LCM Specification

☑Preliminary specification

☐Final Specification

Project No. 项目编号	TFT-H040A3WVTNT0N30
Customer 客户名称	
Module No. 客户型号	
Product type 产品内容	TFT LCD Module 480 x 3RGB x 800 Dots 4.0"TFT LCD

Signature by customer:

客户确认签章:

☐Trial production ☐Mass production

编　　制	电子审核	结构审核	批　　准
Y.L			

深圳市鑫洪泰电子科技有限公司

Shenzhen Hot Display Technology Co.,Ltd

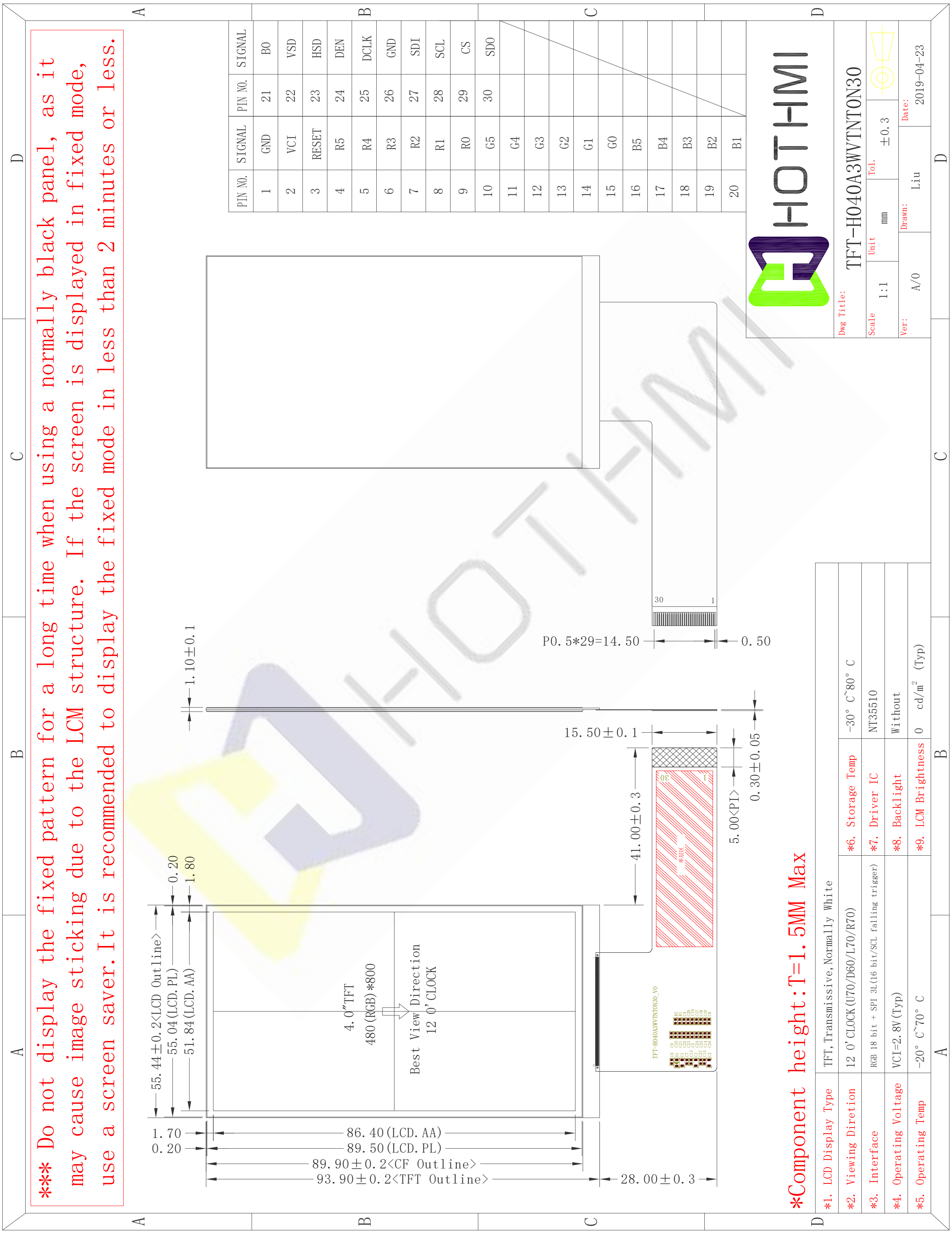
1 Document revision history :

DOCUMENT REVISION	DATE	DESCRIPTION	PREPARED BY	APPROVED BY
0	2019-7-25	First Release.	Y.L	

1. General Feature:

Item	Standard Value	Unit
Display Size	4.0”	--
Number of Pixels	480(H)x3(RGB)*800(V)	--
Active Area	51.84(H) *86.40(V)	mm
Outline Dimension	55.44(H) ×93.90(V) × 1.10(D)	mm
Viewing Direction	12 O'clock	-
Interface	RGB 18 bit + SPI 3L(16 bit/SCL falling trigger)	-
LCM Driver IC	NT35510	-
LCM Driver Condition	VCI=2.8V	V
Backlight	Without Backlight	-
Touch Panel	Without Touch Panel	-
CTP Driver IC	- - -	
CTP Driver Condition	- - -	
Operation Temperature	-20~70	℃
Storage Temperature	-30~80	℃

2.Outline Dimensions

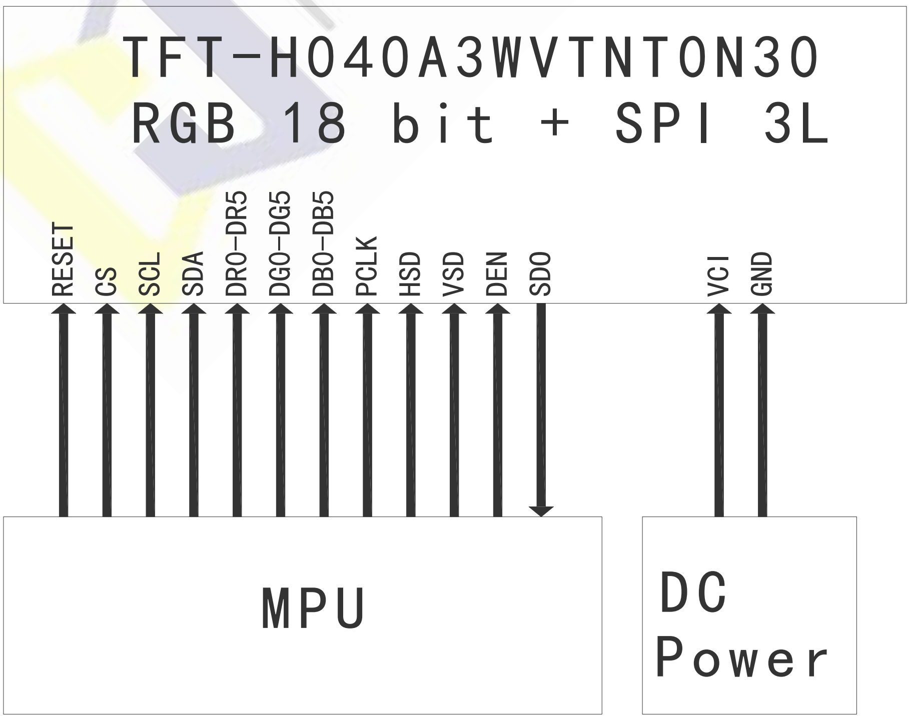


3. Pin Description

3.1 Pin Description

Pin NO.	Symbol	Description
1	GND	Ground
2	VCI	Analog Power(3.0 ~ 3.6 V)
3	RESET	Reset input
4-9	R5-R0	Data bus.
10-15	G5-G0	Data bus.
16-21	B5-B0	Data bus.
22	VSD	Vertical sync signal.
23	HSD	Horizontal sync signal.
24	DEN	Data input enable. Display access is enabled when DE is “L”.
25	PCLK	Pixel clock input pin
26	GND	Ground
27	SDI	Serial data input
28	SCL	Serial clock input for SPI interface
29	CS	A chip select signal
30	SDO	Serial data output
---END---		

