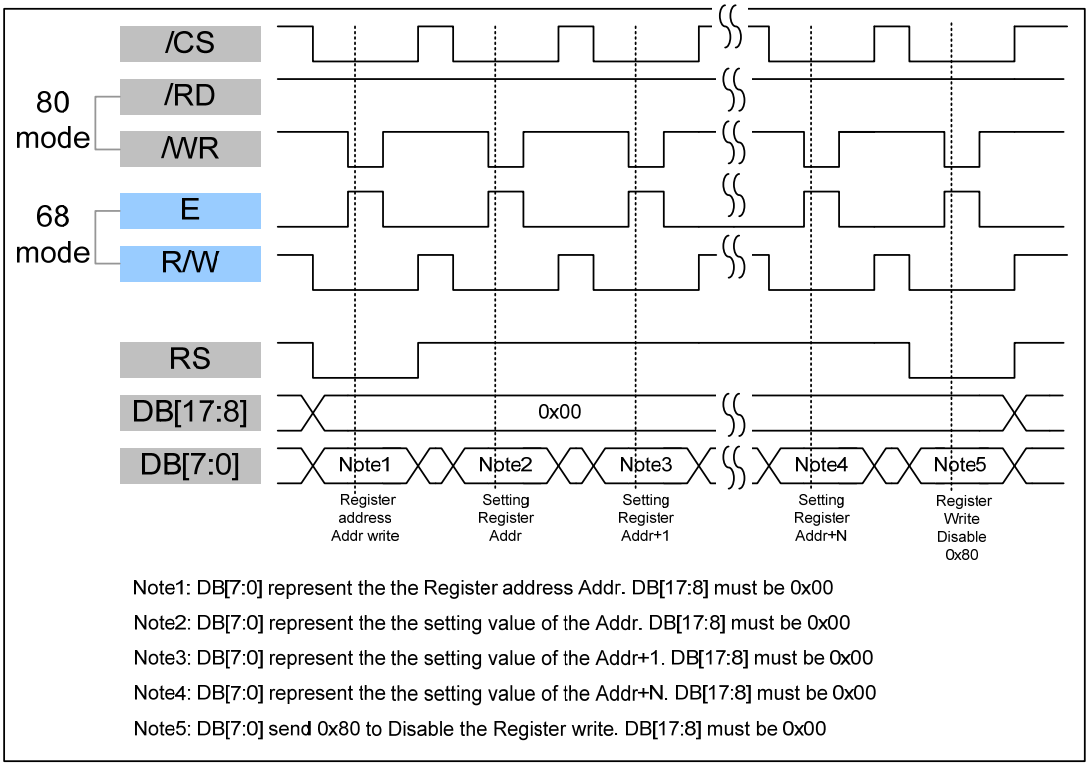
Pin no	Symbol	I/O	Description		Remark
1 ~ 2	VSS		GND		
3	LED_A/PWM		Without LED driver	LED Anode	
			With LED Driver	**PWM**	**Default**
4	LED_K		Without LED driver	LED Cathode	
			With LED Driver	**Must be OPEN**	**Default**
5	/RESET	I	Reset signal for TFT LCD controller		
6	RS	I	Register and Data select for TFT LCD controller		
7	/CS	I	Chip select low active signal for TFT LCD controller		
8	/WR(E)	I	80mode : /WR low active signal for TFT LCD controller 68mode : E signal latch on rising edge		
9	/RD(R/W)	I	80mode : /RD low active signal for TFT LCD controller 68mode : R/W signal Hi: read Lo:Write		
10 ~ 27	DB0 ~ DB17	I/O	Data Bus		
28	65K/262K	I	Select colors data format H : 262K L : 65K		
29	VSS		GND		
30	SK/X1	-	**Serial clock for Touch panel controller**		
31	DO/X2	-	**Data Output for Touch panel controller**		
32	DI / Y1	-	**Data In for Touch panel controller**		
33	TPCS / Y2	-	**Chip Select for Touch panel controller**		
34	IRQ	-	Interrupt for Touch panel controller		
35 ~ 37	VDD		Power supply for the logic (3.3V)		
38 ~ 40	VSS		GND		

7 BLOCK DIAGRAM



8 Interface Protocol

8.1 18Bit-80/68-Write to Command Register



Note1: DB[7:0] represent the the Register address Addr. DB[17:8] must be 0x00

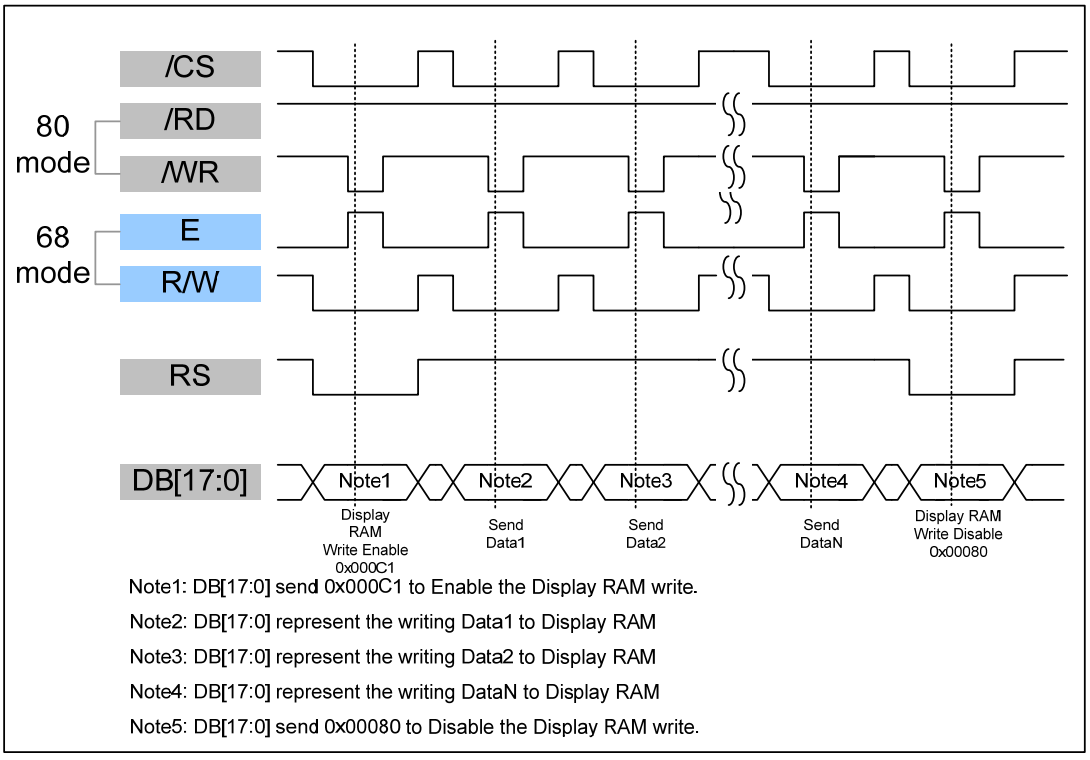
Note2: DB[7:0] represent the the setting value of the Addr. DB[17:8] must be 0x00

Note3: DB[7:0] represent the the setting value of the Addr+1. DB[17:8] must be 0x00

Note4: DB[7:0] represent the the setting value of the Addr+N. DB[17:8] must be 0x00

Note5: DB[7:0] send 0x80 to Disable the Register write. DB[17:8] must be 0x00

8.2 18Bit-80/68-Write to Display RAM



Note1: DB[17:0] send 0x000C1 to Enable the Display RAM write.