3.2 Wiring Diagram



4. Electrical Characteristics

4-1 TFT LCD Module Operating Conditions

Item	Symbol	Condition	Min	Type	Max	Uint
Interface logic circuits	IOVCC	-	1.65	1.8	3.6	V
Analog Power supply	VCI	-	2.65	2.8	3.6	V
TFT Gate on voltage	VGH	-	10.0	-	16.0	V
TFT Gate off voltage	VGL	-	-16.0	-	-10.0	V

4-2 LED back light specification (pera chip)

Item	Symbol	Condition	Min	Type	Max	Uint
Forward voltage	Vt	If=20mA /1-chip	-	-	-	V
Forward current	Ipn		-	-	-	mA
Luminance(With LCD)	Lv	If=20mA	-	-	-	cd/m²
Luminous color	White					

4-3 CTP Operating Conditions

Item	Symbol	Condition	Min	Type	Max	Uint
Power Supply Voltages	VDD	-	2.8	3.30	3.60	V
I/O Digital Voltage	IOVDD	-	2.8	3.30	3.60	V

5. OPTICAL SPECIFICATION

5.1 Overview

The test of Optical specifications shall be measured in a dark room (ambient luminance 1lux and temperature = 25 2℃) with the equipment of Luminance meter system (Goniometer system and TOPCON BM-5) and test unit shall be located at an approximate distance 50cm from the LCD surface at a viewing angle of θ and Φ equal to 0 . The center of the measuring spot on the Display surface shall stay fixed. The backlight should be operating for 30 minutes prior to measurement.

5.2 Optical Specifications

Parameter		Symbol	Condition	Min.	Typ.	Max.	Unit	Remark
Viewing Angle Range	Horizontal	ΘL	CR>10	-	70	-	Deg.	Note 1
		ΘR		-	70	-	Deg.	
	Vertical	ΘU		-	70	-	Deg.	
		ΘD		-	60	-	Deg.	
Contrast ratio		CR	Θ =0°	-	700	-		Note2
Color Gamut		CG		-	55.6	-	%	
White Chromaticity		Wx	Θ =0°	-0.02	(0.308)	+0.02		
		Wy			(0.328)			
Reproduction of color	Red	Rx			(0.608)			Note4 (Based on C Light)
		Ry			(0.330)			
	Green	Gx			(0.302)			
		Gy			(0.565)			
	Blue	Bx			(0.144)			
		By			(0.111)			
Response Time (Rising + Falling)		Tr+Tf	Θ =0° Ta= 25℃	-	25	40	ms	Note5
Transmittance(with Polarizer)		Tr		-	(4.0)	-	%	Note3

Note:

1.Viewing angle is the angle at which the contrast ratio is greater than 10. The viewing angles are determined for the horizontal or 3, 9 o’clock direction and the vertical or 6, 12 o’clock direction with respect to the optical axis which is normal to the LCD surface (see FIGURE 1).

2.Contrast measurements shall be made at viewing angle of Θ= 0 and at the center of the LCD surface. Luminance shall be measured with all pixels in the view field set first to white, then to the dark (black) state . (see FIGUR 1) Luminance Contrast Ratio (CR) is defined mathematically.

$$CR = \frac{\text{Luminance when displaying a white raster}}{\text{Luminance when displaying a black raster}}$$

3.Transmittance is the Value without APF and without CG.

4.The color chromaticity coordinates specified in the above table shall be calculated from the spectral data measured with all pixels first in red, green, blue and white. Measurements shall be made at the center of the panel.

5.The electro-optical response time measurements shall be made as FIGURE 2 by switching the "data" input signal ON and OFF. The times needed for the luminance to change from 10% to 90% is Tr, and 90% to 10% is Tf.

Figure1 Measurement Set Up

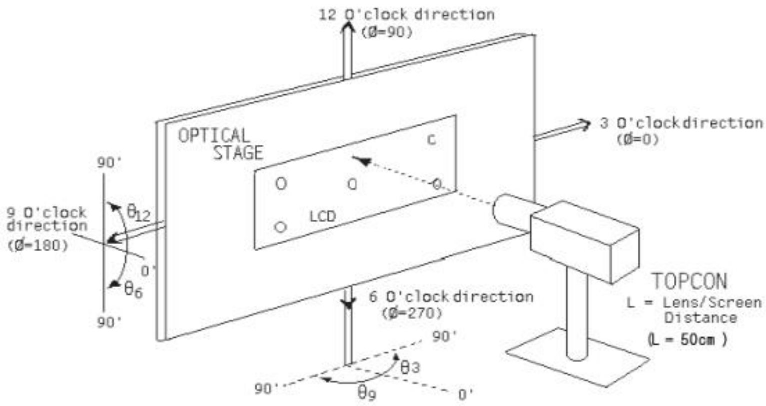
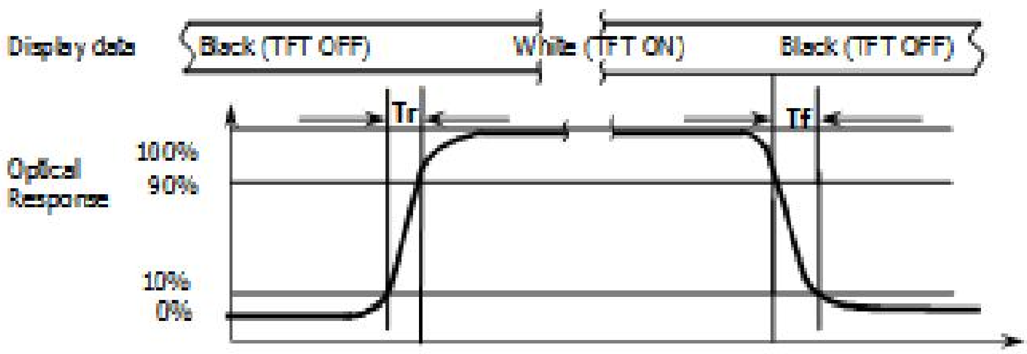
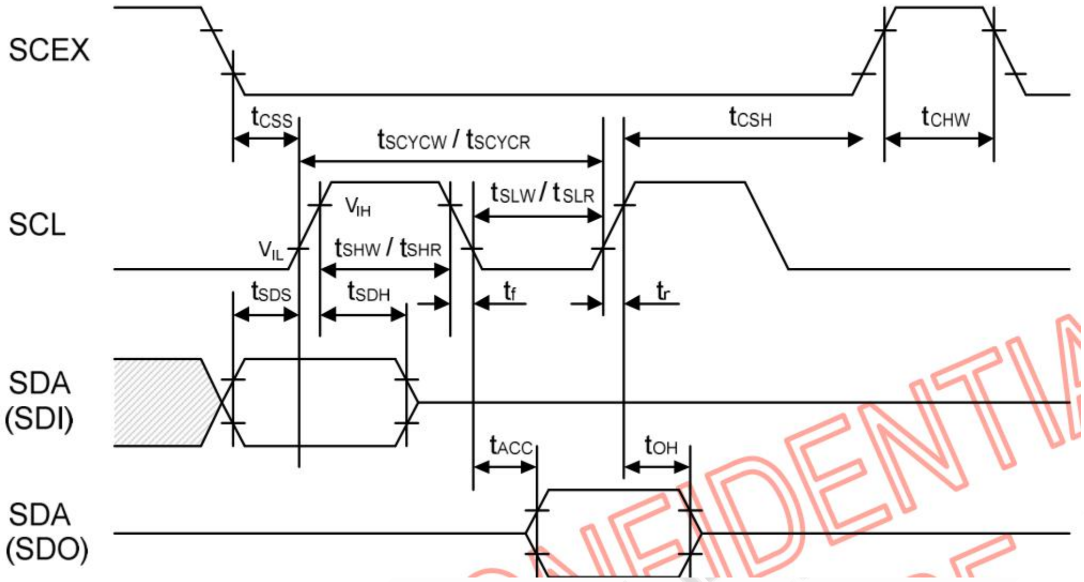


Figure2 Response Time Testing



6.Timing Characteristics of Input Signals

6-1 Serial Interface Characteristics (3-line serial)

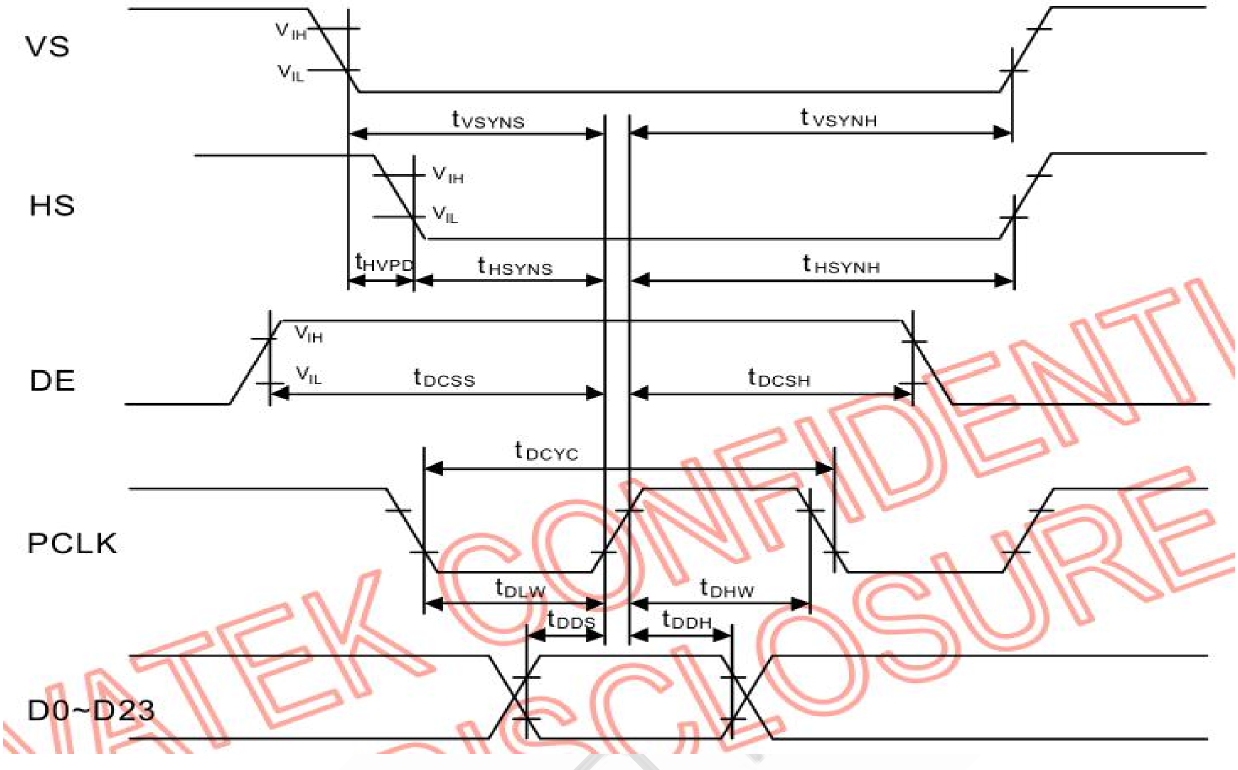


(VSS=VSSI=DVSS=0V, VDDI=1.65V to 3.3V, VDD=2.3V to 4.8V,Ta = -30 to 70˚C)

Signal	Symbol	Parameter	MIN	MAX	Unit	Description
SCL	t_{SCYCW}	Serial clock cycle (Write)	100	-	ns	
	t_{SHW}	SCL "H" pulse width (Write)	40	-	ns	
	t_{SLW}	SCL "L" pulse width (Write)	40	-	ns	
	t_{SCYCR}	Serial clock cycle (Read GRAM)	300	-	ns	
	t_{SHR}	SCL "H" pulse width (Read GRAM)	140	-	ns	
	t_{SLR}	SCL "L" pulse width (Read GRAM)	140	-	ns	
	t_{SCYCR}	Serial clock cycle (Read ID)	300	-	ns	
	t_{SHR}	SCL "H" pulse width (Read ID)	140	-	ns	
	t_{SLR}	SCL "L" pulse width (Read ID)	140	-	ns	
SDI (SDO)	t_{SDS}	Data setup time	20	-	ns	
	t_{SDH}	Data hold time	20	-	ns	
	t_{ACC}	Access time	-	120	ns	
	t_{OH}	Output disable time	5	-	ns	
CSX	t_{CHW}	Chip select "H" pulse width	45	-	ns	
	t_{CSS}	Chip select setup time	20	-	ns	
	t_{CSH}	Chip select hold time	50	-	ns	

Note 1) VDDI=1.65 to 3.3V, VDD=2.3 to 4.8V, VSS=VSSI=DVSS=0V, Ta=-30 to 70 ºC (to +85 ºC no damage)
VDD means VDDA, VDDR, VDDB and VSS means VSSA, VSSR, VSSB
Note 2) The input signal rise time and fall time (tr, tf) is specified at 15 ns or less.

6-2 RGB Interface Characteristics



(VSS=VSSI=DVSS=0V, VDDI=1.65V to 3.3V, VDD=2.3V to 4.8V,Ta = -30 to 70˚C)

Signal	Symbol	Parameter	MIN	TYP	MAX	Unit	Description
VS	t_{VSYNS}	VSYNC setup time	10	-	-	ns	
	t_{VSYNH}	VSYNC hold time	10	-	-	ns	
HS	t_{HSYNS}	HSYNC setup time	10	-	-	ns	
	t_{SCYCR}	HSYNC hold time	10	-	-	ns	
	t_{HVPD}	HSYNC to VSYNC falling edge	400	-	-	ns	
PCLK	t_{DCYC}	PCLK cycle time	33	-	125	ns	
	t_{DLW}	PCLK "L" pulse width	11	-	-	ns	
	t_{DHW}	PCLK "H" pulse width	11	-	-	ns	
	f_{DFREQ}	PCLK frequency	8	-	30	MHz	
DE	t_{DCSS}	DE setup time	10	-	-	ns	
	t_{DCSH}	DE hold Time	10	-	-	ns	
D0~D23	t_{DDS}	RGB Data setup time	10	-	-	ns	
	t_{DDH}	RGB Data hold time	10	-	-	ns	

Note 1) VDDI=1.65 to 3.3V, VDD=2.3 to 4.8V, VSS=VSSI=DVSS=0V, Ta=-30 to 70 ºC (to +85 ºC no damage)
VDD means VDDA, VDDR, VDDB and VSS means VSSA, VSSR, VSSB

Note 2) The input signal rise time and fall time (tr, tf) is specified at 15 ns or less.