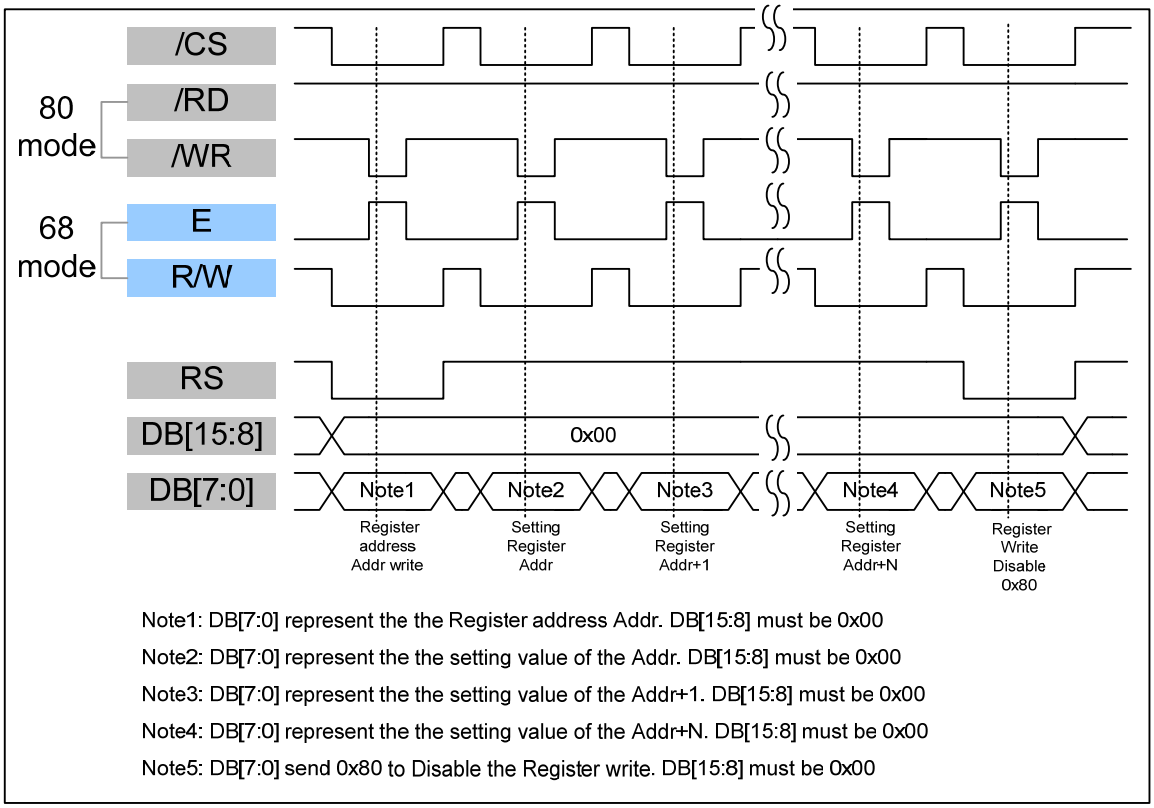
Note2: DB[17:0] represent the writing Data1 to Display RAM

Note3: DB[17:0] represent the writing Data2 to Display RAM

Note4: DB[17:0] represent the writing DataN to Display RAM

Note5: DB[17:0] send 0x00080 to Disable the Display RAM write.

8.3 16Bit-80/68- Write to Command Register



Note1: DB[7:0] represent the the Register address Addr. DB[15:8] must be 0x00

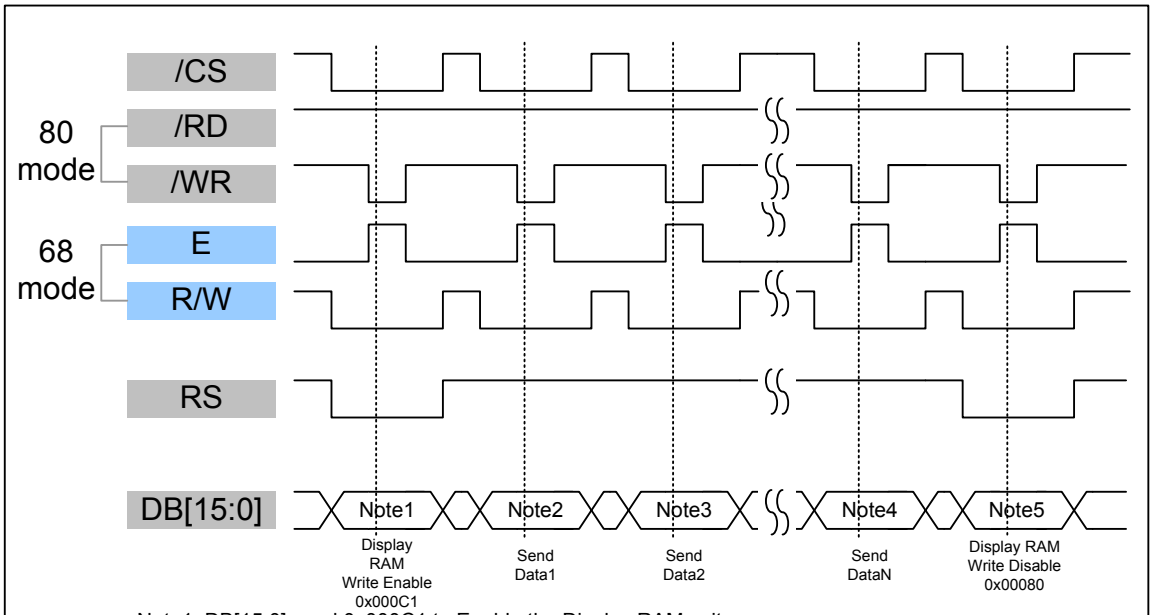
Note2: DB[7:0] represent the the setting value of the Addr. DB[15:8] must be 0x00

Note3: DB[7:0] represent the the setting value of the Addr+1. DB[15:8] must be 0x00

Note4: DB[7:0] represent the the setting value of the Addr+N. DB[15:8] must be 0x00

Note5: DB[7:0] send 0x80 to Disable the Register write. DB[15:8] must be 0x00

8.4 16Bit-80/68-Write to Display RAM



Note1: DB[15:0] send 0x000C1 to Enable the Display RAM write.

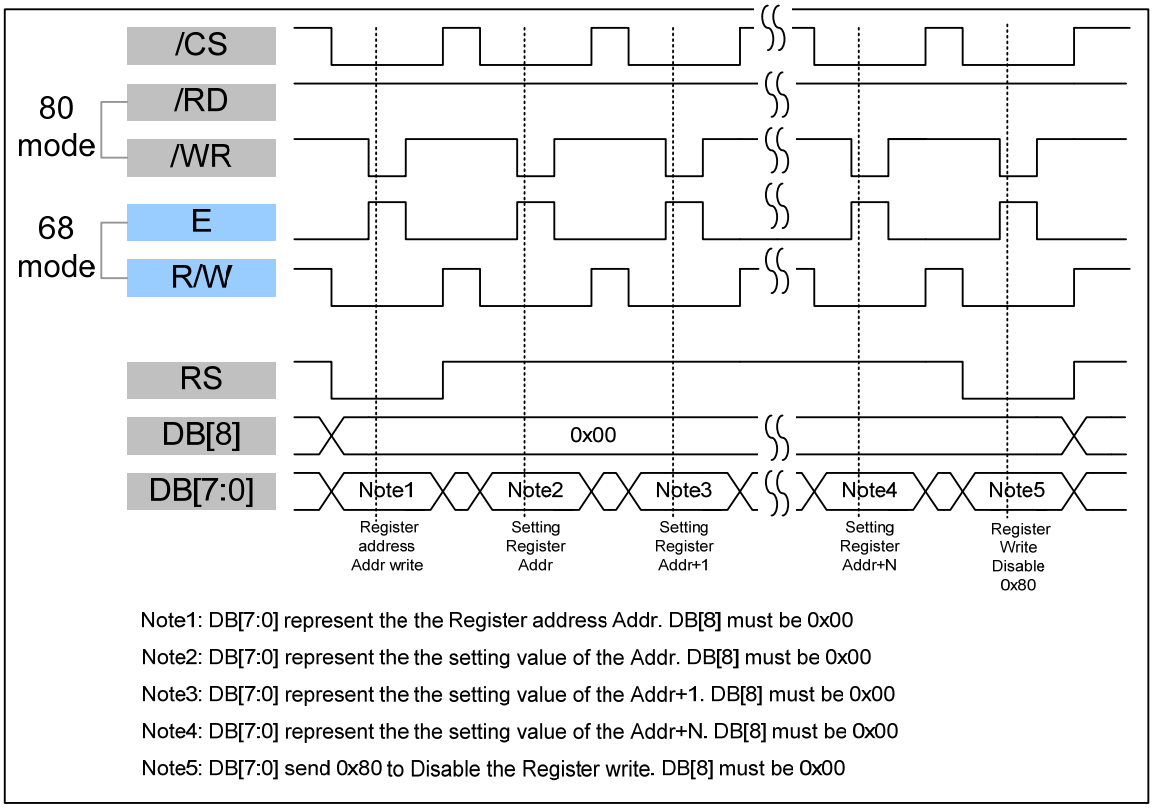
Note2: DB[15:0] represent the writing Data1 to Display RAM

Note3: DB[15:0] represent the writing Data2 to Display RAM

Note4: DB[15:0] represent the writing DataN to Display RAM

Note5: DB[15:0] send 0x00080 to Disable the Display RAM write.