6-3 RGB Interface Mode Set

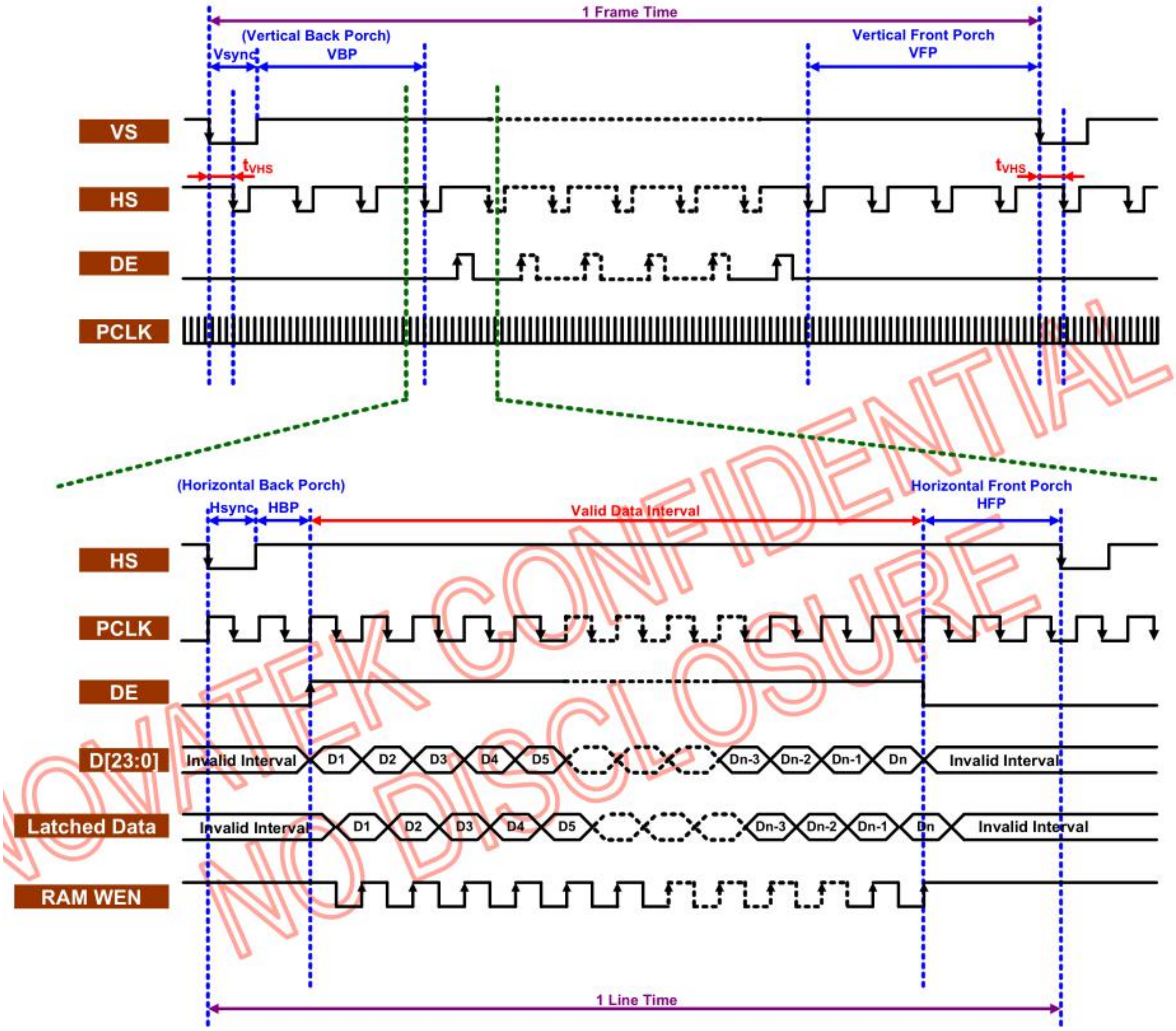
RGB I/F Mode	PCLK	DE	D23-D0	VS	HS	Register VFP[7:0], VBP[7:0] HFP[7:0], HBP[7:0]
RGB Mode 1 (SYNC + DE)	Used	Used	Used	Used	Used	Not used
RGB Mode 2 (SYNC only)	Used	Not used	Used	Used	Used	Used

In RGB Mode 1, writing data to line buffer is done by PCLK and Video Data Bus (D23 to D0), when DE is high state. The external clocks (PCLK, VS and HS) are used for internal displaying clock. So, controller must always transfer PCLK, VS and HS signal to NT35510 DDI.

In RGB Mode 2, back porch of Vsync VBP is defined by VBP[7:0] of RGBCTR command. And back porch of Hsync HBP is defined by HBP[7:0] of RGRCTR command. Front porch of Vsync VFP is defined by VFP[7:0] of RGBCTR command. And front porch of Hsync HFP is defined by HFP[7:0] of RGBCTR command.

Note: VBP[7:0]=Vsync+VBP and HBP[7:0]=Hsync+HBP.

6-3-1 RGB Mode 1 Interface



Notes1： Constraint:

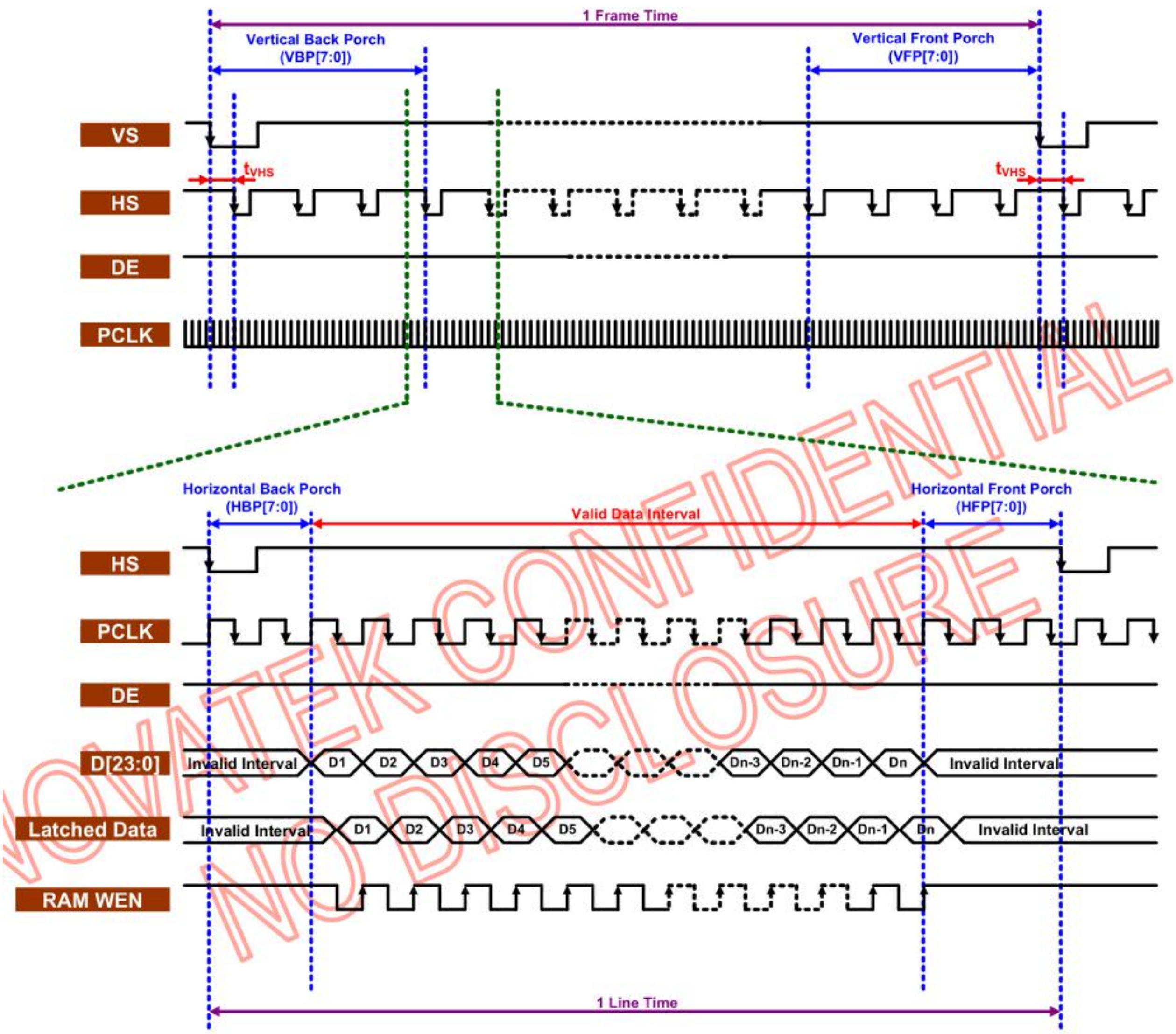
V-Back Porch (VBP[7:0]) ≧ 5 HS lines, V-Front Porch (VFP[7:0]) ≧ 2 HS lines