8.5 9Bit-80/68- Write to Command Register



Note1: DB[7:0] represent the the Register address Addr. DB[8] must be 0x00

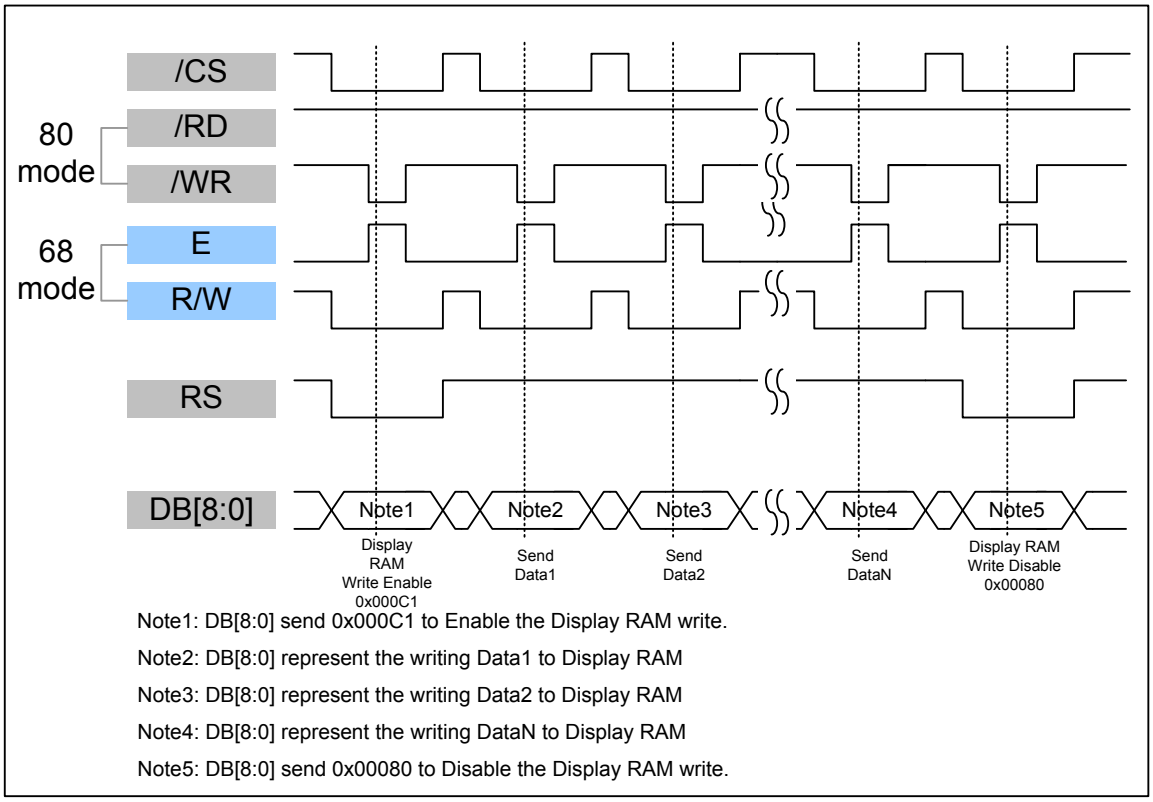
Note2: DB[7:0] represent the the setting value of the Addr. DB[8] must be 0x00

Note3: DB[7:0] represent the the setting value of the Addr+1. DB[8] must be 0x00

Note4: DB[7:0] represent the the setting value of the Addr+N. DB[8] must be 0x00

Note5: DB[7:0] send 0x80 to Disable the Register write. DB[8] must be 0x00

8.6 9Bit-80/68-Write to Display RAM



Note1: DB[8:0] send 0x000C1 to Enable the Display RAM write.

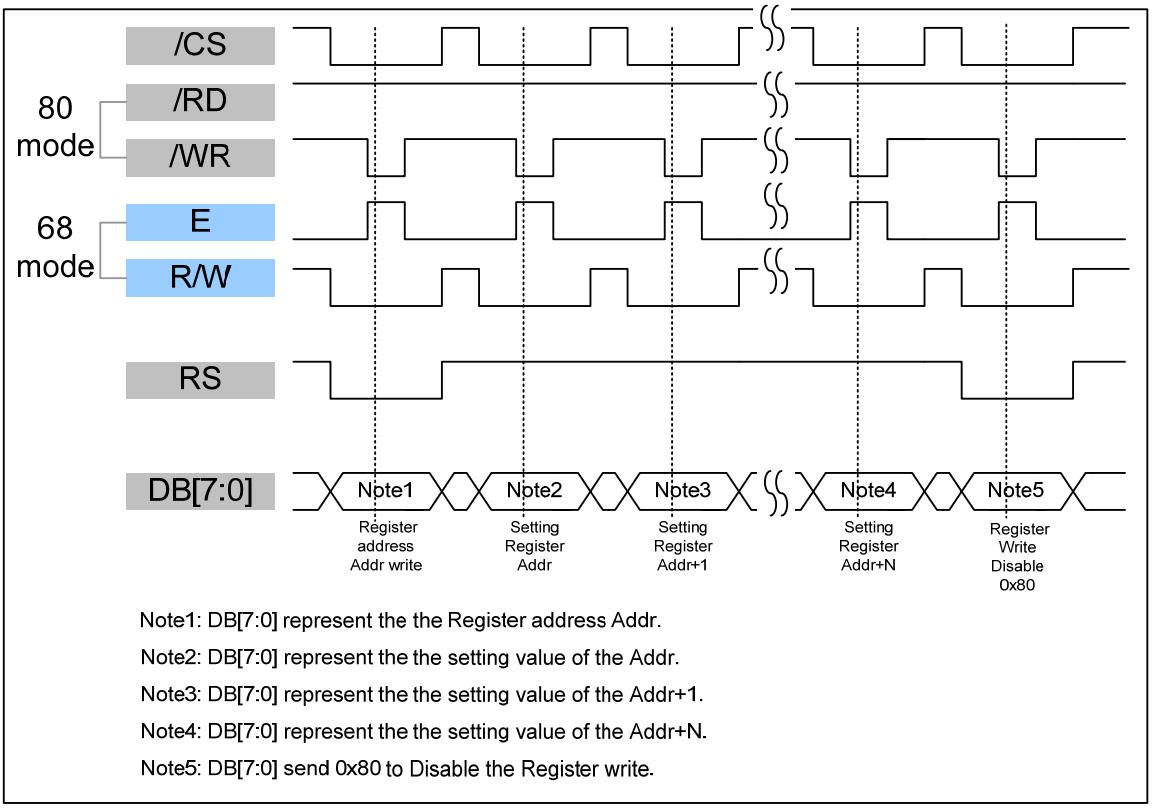
Note2: DB[8:0] represent the writing Data1 to Display RAM

Note3: DB[8:0] represent the writing Data2 to Display RAM

Note4: DB[8:0] represent the writing DataN to Display RAM

Note5: DB[8:0] send 0x00080 to Disable the Display RAM write.

8.7 8Bit-80/68- Write to Command Register



Note1: DB[7:0] represent the the Register address Addr.