VBP[7:0]+VFP[7:0] (porch of RGB signal) > VBPA/B/C[7:0] (internal display back porch)

H-Back Porch (HBP[7:0]) ≧ 5 PCLK clocks, H-Back Porch (HFP[7:0]) ≧ 2 PCLK clocks

Notes2： t VHS ≧ 400ns

6-3-2 RGB Mode 2 Interface



Notes1： Constraint:

V-Back Porch (VBP[7:0]) ≧ 5 HS lines, V-Front Porch (VFP[7:0]) ≧ 2 HS lines

VBP[7:0]+VFP[7:0] (porch of RGB signal) > VBPA/B/C[7:0] (internal display back porch)

H-Back Porch (HBP[7:0]) ≧ 5 PCLK clocks, H-Back Porch (HFP[7:0]) ≧ 2 PCLK clocks

Notes2： t VHS ≧ 400ns

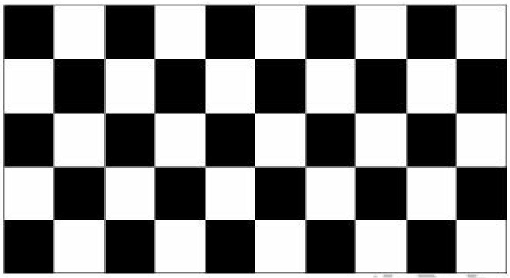
7. RELIABILITY TEST

7-1 Temperature and Humidity

TEST ITEMS	CONDITIONS	NOTE
High Temperature Storage	Ta=+80 o C, 240hrs	
Low Temperature Storage	Ta=-30 o C, 240hrs	
High Temperature Operation	Ta=+70 o C, 240hrs	
Low Temperature Operation	Ta=-20 o C, 240hrs	
High Temperature and High Humidity (Operating)	Ta=+60 o C, 90%RH, 240hrs	

Note: (1) All tests above are practiced at module type.

(2) There is no display function NG issue occurred, all the cosmetic specification is judged before the reliability stress.



(a) Test Pattern (chess board Pattern)

(b) Gray Pattern

7-2 Shock and Vibration

ITEMS	CONDITIONS
Packing Shock (Non-Operation)	● Shock level:980m/s^2 ● Waveform:1/2 Sine wave,6msec ● ±X, ±Y ±Z,each axis 1 times
Packing Vibration (Non-Operation)	● Frequency range:8-33.3HZ ● Stoke:1.0mm ● Sweep: 10Hz-50Hz ● x,y,z 2 hours for each direction

7-3 Electrostatic Discharge

TEST ITEM	CONDITIONS
ESD (Non-operation)	150pF,330Ω， Contact±4KV,Air :±8KV.Note 1
	200pF,0Ω， ±200V Contact test.Note 2

Note:Measure Point:

1.LCD glass and metal bezel

2.IF connector pins

8. HANDDLING & CAUTIONS

8-1 Caution For Operation

♦Since the LCM is made of glass, do not apply strong mechanical impact or static load onto it.Handling with care since shock, vibration, and careless handling may seriously affect the product.If it falls from a high place or receives a strong shock, the glass maybe broken.

♦It is indispensable to drive the LCM within the specified voltage limit since the higher voltage than the limit causes LCM's life shorter. An electro-chemical reaction due to DC causes undesirable deterioration of the LCM so that the use of DC drive should avoid.

♦Do not connect or disconnect the LCM to or from the system when power is on.

♦Never use the LCM under abnormal conditions of high temperature and high humidity.

♦When expose to drastic fluctuation of temperature(hot to cold or cold to hot), the LCM may be affected; specifically, drastic temperature fluctuation from cold to hot, produces dew on the LCM's surface which may affect the operation of the polarizer on the LCM.

♦Response time will be extremely delay at lower temperature than the operating temperature range and on the other hand LCM may turn black at temperature above its operational range. However those phenomenon do not mean malfunction or out of order with the LCM. The LCM will revert to normal operation once the temperature returns to the recommended temperature range for normal operation.

♦Do not display the fixed pattern for a long time when using a normally black panel, as it may cause image sticking due to the LCM structure. If the screen is displayed in fixed mode, use a screen saver.It is recommended to display the fixed mode in less than 2 minutes or less.

♦Do not disassemble and/or re-assemble LCM module

7-2 Caution Against Static Charge

♦The LCM use C-MOS LSI drivers, so customers are recommended that any unused input terminal would be connected to Vdd or Vss, do not input any signals before power is turn on, and ground you body, work/assembly area, assembly equipments to protect against static electricity.

♦Remove the protective film slowly, keeping the removing direction approximate 30-degree not vertical from panel surface, if possible, under ESD control device like ion blower, and the humidity of working room should be kept over 50%RH to reduce the risk of static charge.

♦Avoid the use work clothing made of synthetic fibers. We recommend cotton clothing or other conductivity-treated fibers.

♦In handling the LCM, wear non-charged material gloves. And the conducting wrist to the earth and the conducting shoes to the earth are necessary

9. LCD display initialization code

```
Void Panel_initial_code(void)
{
//--------------------Reset sequence----------//
LCD_RESET=1;
Delayms(1); //Delay 1ms
LCD_RESET=0;
Delayms(10); //Delay 10ms
LCD_RESET=1;
Delayms(120); //Delay 120ms
//-------------------------------------------------//
SPI_WriteComm(0xF000);    SPI_WriteData(0x55);
SPI_WriteComm(0xF001);    SPI_WriteData(0xAA);
SPI_WriteComm(0xF002);    SPI_WriteData(0x52);
SPI_WriteComm(0xF003);    SPI_WriteData(0x08);
SPI_WriteComm(0xF004);    SPI_WriteData(0x01);

SPI_WriteComm(0xB000);    SPI_WriteData(0x0D);
SPI_WriteComm(0xB001);    SPI_WriteData(0x0D);
SPI_WriteComm(0xB002);    SPI_WriteData(0x0D);

SPI_WriteComm(0xB100);    SPI_WriteData(0x0D);
SPI_WriteComm(0xB101);    SPI_WriteData(0x0D);
SPI_WriteComm(0xB102);    SPI_WriteData(0x0D);

SPI_WriteComm(0xB200);    SPI_WriteData(0x00);
SPI_WriteComm(0xB201);    SPI_WriteData(0x00);
SPI_WriteComm(0xB202);    SPI_WriteData(0x00);

SPI_WriteComm(0xB600);    SPI_WriteData(0x34);
SPI_WriteComm(0xB601);    SPI_WriteData(0x34);
SPI_WriteComm(0xB602);    SPI_WriteData(0x34);

SPI_WriteComm(0xB700);    SPI_WriteData(0x34);
SPI_WriteComm(0xB701);    SPI_WriteData(0x34);
SPI_WriteComm(0xB702);    SPI_WriteData(0x34);

SPI_WriteComm(0xB800);    SPI_WriteData(0x24);
SPI_WriteComm(0xB801);    SPI_WriteData(0x24);
```

```
SPI_WriteComm(0xB802);    SPI_WriteData(0x24);
SPI_WriteComm(0xBF00);    SPI_WriteData(0x01);

SPI_WriteComm(0xB300);    SPI_WriteData(0x0F);    //08
SPI_WriteComm(0xB301);    SPI_WriteData(0x0F);    //08
SPI_WriteComm(0xB302);    SPI_WriteData(0x0F);    //08

SPI_WriteComm(0xB900);    SPI_WriteData(0x34);
SPI_WriteComm(0xB901);    SPI_WriteData(0x34);
SPI_WriteComm(0xB902);    SPI_WriteData(0x34);

SPI_WriteComm(0xB500);    SPI_WriteData(0x08);
SPI_WriteComm(0xB501);    SPI_WriteData(0x08);
SPI_WriteComm(0xB502);    SPI_WriteData(0x08);

SPI_WriteComm(0xC200);    SPI_WriteData(0x03);

SPI_WriteComm(0xBA00);    SPI_WriteData(0x24);
SPI_WriteComm(0xBA01);    SPI_WriteData(0x24);
SPI_WriteComm(0xBA02);    SPI_WriteData(0x24);

SPI_WriteComm(0xBC00);    SPI_WriteData(0x00);
SPI_WriteComm(0xBC01);    SPI_WriteData(0x78);
SPI_WriteComm(0xBC02);    SPI_WriteData(0x00);

SPI_WriteComm(0xBD00);    SPI_WriteData(0x00);
SPI_WriteComm(0xBD01);    SPI_WriteData(0x78);
SPI_WriteComm(0xBD02);    SPI_WriteData(0x00);

SPI_WriteComm(0xBE00);    SPI_WriteData(0x00);
SPI_WriteComm(0xBE01);    SPI_WriteData(0x8b);
//Gamma (R+)
SPI_WriteComm(0xD100);    SPI_WriteData(0x00);
SPI_WriteComm(0xD101);    SPI_WriteData(0x06);
SPI_WriteComm(0xD102);    SPI_WriteData(0x00);
SPI_WriteComm(0xD103);    SPI_WriteData(0x07);
SPI_WriteComm(0xD104);    SPI_WriteData(0x00);
SPI_WriteComm(0xD105);    SPI_WriteData(0x0E);
SPI_WriteComm(0xD106);    SPI_WriteData(0x00);
SPI_WriteComm(0xD107);    SPI_WriteData(0x22);
```

```
SPI_WriteComm(0xD108);    SPI_WriteData(0x00);
SPI_WriteComm(0xD109);    SPI_WriteData(0x3B);
SPI_WriteComm(0xD10A);    SPI_WriteData(0x00);
SPI_WriteComm(0xD10B);    SPI_WriteData(0x71);
SPI_WriteComm(0xD10C);    SPI_WriteData(0x00);
SPI_WriteComm(0xD10D);    SPI_WriteData(0x9F);
SPI_WriteComm(0xD10E);    SPI_WriteData(0x00);
SPI_WriteComm(0xD10F);    SPI_WriteData(0xE2);
SPI_WriteComm(0xD110);    SPI_WriteData(0x01);
SPI_WriteComm(0xD111);    SPI_WriteData(0x12);
SPI_WriteComm(0xD112);    SPI_WriteData(0x01);
SPI_WriteComm(0xD113);    SPI_WriteData(0x57);
SPI_WriteComm(0xD114);    SPI_WriteData(0x01);
SPI_WriteComm(0xD115);    SPI_WriteData(0x88);
SPI_WriteComm(0xD116);    SPI_WriteData(0x01);
SPI_WriteComm(0xD117);    SPI_WriteData(0xCE);
SPI_WriteComm(0xD118);    SPI_WriteData(0x02);
SPI_WriteComm(0xD119);    SPI_WriteData(0x07);
SPI_WriteComm(0xD11A);    SPI_WriteData(0x02);
SPI_WriteComm(0xD11B);    SPI_WriteData(0x08);
SPI_WriteComm(0xD11C);    SPI_WriteData(0x02);
SPI_WriteComm(0xD11D);    SPI_WriteData(0x39);
SPI_WriteComm(0xD11E);    SPI_WriteData(0x02);
SPI_WriteComm(0xD11F);    SPI_WriteData(0x6C);
SPI_WriteComm(0xD120);    SPI_WriteData(0x02);
SPI_WriteComm(0xD121);    SPI_WriteData(0x87);
SPI_WriteComm(0xD122);    SPI_WriteData(0x02);
SPI_WriteComm(0xD123);    SPI_WriteData(0xA6);
SPI_WriteComm(0xD124);    SPI_WriteData(0x02);
SPI_WriteComm(0xD125);    SPI_WriteData(0xBA);
SPI_WriteComm(0xD126);    SPI_WriteData(0x02);
SPI_WriteComm(0xD127);    SPI_WriteData(0xD2);
SPI_WriteComm(0xD128);    SPI_WriteData(0x02);
SPI_WriteComm(0xD129);    SPI_WriteData(0xE2);
SPI_WriteComm(0xD12A);    SPI_WriteData(0x02);
SPI_WriteComm(0xD12B);    SPI_WriteData(0xF7);
SPI_WriteComm(0xD12C);    SPI_WriteData(0x03);
SPI_WriteComm(0xD12D);    SPI_WriteData(0x06);
SPI_WriteComm(0xD12E);    SPI_WriteData(0x03);
SPI_WriteComm(0xD12F);    SPI_WriteData(0x1E);
```

```