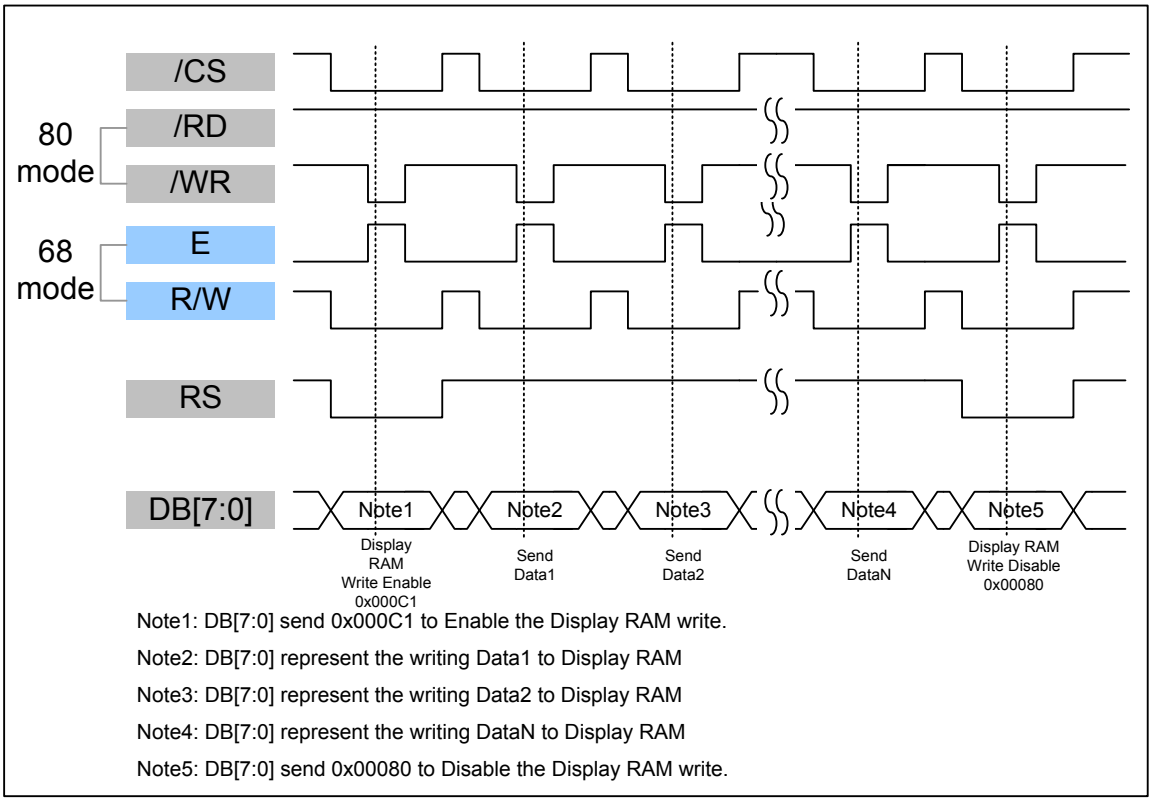
Note2: DB[7:0] represent the the setting value of the Addr.

Note3: DB[7:0] represent the the setting value of the Addr+1.

Note4: DB[7:0] represent the the setting value of the Addr+N.

Note5: DB[7:0] send 0x80 to Disable the Register write.

8.8 8Bit-80/68-Write to Display RAM



Note1: DB[7:0] send 0x000C1 to Enable the Display RAM write.

Note2: DB[7:0] represent the writing Data1 to Display RAM

Note3: DB[7:0] represent the writing Data2 to Display RAM

Note4: DB[7:0] represent the writing DataN to Display RAM

Note5: DB[7:0] send 0x00080 to Disable the Display RAM write.

8.9 Data transfer order Setting

8.9.1 18 bit interface 262K color only (Pin28 65K/262K =High)

DB	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0

8.9.2 16 bit interface 65K color (Pin28 65K/262K =Low)

DB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

8.9.3 16 bit interface 262K color (Pin28 65K/262K =High)

DB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st data	X	X	X	X	X	X	X	X	X	X	X	X	X	X	R5	R4
2nd data	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0

8.9.4 9 bit interface 262K color only (Pin28 65K/262K =High)

DB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st data	X	X	X	X	X	X	X	R5	R4	R3	R2	R1	R0	G5	G4	G3
2nd data	X	X	X	X	X	X	X	G2	G1	G0	B5	B4	B3	B2	B1	B0

8.9.5 8 bit interface 65K color (Pin28 65K/262K =Low)

DB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st data	X	X	X	X	X	X	X	X	R4	R3	R2	R1	R0	G5	G4	G3
2nd data	X	X	X	X	X	X	X	X	G2	G1	G0	B4	B3	B2	B1	B0

8.9.6 8 bit interface 262K color (Pin28 65K/262K =High)

DB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st data	X	X	X	X	X	X	X	X							R5	R4
2nd data	X	X	X	X	X	X	X	X	R3	R2	R1	R0	G5	G4	G3	G2
3rd data	X	X	X	X	X	X	X	X	G1	G0	B5	B4	B3	B2	B1	B0

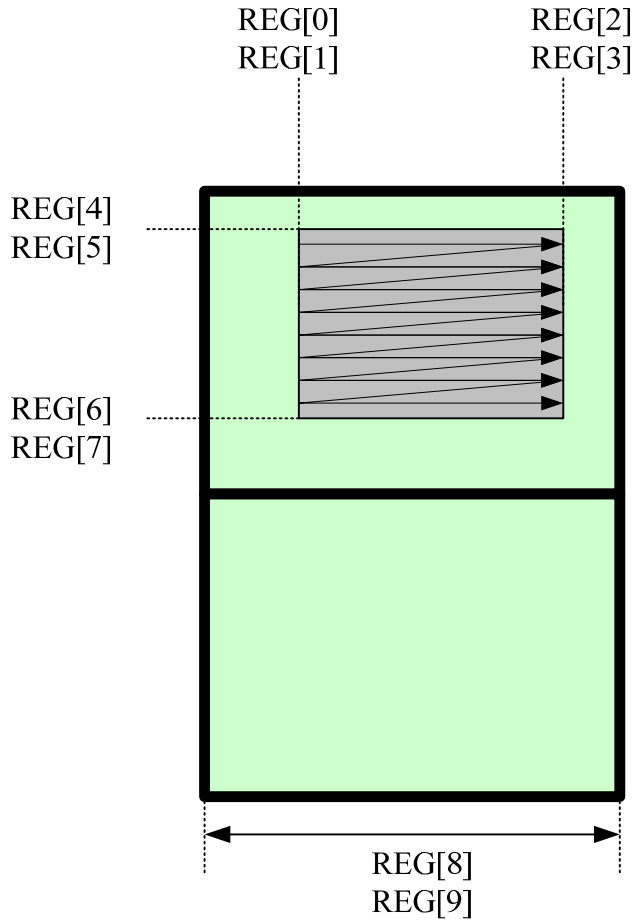
9 Register Depiction

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
00	00	MSB of X-axis start position								
Description	set the horizontals start position of display active region									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
01	00	LSB of X-axis start position								
Description	set the horizontals start position of display active region									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
02	01	MSB of X-axis end position								
Description	set the horizontals end position of display active region									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
03	3F	LSB of X-axis end position								
Description	set the horizontals end position of display active region									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
04	00	MSB of Y-axis start position								
Description	set the vertical start position of display active region									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
05	00	LSB of Y-axis start position								
Description	Set the vertical start position of display active region									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
06	00	MSB of Y-axis end position								
Description	set the vertical end position of display active region									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
07	EF	LSB of Y-axis end position								
Description	Set the vertical end position of display active region									

To simplify the address control of display RAM access, the window area address function allows for writing data only within a window area of display RAM specified by registers REG[00]~REG[07] .

After writing data to the display RAM, the Address counter will be increased within setting window address-range which is specified by

MIN X address (REG[0] & REG[1])
MAX X address (REG[2] & REG[3])
MIN Y address (REG[4] & REG[5])
MAX Y address (REG[6] & REG[7])
Therefore, data can be written consecutively without thinking the data address.



Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
08	01	X	X	X	X	X	X	_PanelXSize H_Byte[1:0]		
Description	Set the panel X size									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
09	40	_PanelXSize L_Byte[7:0]								
Description	Set the panel X size									

The register REG[08] and REG[09] is use to calculate the RAM address. If you want to use the TFT as Landscape mode (320x240), the REG[08] & RGE[09 must set to 320. If you want to use the TFT as Portrait mode (240x320), the REG[08] & RGE[09] must set to 240.

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0A	00	X	X	X	X	X	[17:16] bits of memory write start address			
Description	Memory write start address									

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0B	00	[15:8] bits of memory write start address								
Description	Memory write start address									

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0C	00	[7:0] bits of memory write start address								
Description	Memory write start address									

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x10	0x0D	Bit_SWAP	OUT_TEST	BUS_SEL		Blanking	P/S_SEL	CLK_SEL		
Description	"0x10_Clk_sel[1:0]" : The TFT controller built-in 40Mhz PLL clock. These bits are for select the TFT panel dot clock frequency. 00 : 20Mhz 01: 10Mhz 02: 5 Mhz									
	"0x10_ps_sel[2]" : The TFT controller support parallel and serial RGB interface. These bits are for select the output timing. 0 : serial Panel 1: Parallel panel									
	"0x10_blanking_tmp[3]" 0 : OFF (blanking) 1: ON (normal operation)									
	"0x10_bus_sel[5:4]" : It only for serial Panel 00=R , 01=G , 10=B									
	"0x10_out_test[6]" : Self test 0 : normal operation 1: for test (don't use for normal operation) When set the bit to "1" , the Rout=(Reg 2a[6:0]) Gout=(Reg 2b[6:0]) Bout=(Reg 2c[6:0])									
	"0x10_bit_swap[7]" : 0-normal									
	The default setting is suitable for AM320240N1. Don't need to modify it.									