SPI_WriteComm(0xD130);    SPI_WriteData(0x03);
SPI_WriteComm(0xD131);    SPI_WriteData(0x55);
SPI_WriteComm(0xD132);    SPI_WriteData(0x03);
SPI_WriteComm(0xD133);    SPI_WriteData(0xFF);

//Gamma (G+)
SPI_WriteComm(0xD200);    SPI_WriteData(0x00);
SPI_WriteComm(0xD201);    SPI_WriteData(0x06);
SPI_WriteComm(0xD202);    SPI_WriteData(0x00);
SPI_WriteComm(0xD203);    SPI_WriteData(0x07);
SPI_WriteComm(0xD204);    SPI_WriteData(0x00);
SPI_WriteComm(0xD205);    SPI_WriteData(0x0E);
SPI_WriteComm(0xD206);    SPI_WriteData(0x00);
SPI_WriteComm(0xD207);    SPI_WriteData(0x22);
SPI_WriteComm(0xD208);    SPI_WriteData(0x00);
SPI_WriteComm(0xD209);    SPI_WriteData(0x3B);
SPI_WriteComm(0xD20A);    SPI_WriteData(0x00);
SPI_WriteComm(0xD20B);    SPI_WriteData(0x71);
SPI_WriteComm(0xD20C);    SPI_WriteData(0x00);
SPI_WriteComm(0xD20D);    SPI_WriteData(0x9F);
SPI_WriteComm(0xD20E);    SPI_WriteData(0x00);
SPI_WriteComm(0xD20F);    SPI_WriteData(0xE2);
SPI_WriteComm(0xD210);    SPI_WriteData(0x01);
SPI_WriteComm(0xD211);    SPI_WriteData(0x12);
SPI_WriteComm(0xD212);    SPI_WriteData(0x01);
SPI_WriteComm(0xD213);    SPI_WriteData(0x57);
SPI_WriteComm(0xD214);    SPI_WriteData(0x01);
SPI_WriteComm(0xD215);    SPI_WriteData(0x88);
SPI_WriteComm(0xD216);    SPI_WriteData(0x01);
SPI_WriteComm(0xD217);    SPI_WriteData(0xCE);
SPI_WriteComm(0xD218);    SPI_WriteData(0x02);
SPI_WriteComm(0xD219);    SPI_WriteData(0x07);
SPI_WriteComm(0xD21A);    SPI_WriteData(0x02);
SPI_WriteComm(0xD21B);    SPI_WriteData(0x08);
SPI_WriteComm(0xD21C);    SPI_WriteData(0x02);
SPI_WriteComm(0xD21D);    SPI_WriteData(0x39);
SPI_WriteComm(0xD21E);    SPI_WriteData(0x02);
SPI_WriteComm(0xD21F);    SPI_WriteData(0x6C);
SPI_WriteComm(0xD220);    SPI_WriteData(0x02);
SPI_WriteComm(0xD221);    SPI_WriteData(0x87);
```

```
SPI_WriteComm(0xD222);    SPI_WriteData(0x02);
SPI_WriteComm(0xD223);    SPI_WriteData(0xA6);
SPI_WriteComm(0xD224);    SPI_WriteData(0x02);
SPI_WriteComm(0xD225);    SPI_WriteData(0xBA);
SPI_WriteComm(0xD226);    SPI_WriteData(0x02);
SPI_WriteComm(0xD227);    SPI_WriteData(0xD2);
SPI_WriteComm(0xD228);    SPI_WriteData(0x02);
SPI_WriteComm(0xD229);    SPI_WriteData(0xE2);
SPI_WriteComm(0xD22A);    SPI_WriteData(0x02);
SPI_WriteComm(0xD22B);    SPI_WriteData(0xF7);
SPI_WriteComm(0xD22C);    SPI_WriteData(0x03);
SPI_WriteComm(0xD22D);    SPI_WriteData(0x06);
SPI_WriteComm(0xD22E);    SPI_WriteData(0x03);
SPI_WriteComm(0xD22F);    SPI_WriteData(0x1E);
SPI_WriteComm(0xD230);    SPI_WriteData(0x03);
SPI_WriteComm(0xD231);    SPI_WriteData(0x55);
SPI_WriteComm(0xD232);    SPI_WriteData(0x03);
SPI_WriteComm(0xD233);    SPI_WriteData(0xFF);

//Gamma (B+)
SPI_WriteComm(0xD300);    SPI_WriteData(0x00);
SPI_WriteComm(0xD301);    SPI_WriteData(0x06);
SPI_WriteComm(0xD302);    SPI_WriteData(0x00);
SPI_WriteComm(0xD303);    SPI_WriteData(0x07);
SPI_WriteComm(0xD304);    SPI_WriteData(0x00);
SPI_WriteComm(0xD305);    SPI_WriteData(0x0E);
SPI_WriteComm(0xD306);    SPI_WriteData(0x00);
SPI_WriteComm(0xD307);    SPI_WriteData(0x22);
SPI_WriteComm(0xD308);    SPI_WriteData(0x00);
SPI_WriteComm(0xD309);    SPI_WriteData(0x3B);
SPI_WriteComm(0xD30A);    SPI_WriteData(0x00);
SPI_WriteComm(0xD30B);    SPI_WriteData(0x71);
SPI_WriteComm(0xD30C);    SPI_WriteData(0x00);
SPI_WriteComm(0xD30D);    SPI_WriteData(0x9F);
SPI_WriteComm(0xD30E);    SPI_WriteData(0x00);
SPI_WriteComm(0xD30F);    SPI_WriteData(0xE2);
SPI_WriteComm(0xD310);    SPI_WriteData(0x01);
SPI_WriteComm(0xD311);    SPI_WriteData(0x12);
SPI_WriteComm(0xD312);    SPI_WriteData(0x01);
SPI_WriteComm(0xD313);    SPI_WriteData(0x57);
```

```
SPI_WriteComm(0xD314);    SPI_WriteData(0x01);
SPI_WriteComm(0xD315);    SPI_WriteData(0x88);
SPI_WriteComm(0xD316);    SPI_WriteData(0x01);
SPI_WriteComm(0xD317);    SPI_WriteData(0xCE);
SPI_WriteComm(0xD318);    SPI_WriteData(0x02);
SPI_WriteComm(0xD319);    SPI_WriteData(0x07);
SPI_WriteComm(0xD31A);    SPI_WriteData(0x02);
SPI_WriteComm(0xD31B);    SPI_WriteData(0x08);
SPI_WriteComm(0xD31C);    SPI_WriteData(0x02);
SPI_WriteComm(0xD31D);    SPI_WriteData(0x39);
SPI_WriteComm(0xD31E);    SPI_WriteData(0x02);
SPI_WriteComm(0xD31F);    SPI_WriteData(0x6C);
SPI_WriteComm(0xD320);    SPI_WriteData(0x02);
SPI_WriteComm(0xD321);    SPI_WriteData(0x87);
SPI_WriteComm(0xD322);    SPI_WriteData(0x02);
SPI_WriteComm(0xD323);    SPI_WriteData(0xA6);
SPI_WriteComm(0xD324);    SPI_WriteData(0x02);
SPI_WriteComm(0xD325);    SPI_WriteData(0xBA);
SPI_WriteComm(0xD326);    SPI_WriteData(0x02);
SPI_WriteComm(0xD327);    SPI_WriteData(0xD2);
SPI_WriteComm(0xD328);    SPI_WriteData(0x02);
SPI_WriteComm(0xD329);    SPI_WriteData(0xE2);
SPI_WriteComm(0xD32A);    SPI_WriteData(0x02);
SPI_WriteComm(0xD32B);    SPI_WriteData(0xF7);
SPI_WriteComm(0xD32C);    SPI_WriteData(0x03);
SPI_WriteComm(0xD32D);    SPI_WriteData(0x06);
SPI_WriteComm(0xD32E);    SPI_WriteData(0x03);
SPI_WriteComm(0xD32F);    SPI_WriteData(0x1E);