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x11	00	X	X	EVEN			_ODD			
Description	" Even line of serial panel data out sequence or data bus order of parallel panel 000: RGB 001: RBG 010: GRB 011: GBR									

	100: BRG 101: BGR Others: reserved Odd line of serial panel data out sequence 000: RGB 001: RBG 010: GRB 011: GBR 100: BRG **101: BGR** Others: reserved **Must Set to 0x05 for AM320240N1**

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x12	00					Hsync_stH_Byte[3:0]				
Description	For TFT output timing adjust: Hsync start position H-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x13	00	Hsync_stL_Byte[7:0]								
Description	For TFT output timing adjust: Hsync start position L-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x14	00					Hsync_pwH_Byte[3:0]				
Description	For TFT output timing adjust: Hsync pulse width H-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x15	10	Hsync_pwL_Byte[7:0]								
Description	For TFT output timing adjust: Hsync pulse width L-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x16	00					Hact_stH_Byte[3:0]				
Description	For TFT output timing adjust: DE pulse start position H-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register	Default	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark

Address (Hex)	(Hex)									
0x17	38	Hact_stL_Byte[7:0]								
Description	For TFT output timing adjust: DE pulse start position L-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x18	01					Hact_pwH_Byte[3:0]				
Description	For TFT output timing adjust: DE pulse width H-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x19	40	Hact_pwL_Byte[7:0]								
Description	For TFT output timing adjust: DE pulse width L-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x1A	01					HtotalH_Byte[3:0]				
Description	For TFT output timing adjust: Hsync total clocks H-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x1B	B8	HtotalL_Byte[7:0]								
Description	For TFT output timing adjust: Hsync total clocks H-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x1C	00					Vsync_stH_Byte[3:0]				
Description	For TFT output timing adjust: Vsync start position H-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x1D	00	Vsync_stL_Byte[7:0]								
Description	For TFT output timing adjust:									

	Vsync start position L-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x1E	00					Vsync_pwH_Byte[3:0]				
Description	For TFT output timing adjust: Vsync pulse width H-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x1F	08	Vsync_pwL_Byte[7:0]								
Description	For TFT output timing adjust: Vsync pulse width L-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x20	00					Vact_stH_Byte[3:0]				
Description	For TFT output timing adjust: Vertical DE pulse start position H-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x21	12	Vact_stL_Byte[7:0]								
Description	For TFT output timing adjust: Vertical DE pulse start position L-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x22	00					Vact_pwH_Byte[3:0]				
Description	For TFT output timing adjust: Vertical Active width H-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x23	F0	Vact_pwL_Byte[7:0]								
Description	For TFT output timing adjust: Vertical Active width H-Byte The default setting is suitable for AM320240N1. Don't need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x24	01					VtotalH_Byte[3:0]				
Description	For TFT output timing adjust: Vertical total width H-Byte									

	The default setting is suitable for AM320240N1. Don’t need to modify it.									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x25	09	VtotalL_Byte[7:0]								
Description	For TFT output timing adjust: Vertical total width L-Byte The default setting is suitable for AM320240N1. Don’t need to modify it.									

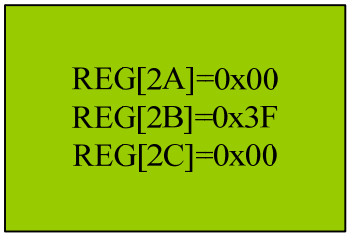
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
26	00	X	X	X	X	X	[17:16] bits of memory read start address			
Description	Memory read start address									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
27	00	[15:8] bits of memory write start address								
Description	Memory read start address									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
28	00	[7:0] bits of memory write start address								
Description	Memory read start address									

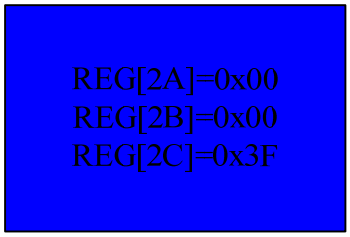
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
29	00	[7:1] Reversed								
Description	[0] Load output timing related setting (H sync., V sync. and DE) to take effect									

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x2A	00	X	TestPatternRout[6:0]							
Description	When " REG[0x10]_out_test[6]" : Self test =1 ; The Rout data equal to TestPatternRout[6:0]									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x2B	00	X	TestPatternGout[6:0]							
Description	When " REG[0x10]_out_test[6]" : Self test =1 ; The Gout data equal to TestPatternGout[6:0]									
Register	Default	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark

Address (Hex)	(Hex)									
0x2C	00	X	TestPatternBout[6:0]							
Description	When " REG[0x10]_out_test[6]" : Self test =1 ; The Bout data equal to TestPatternBout[6:0]									

If you set the " REG[0x10]_out_test[6]" : Self test =1 , the TFT controller will skip the connect of the display RAM. The Output port will send the REG[2A] ,REG[2B],REG[2C] data.





Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
0x2D	00	X	X	X	X	[3]	Rising/falling edge[2]	_rotate [1:0]		
Description	[3] Output pin X_DCON level control ; TFT Power ON/OFF control 0: TFT POWER circuit OFF 1: TFT POWER circuit ON									
	Rising/falling edge[2] : 0: The RGB out put data are on the Rising edge of the DCLK. 1: The RGB out put data are on the Falling edge of the DCLK.									
	_rotate [1:0]: 00 : rotate 0 degree 01 : rotate90 degree 10 : rotate 270 degree 11 : rotate 180 degree									

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
30	00	X	X	X	X	X	_H byte H-Offset[3:0]			
Description	Set the Horizontal offset									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
31	00	_L byte H-Offset[7:0]								
Description	Set the Horizontal offset									

Register Address	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark

(Hex)										
32	00	X	X	X	X	X	_H byte V-Offset[3:0]			
Description	Set the Vertical offset									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
33	00	_L byte V-Offset[7:0]								
Description	Set the Vertical offset									

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
34	00	[7:4] Reserved					_H byte H-def[3:0]			
Description	[3:0] MSB of image horizontal physical resolution in memory									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
35	40	_L byte H-def[7:0]								
Description	[7:0] LSB of image horizontal physical resolution in memory									

Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
36	01	[7:4] Reserved					_H byte V-def[3:0]			
Description	[3:0] MSB of image vertical physical resolution in memory									
Register Address (Hex)	Default (Hex)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Remark
37	E0	_L byte V-def[7:0]								
Description	[7:0] LSB of image vertical physical resolution in memory									

The total RAM size is 640x240x18bit. The user can arrange the Horizontal ram size by REG[34],REG[35] and the Vertical ram size by REG[36],REG[37].

EX: 320x480x18bit REG[34]=0x01 , REG[35]=0x40 , REG[36]=0x01 , REG[37]=0xE0

EX: 640x240x18bit. REG[34]=0x02 , REG[35]=0x80 , REG[36]=0x00 , REG[37]=0xF0

DISPLAYED COLOR AND INPUT DATA

	Color & Gray Scale	DATA SIGNAL																	
		R5	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1	B0
Basic Color	Black	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Red(0)	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	Green(0)	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
	Blue(0)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
	Cyan	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
	Magenta	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1
	Yellow	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
	White	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Red	Black	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Red(62)	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	Red(61)	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
	Red(31)	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
	Red(1)	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	Red(0)	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Green	Black	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Green(62)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	Green(61)	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
	Green(31)	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
	Green(1)	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
	Green(0)	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
Blue	Black	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Blue(62)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	Blue(61)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
	Blue(31)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
	Blue(1)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0
	Blue(0)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

10. LCD Controller initial code

```
/* Exported types ------------------------------------------------------------*/
typedef   unsigned        char       uint8;
typedef   signed          char       int8;
typedef   unsigned        short      uint16;
typedef   signed          short      int16;
typedef   unsigned long       uint32;
typedef   signed          int        int32;


/*****************************************************************/
/*                    STEP1: Define MCU BUS type                 */
/*****************************************************************/
  #define Mode80     // 8080 MCU    /WR    /RD
  //#define Mode68          // 6800 MCU    R/W    E
/*****************************************************************/
/*                    STEP2: Define BUS wide                     */
/*****************************************************************/
  //#define C80_18B
  #define C80_16B
  //#define C80_9B
  //#define C80_8B
/*****************************************************************/
/*                    STEP3: Define Landscap/Portrait            */
/*****************************************************************/
  #define Landscap
  //#define Portrait
/*****************************************************************/
/*                    STEP4: Define Resolution                   */
/*****************************************************************/
#ifdef Landscap
#define       Resolution_X 480
#define       Resolution_Y 272

#endif

#ifdef Portrait
```

```
#define         Resolution_X 272
#define         Resolution_Y 480
#endif
/*****************************************************************/
/*                        STEP5: TFT timing                      */
/*****************************************************************/
# define Rising    0<<2       // Don't need to change
# define Falling 1<<2         // Don't need to change


#define LCD_DCLK 12            /*    LCD_DCLK=(40*(0x42)/(0x41))/R10_B10*/
                               /*5, 6.67, 7.5, 8.57, 10, 12, 15, */
#define LCD_DCLK_Latch        Falling //Rising           //0<<2:normal    , 1<<2:Inverse

#define H_Sync_Pluse_Wide    41
#define H_Sync_to_DE        43// DE horizontal start position
#define H_Sync_total             525
#define V_Sync_Pluse_Wide    10
#define V_Sync_to_DE        12// DE horizontal start position
#define V_Sync_total             286

//*************************************************************//
/**************Don't need to change the bellow macro**************/

#if LCD_DCLK== 5
   #define R41              1
   #define R42              1
   #define R10_B10          2
#endif

#if LCD_DCLK== 6
   #define R41              3
   #define R42              4
   #define R10_B10          2
#endif

#if LCD_DCLK== 7
   #define R41              4
   #define R42              3
```

```
   #define R10_B10       1
#endif

#if LCD_DCLK== 8
   #define R41           12
   #define R42           10
   #define R10_B10       1
#endif

#if LCD_DCLK== 10
   #define R41           1
   #define R42           1
   #define R10_B10       1
#endif

#if LCD_DCLK== 12
   #define R41           5
   #define R42           6
   #define R10_B10       1
#endif


#if LCD_DCLK== 15
   #define R41           2
   #define R42           3
   #define R10_B10       1
#endif


#define _DisplayRAM_WriteEnable_ 0xc1
#define _DisplayRAM_WriteDisable_ 0x80


typedef struct
{
      uint8   REG_Index;
      uint8   REG_Value;

}FSA506_REG_Setting;
```

```
#ifdef    Landscap

static    FSA506_REG_Setting FSA506_A[] =

{
  {0x40,0x12},
  {0x41,R41},
  {0x42,R42},
  {0x08,(uint8)(Resolution_X>>8)},
  {0x09,(uint8)(Resolution_X)},
  {0x0a,0x00},
  {0x0b,0x00},
  {0x0c,0x00},
  {0x10,0x0C|R10_B10},
  //{0x10,0x0C|0x02},
  {0x11,0x05},
  {0x12,0x00},
  {0x13,0x00},
  {0x14,(uint8)(H_Sync_Pluse_Wide>>8)},
  {0x15,(uint8)(H_Sync_Pluse_Wide)},
  {0x16,(uint8)(H_Sync_to_DE>>8)},
  {0x17,(uint8)(H_Sync_to_DE)},
  {0x18,(uint8)(Resolution_X>>8)},
  {0x19,(uint8)(Resolution_X)},
  {0x1a,(uint8)(H_Sync_total>>8)},
  {0x1b,(uint8)(H_Sync_total)},
  {0x1c,0x00},
  {0x1d,0x00},
  {0x1e,(uint8)(V_Sync_Pluse_Wide>>8)},
  {0x1f,(uint8)(V_Sync_Pluse_Wide)},
  {0x20,(uint8)(V_Sync_to_DE>>8)},
  {0x21,(uint8)(V_Sync_to_DE)},
  {0x22,(uint8)(Resolution_Y>>8)},
  {0x23,(uint8)(Resolution_Y)},
  {0x24,(uint8)(V_Sync_total>>8)},
  {0x25,(uint8)(V_Sync_total)},
  {0x26,0x00},
  {0x27,0x00},
```

```
  {0x28,0x00},
  {0x29,0x01},


  {0x2d,LCD_DCLK_Latch|0x08},
  // [7:4] Reserved
  // [3] Output pin X_DCON level control
  // [2] Output clock inversion   0: Normal 1: Inverse
  // [1:0] Image rotate
//        00: 0°    01: 90°    10: 270° 11: 180°

  {0x30,0x00},
  {0x31,0x00},
  {0x32,0x00},
  {0x33,0x00},
  {0x34,(uint8)(Resolution_X>>8)},
  {0x35,(uint8)(Resolution_X)},
  {0x36,(uint8)((2*Resolution_Y)>>8)},
  {0x37,(uint8)(2*Resolution_Y)},

};
#endif

#ifdef    Portrait

static    FSA506_REG_Setting FSA506_A[] =

{
  {0x40,0x12},
  {0x41,R41},
  {0x42,R42},
  {0x08,(uint8)(Resolution_X>>8)},
  {0x09,(uint8)(Resolution_X)},
  {0x0a,0x00},
  {0x0b,0x00},
  {0x0c,0x00},
  {0x10,0x0C|R10_B10},
  //{0x10,0x0C|0x02},
  {0x11,0x05},
```

```
  {0x12,0x00},
  {0x13,0x00},
  {0x14,(uint8)(H_Sync_Pluse_Wide>>8)},
  {0x15,(uint8)(H_Sync_Pluse_Wide)},
  {0x16,(uint8)(H_Sync_to_DE>>8)},
  {0x17,(uint8)(H_Sync_to_DE)},
  {0x18,(uint8)(Resolution_Y>>8)},
  {0x19,(uint8)(Resolution_Y)},
  {0x1a,(uint8)(H_Sync_total>>8)},
  {0x1b,(uint8)(H_Sync_total)},
  {0x1c,0x00},
  {0x1d,0x00},
  {0x1e,(uint8)(V_Sync_Pluse_Wide>>8)},
  {0x1f,(uint8)(V_Sync_Pluse_Wide)},
  {0x20,(uint8)(V_Sync_to_DE>>8)},
  {0x21,(uint8)(V_Sync_to_DE)},
  {0x22,(uint8)(Resolution_X>>8)},
  {0x23,(uint8)(Resolution_X)},
  {0x24,(uint8)(V_Sync_total>>8)},
  {0x25,(uint8)(V_Sync_total)},
  {0x26,0x00},
  {0x27,0x00},
  {0x28,0x00},
  {0x29,0x01},


  {0x2d,LCD_DCLK_Latch|0x08|0x01},
  // [7:4] Reserved
  // [3] Output pin X_DCON level control
  // [2] Output clock inversion  0: Normal 1: Inverse
  // [1:0] Image rotate
//       00: 0°    01: 90°    10: 270° 11: 180°

  {0x30,0x00},
  {0x31,0x00},
  {0x32,0x00},
  {0x33,0x00},
  {0x34,(uint8)(Resolution_X>>8)},
  {0x35,(uint8)(Resolution_X)},
```

```
  {0x36,(uint8)((2*Resolution_Y)>>8)},
  {0x37,(uint8)(2*Resolution_Y)},

};
#define          NOP()              __asm{NOP}

#endif
/**************Don't need to change the above macro**************/



void AMP506_80Mode_Command_SendAddress(uint8 Addr);
void AMP506_80Mode_Command_SendData(uint8 Data);
void AMP506_80Mode_16Bit_Memory_SendData(uint16 Dat16bit);
void AMP506_Command_Write(uint8 CMD_Address,uint8 CMD_Value);
void Initial_AMP506(void) ;
void AMP506_WindowSet(uint16 S_X,uint16 S_Y,uint16 E_X,uint16 E_Y)    ;
void FD506_DisplayRAM_WriteEnable(void);
void FD506_DisplayRAM_WriteDisable(void);
void GUI_RectangleFill(uint32 x0, uint32 y0, uint32 x1, uint32 y1, uint16 color);
void Full_LCD(uint16 Dat16bit);
void LCD_Pixel(uint16 x , uint16 y , uint16 couleur);

/**************FSA506 Write Registr Address function    ************************/
void AMP506_80Mode_Command_SendAddress(uint8 Addr)
{

   #ifdef Mode68
   uint16 i;
   CLR_nWRL;
   CLR_RS;
   CLR_CS1;
   CLR_nRD;
   DB16OUT(Addr);
   NOP();NOP();
   SET_nWRL;          //Enable
   NOP();NOP();NOP();    NOP();NOP();//NOP(); NOP();NOP();NOP();
   CLR_nWRL;          //Enable
   SET_RS;
```

```
  SET_CS1;
  #endif



  #ifdef Mode80
  SET_nRD;                  //SET_RW
  CLR_RS;
  DB16OUT(Addr); NOP();
  CLR_CS1;
  CLR_nWRL;                 //CLR_E

  NOP();NOP();NOP();
  SET_nWRL;                 //SER_E       // Low to High Latch Data to AMP506 Buffer
  SET_RS;
  SET_CS1;
  #endif


}
/**************FSA506 Write Command Data function    ************************/
void AMP506_80Mode_Command_SendData(uint8 Data)
{
  #ifdef Mode68
  uint16 i;
  CLR_nWRL;       //E
  SET_RS;
  CLR_CS1;
  CLR_nRD;    //W/R
  DB16OUT(Data);
  NOP();NOP();
  SET_nWRL;
  NOP();NOP();NOP();NOP();NOP();//NOP();NOP();NOP();
  CLR_nWRL;       //E nable
  SET_RS;
  SET_CS1;

  #endif
```

```
   #ifdef Mode80
   SET_nRD;
   SET_RS;
   DB16OUT(Data); NOP(); // NOP()
   CLR_CS1;
   CLR_nWRL;

   NOP();NOP();NOP();
   SET_nWRL;                          // Low to High Latch Data to AMP506 Buffer
   SET_RS;
   SET_CS1;
   #endif

}
/**************FSA506 Write Data function    ************************/
void AMP506_80Mode_16Bit_Memory_SendData(uint16 Dat16bit)
{


  #ifdef Mode80
   #ifdef C80_16B
   SET_nRD;
   SET_RS;
   DB16OUT(Dat16bit);NOP();
   CLR_CS1;

   CLR_nWRL;

   NOP(); NOP();    NOP();

   SET_nWRL;                          // Low to High Latch Data to AMP506 Buffer
   SET_CS1;

   #endif

   #ifdef C80_8B
   DB16OUT(Dat16bit>>8);NOP();NOP();
   SET_nRD;
```

```
SET_RS;

CLR_CS1;
CLR_nWRL;

NOP(); NOP();    NOP();
SET_nWRL;                          // Low to High Latch Data to AMP506 Buffer
SET_CS1;

//Delay_uS(1);
DB16OUT(Dat16bit);NOP(); NOP();
SET_nRD;
SET_RS;

CLR_CS1;
CLR_nWRL;
NOP(); NOP();    NOP();
SET_nWRL;                          // Low to High Latch Data to AMP506 Buffer
SET_CS1;

#endif
//Delay_uS(1);


#ifdef C80_18B

uint32 k=0;
uint16 R_temp,G_temp,B_temp;

R_temp=((0xf800&Dat16bit)>>11);
G_temp=((0x07e0&Dat16bit)>>5);
B_temp=((0x001f&Dat16bit));

k|=((R_temp<<1)<<12);    //+G_temp+B_temp;
k|=(G_temp<<6);
k|=(B_temp<<1);

FIO1MASK=0xFFE0FFFF;                    // FIOMASK  只可寫 P1.20~P1.16
FIO1PIN=k;                    //  將 Address A20~A16 寫入 P1.20~P1.16
```

```
FIO1MASK=0x00;


SET_nRD;
SET_RS;
DB16OUT(k);NOP();
CLR_CS1;
CLR_nWRL;

 NOP(); NOP();    NOP();


SET_nWRL;                              // Low to High Latch Data to AMP506 Buffer
SET_CS1;

#endif

#ifdef C80_9B

uint32 k=0;
uint16 R_temp,G_temp,B_temp;

R_temp=((0xf800&Dat16bit)>>11);
G_temp=((0x07e0&Dat16bit)>>5);
B_temp=((0x001f&Dat16bit));

k|=((R_temp<<1)<<12);    //+G_temp+B_temp;
k|=(G_temp<<6);
k|=(B_temp<<1);



SET_nRD;
SET_RS;
CLR_CS1;
CLR_nWRL;

DB16OUT(((k&0x3FE00)>>9));
```

```
    SET_nWRL;                           // Low to High Latch Data to AMP506 Buffer
    DB16OUT((k&0x1FF)); NOP();
    SET_CS1;
   // Delay_uS(1);
    SET_nRD;
    SET_RS;
    CLR_CS1;
    CLR_nWRL;

    NOP(); NOP();    NOP();

    SET_nWRL;                           // Low to High Latch Data to AMP506 Buffer
    SET_CS1;

    #endif
   #endif


#ifdef Mode68
    #ifdef C80_16B
    uint16 i;
    NOP();NOP();

    CLR_nWRL;       //E=0
    SET_RS;
    CLR_CS1;
    CLR_nRD;    // W/R=0

    DB16OUT(Dat16bit);

    SET_nWRL;                           // Low to High Latch Data to AMP506 Buffer
    NOP();NOP();NOP();NOP();//NOP();NOP();NOP();NOP();
    CLR_nWRL;                           // Low to High Latch Data to AMP506 Buffer
    SET_CS1;
    #endif


    #ifdef C80_8B
    uint16 i;
```

```
//for (i=0;i<16;i++);
NOP();NOP();
CLR_nWRL;       //E=0
SET_RS;
CLR_CS1;
CLR_nRD;   // W/R=0

DB16OUT(Dat16bit>>8);

SET_nWRL;                          // Low to High Latch Data to AMP506 Buffer
NOP();NOP();NOP();NOP();//NOP();NOP();NOP();NOP();
CLR_nWRL;                          // Low to High Latch Data to AMP506 Buffer
SET_CS1;

CLR_nWRL;       //E=0
SET_RS;
CLR_CS1;
CLR_nRD;   // W/R=0

DB16OUT(Dat16bit);

SET_nWRL;                          // Low to High Latch Data to AMP506 Buffer
NOP();NOP();NOP();NOP();//NOP();NOP();NOP();NOP();
CLR_nWRL;                          // Low to High Latch Data to AMP506 Buffer
SET_CS1;

#endif
//Delay_uS(1);


#ifdef C80_18B

uint32 k=0;
uint16 R_temp,G_temp,B_temp;
uint16 i;
NOP();NOP();

R_temp=((0xf800&Dat16bit)>>11);
G_temp=((0x07e0&Dat16bit)>>5);
```

```
B_temp=((0x001f&Dat16bit));

k|=((R_temp<<1)<<12);    //+G_temp+B_temp;
k|=(G_temp<<6);
k|=(B_temp<<1);

FIO1MASK=0xFFE0FFFF;                    // FIOMASK  只可寫 P1.20~P1.16
FIO1PIN=k;                      //  將 Address A20~A16 寫入 P1.20~P1.16
FIO1MASK=0x00;



CLR_nWRL;        //E=0
SET_RS;
CLR_CS1;
CLR_nRD;    // W/R=0

DB16OUT(k);

SET_nWRL;                       // Low to High Latch Data to AMP506 Buffer
NOP();NOP();NOP();NOP();NOP();//NOP();NOP();NOP();
CLR_nWRL;                       // Low to High Latch Data to AMP506 Buffer
SET_CS1;

#endif

#ifdef C80_9B

uint32 k=0;
uint16 R_temp,G_temp,B_temp;
uint16 i;
//for (i=0;i<16;i++);
NOP();NOP();
R_temp=((0xf800&Dat16bit)>>11);
G_temp=((0x07e0&Dat16bit)>>5);
B_temp=((0x001f&Dat16bit));

k|=((R_temp<<1)<<12);    //+G_temp+B_temp;
k|=(G_temp<<6);
k|=(B_temp<<1);
```

```
   CLR_nWRL;       //E=0
   SET_RS;
   CLR_CS1;
   CLR_nRD;   // W/R=0

   DB16OUT(((k&0x3FE00)>>9));

   SET_nWRL;                              // Low to High Latch Data to AMP506 Buffer
   NOP();NOP();NOP();NOP();//NOP();NOP();NOP();NOP();
   CLR_nWRL;                              // Low to High Latch Data to AMP506 Buffer
   SET_CS1;

  // Delay_uS(1);
   CLR_nWRL;       //E=0
   SET_RS;
   CLR_CS1;
   CLR_nRD;   // W/R=0

   DB16OUT((k&0x1FF));

   SET_nWRL;                              // Low to High Latch Data to AMP506 Buffer
   NOP();NOP();NOP();NOP();//NOP();NOP();NOP();NOP();
   CLR_nWRL;                              // Low to High Latch Data to AMP506 Buffer
   SET_CS1;

   #endif
  #endif


}

/**************FSA506 Write Command function    ************************/
void AMP506_Command_Write(uint8 CMD_Address,uint8 CMD_Value)
{
   AMP506_80Mode_Command_SendAddress(CMD_Address);
   AMP506_80Mode_Command_SendData(CMD_Value);
```

```
}

/**************FSA506 Initial function    ************************/

void Initial_AMP506(void)                //

{          uint8 i;


      for(i=0;i < (sizeof(FSA506_A) / sizeof (FSA506_A[0]));i++)

      {

         AMP506_Command_Write(FSA506_A[i].REG_Index , FSA506_A[i].REG_Value);

      }

}

/**************FSA506 Set Start & End area function    ************************/
void AMP506_WindowSet(uint16 S_X,uint16 S_Y,uint16 E_X,uint16 E_Y)

{

            AMP506_80Mode_Command_SendAddress(0x00);

            AMP506_80Mode_Command_SendData((S_X)>>8);
            AMP506_80Mode_Command_SendData(S_X);

            AMP506_80Mode_Command_SendData((E_X-1)>>8);
            AMP506_80Mode_Command_SendData(E_X-1);

            AMP506_80Mode_Command_SendData(S_Y>>8);
            AMP506_80Mode_Command_SendData(S_Y);

            AMP506_80Mode_Command_SendData((E_Y-1)>>8);
            AMP506_80Mode_Command_SendData(E_Y-1);
```

```
}

//*****************************************************************************
//                          Enable Display RAM Write
//*****************************************************************************

void FD506_DisplayRAM_WriteEnable(void)
{

        AMP506_80Mode_Command_SendAddress(_DisplayRAM_WriteEnable_);

}

//*****************************************************************************
//                          Disable Display RAM Write
//*****************************************************************************

void FD506_DisplayRAM_WriteDisable(void)
{

        AMP506_80Mode_Command_SendAddress(_DisplayRAM_WriteDisable_);

}

/**************FSA506 Set Start & End area function    ************************/
void GUI_RectangleFill(uint32 x0, uint32 y0, uint32 x1, uint32 y1, uint16 color)
{
    uint32 k,l;


  AMP506_WindowSet(x0,y0,x1,y1);
  FD506_DisplayRAM_WriteEnable();
 for(k=y0;k<y1;k++)

    {
     for(l=x0;l<x1;l++)
       {
            AMP506_80Mode_16Bit_Memory_SendData(color);
         }
```

```
    }
   FD506_DisplayRAM_WriteDisable();

}
/**************Full Display function    ************************/
void Full_LCD(uint16 Dat16bit)
{

   GUI_RectangleFill(0,0,Resolution_X,Resolution_Y,Dat16bit);

}


void LCD_Pixel(uint16 x , uint16 y , uint16 couleur)
{
    uint8 hiByte, lowByte;



    AMP506_80Mode_Command_SendAddress(0x00);
    AMP506_80Mode_Command_SendData((x)>>8);
    AMP506_80Mode_Command_SendData(x);
    AMP506_80Mode_Command_SendData((x)>>8);
    AMP506_80Mode_Command_SendData(x);
    AMP506_80Mode_Command_SendData(y>>8);
    AMP506_80Mode_Command_SendData(y);
    AMP506_80Mode_Command_SendData((y)>>8);
    AMP506_80Mode_Command_SendData(y);

    FD506_DisplayRAM_WriteEnable();
    AMP506_80Mode_16Bit_Memory_SendData(couleur);
    FD506_DisplayRAM_WriteDisable();

}
```

11 RELIABILITY TEST CONDITIONS

ITEM	CONDITIONS	NOTE
HIGH TEMPERATURE OPERATION	70℃ , 240Hrs	
HIGH TEMPERATURE AND HIGH HUMIDITY OPERATION	60℃ , 90%RH , 240Hrs	
HIGH TEMPERATURE STORAGE	80℃ , 240Hrs	
LOW TEMPERATURE OPERATION	-20℃ , 240Hrs	
LOW TEMPERATURE STORAGE	-30℃ , 240Hrs	
THERMAL SHOCK	-30℃(1Hr) ~80℃(1Hr) 200Cycle	

12 USE PRECAUTIONS

12.1 Handling precautions

1) The polarizing plate may break easily so be careful when handling it. Do not touch, press or rub it with a hard-material tool like tweezers.
2) Do not touch the polarizing plate surface with bare hands so as not to make it dirty. If the surface or other related part of the polarizing plate is dirty, soak a soft cotton cloth or chamois leather in benzine and wipe off with it. Do not use chemical liquids such as acetone, toluene and isopropyl alcohol. Failure to do so may bring chemical reaction phenomena and deteriorations.
3) Remove any spit or water immediately. If it is left for hours, the suffered part may deform or decolorize.
4) If the LCD element breaks and any LC stuff leaks, do not suck or lick it. Also if LC stuff is stuck on your skin or clothing, wash thoroughly with soap and water immediately.

12.2 Installing precautions

1) The PCB has many ICs that may be damaged easily by static electricity. To prevent breaking by static electricity from the human body and clothing, earth the human body properly using the high resistance and discharge static electricity during the operation. In this case, however, the resistance value should be approx. 1MΩ and the resistance should be placed near the human body rather than the ground surface. When the indoor space is dry, static electricity may occur easily so be careful. We recommend the indoor space should be kept with humidity of 60% or more. When a soldering iron or other similar tool is used for assembly, be sure to earth it.
2) When installing the module and ICs, do not bend or twist them. Failure to do so may crack LC element and cause circuit failure.
3) To protect LC element, especially polarizing plate, use a transparent protective plate (e.g., acrylic plate, glass etc) for the product case.
4) Do not use an adhesive like a both-side adhesive tape to make LCD surface (polarizing plate) and product case stick together. Failure to do so may cause the polarizing plate to peel off.

12.3 Storage precautions

1) Avoid a high temperature and humidity area. Keep the temperature between 0°C and 35°C and also the humidity under 60%.
2) Choose the dark spaces where the product is not exposed to direct sunlight or fluorescent light.
3) Store the products as they are put in the boxes provided from us or in the same

conditions as we recommend.

12.4 Operating precautions

1) Do not boost the applied drive voltage abnormally. Failure to do so may break ICs. When applying power voltage, check the electrical features beforehand and be careful. Always turn off the power to the LC module controller before removing or inserting the LC module input connector. If the input connector is removed or inserted while the power is turned on, the LC module internal circuit may break.
2) The display response may be late if the operating temperature is under the normal standard, and the display may be out of order if it is above the normal standard. But this is not a failure; this will be restored if it is within the normal standard.
3) The LCD contrast varies depending on the visual angle, ambient temperature, power voltage etc. Obtain the optimum contrast by adjusting the LC dive voltage.
4) When carrying out the test, do not take the module out of the low-temperature space suddenly. Failure to do so will cause the module condensing, leading to malfunctions.
5) Make certain that each signal noise level is within the standard (L level: 0.2Vdd or less and H level: 0.8Vdd or more) even if the module has functioned properly. If it is beyond the standard, the module may often malfunction. In addition, always connect the module when making noise level measurements.
6) The CMOS ICs are incorporated in the module and the pull-up and pull-down function is not adopted for the input so avoid putting the input signal open while the power is ON.
7) The characteristic of the semiconductor element changes when it is exposed to light emissions, therefore ICs on the LCD may malfunction if they receive light emissions. To prevent these malfunctions, design and assemble ICs so that they are shielded from light emissions.
8) Crosstalk occurs because of characteristics of the LCD. In general, crosstalk occurs when the regularized display is maintained. Also, crosstalk is affected by the LC drive voltage. Design the contents of the display, considering crosstalk.

12.5 Other

1) Do not disassemble or take the LC module into pieces. The LC modules once disassembled or taken into pieces are not the guarantee articles.
2) The residual image may exist if the same display pattern is shown for hours. This residual image, however, disappears when another display pattern is shown or the drive is interrupted and left for a while. But this is not a problem on reliability.
3) AMIPRE will provide one year warrantee for all products and three months warrantee for all repairing products.

13 OUTLINE DIMENSION