SPI_WriteComm(0xD330);    SPI_WriteData(0x03);
SPI_WriteComm(0xD331);    SPI_WriteData(0x55);
SPI_WriteComm(0xD332);    SPI_WriteData(0x03);
SPI_WriteComm(0xD333);    SPI_WriteData(0xFF);

//Gamma (R-)
SPI_WriteComm(0xD400);    SPI_WriteData(0x00);
SPI_WriteComm(0xD401);    SPI_WriteData(0x06);
SPI_WriteComm(0xD402);    SPI_WriteData(0x00);
SPI_WriteComm(0xD403);    SPI_WriteData(0x07);
SPI_WriteComm(0xD404);    SPI_WriteData(0x00);
SPI_WriteComm(0xD405);    SPI_WriteData(0x0E);
```

```
SPI_WriteComm(0xD406);    SPI_WriteData(0x00);
SPI_WriteComm(0xD407);    SPI_WriteData(0x22);
SPI_WriteComm(0xD408);    SPI_WriteData(0x00);
SPI_WriteComm(0xD409);    SPI_WriteData(0x3B);
SPI_WriteComm(0xD40A);    SPI_WriteData(0x00);
SPI_WriteComm(0xD40B);    SPI_WriteData(0x71);
SPI_WriteComm(0xD40C);    SPI_WriteData(0x00);
SPI_WriteComm(0xD40D);    SPI_WriteData(0x9F);
SPI_WriteComm(0xD40E);    SPI_WriteData(0x00);
SPI_WriteComm(0xD40F);    SPI_WriteData(0xE2);
SPI_WriteComm(0xD410);    SPI_WriteData(0x01);
SPI_WriteComm(0xD411);    SPI_WriteData(0x12);
SPI_WriteComm(0xD412);    SPI_WriteData(0x01);
SPI_WriteComm(0xD413);    SPI_WriteData(0x57);
SPI_WriteComm(0xD414);    SPI_WriteData(0x01);
SPI_WriteComm(0xD415);    SPI_WriteData(0x88);
SPI_WriteComm(0xD416);    SPI_WriteData(0x01);
SPI_WriteComm(0xD417);    SPI_WriteData(0xCE);
SPI_WriteComm(0xD418);    SPI_WriteData(0x02);
SPI_WriteComm(0xD419);    SPI_WriteData(0x07);
SPI_WriteComm(0xD41A);    SPI_WriteData(0x02);
SPI_WriteComm(0xD41B);    SPI_WriteData(0x08);
SPI_WriteComm(0xD41C);    SPI_WriteData(0x02);
SPI_WriteComm(0xD41D);    SPI_WriteData(0x39);
SPI_WriteComm(0xD41E);    SPI_WriteData(0x02);
SPI_WriteComm(0xD41F);    SPI_WriteData(0x6C);
SPI_WriteComm(0xD420);    SPI_WriteData(0x02);
SPI_WriteComm(0xD421);    SPI_WriteData(0x87);
SPI_WriteComm(0xD422);    SPI_WriteData(0x02);
SPI_WriteComm(0xD423);    SPI_WriteData(0xA6);
SPI_WriteComm(0xD424);    SPI_WriteData(0x02);
SPI_WriteComm(0xD425);    SPI_WriteData(0xBA);
SPI_WriteComm(0xD426);    SPI_WriteData(0x02);
SPI_WriteComm(0xD427);    SPI_WriteData(0xD2);
SPI_WriteComm(0xD428);    SPI_WriteData(0x02);
SPI_WriteComm(0xD429);    SPI_WriteData(0xE2);
SPI_WriteComm(0xD42A);    SPI_WriteData(0x02);
SPI_WriteComm(0xD42B);    SPI_WriteData(0xF7);
SPI_WriteComm(0xD42C);    SPI_WriteData(0x03);
SPI_WriteComm(0xD42D);    SPI_WriteData(0x06);
```

```
SPI_WriteComm(0xD42E);    SPI_WriteData(0x03);
SPI_WriteComm(0xD42F);    SPI_WriteData(0x1E);
SPI_WriteComm(0xD430);    SPI_WriteData(0x03);
SPI_WriteComm(0xD431);    SPI_WriteData(0x55);
SPI_WriteComm(0xD432);    SPI_WriteData(0x03);
SPI_WriteComm(0xD433);    SPI_WriteData(0xFF);

//Gamma (G-)
SPI_WriteComm(0xD500);    SPI_WriteData(0x00);
SPI_WriteComm(0xD501);    SPI_WriteData(0x06);
SPI_WriteComm(0xD502);    SPI_WriteData(0x00);
SPI_WriteComm(0xD503);    SPI_WriteData(0x07);
SPI_WriteComm(0xD504);    SPI_WriteData(0x00);
SPI_WriteComm(0xD505);    SPI_WriteData(0x0E);
SPI_WriteComm(0xD506);    SPI_WriteData(0x00);
SPI_WriteComm(0xD507);    SPI_WriteData(0x22);
SPI_WriteComm(0xD508);    SPI_WriteData(0x00);
SPI_WriteComm(0xD509);    SPI_WriteData(0x3B);
SPI_WriteComm(0xD50A);    SPI_WriteData(0x00);
SPI_WriteComm(0xD50B);    SPI_WriteData(0x71);
SPI_WriteComm(0xD50C);    SPI_WriteData(0x00);
SPI_WriteComm(0xD50D);    SPI_WriteData(0x9F);
SPI_WriteComm(0xD50E);    SPI_WriteData(0x00);
SPI_WriteComm(0xD50F);    SPI_WriteData(0xE2);
SPI_WriteComm(0xD510);    SPI_WriteData(0x01);
SPI_WriteComm(0xD511);    SPI_WriteData(0x12);
SPI_WriteComm(0xD512);    SPI_WriteData(0x01);
SPI_WriteComm(0xD513);    SPI_WriteData(0x57);
SPI_WriteComm(0xD514);    SPI_WriteData(0x01);
SPI_WriteComm(0xD515);    SPI_WriteData(0x88);
SPI_WriteComm(0xD516);    SPI_WriteData(0x01);
SPI_WriteComm(0xD517);    SPI_WriteData(0xCE);
SPI_WriteComm(0xD518);    SPI_WriteData(0x02);
SPI_WriteComm(0xD519);    SPI_WriteData(0x07);
SPI_WriteComm(0xD51A);    SPI_WriteData(0x02);
SPI_WriteComm(0xD51B);    SPI_WriteData(0x08);
SPI_WriteComm(0xD51C);    SPI_WriteData(0x02);
SPI_WriteComm(0xD51D);    SPI_WriteData(0x39);
SPI_WriteComm(0xD51E);    SPI_WriteData(0x02);
SPI_WriteComm(0xD51F);    SPI_WriteData(0x6C);
```

```
SPI_WriteComm(0xD520);    SPI_WriteData(0x02);
SPI_WriteComm(0xD521);    SPI_WriteData(0x87);
SPI_WriteComm(0xD522);    SPI_WriteData(0x02);
SPI_WriteComm(0xD523);    SPI_WriteData(0xA6);
SPI_WriteComm(0xD524);    SPI_WriteData(0x02);
SPI_WriteComm(0xD525);    SPI_WriteData(0xBA);
SPI_WriteComm(0xD526);    SPI_WriteData(0x02);
SPI_WriteComm(0xD527);    SPI_WriteData(0xD2);
SPI_WriteComm(0xD528);    SPI_WriteData(0x02);
SPI_WriteComm(0xD529);    SPI_WriteData(0xE2);
SPI_WriteComm(0xD52A);    SPI_WriteData(0x02);
SPI_WriteComm(0xD52B);    SPI_WriteData(0xF7);
SPI_WriteComm(0xD52C);    SPI_WriteData(0x03);
SPI_WriteComm(0xD52D);    SPI_WriteData(0x06);
SPI_WriteComm(0xD52E);    SPI_WriteData(0x03);
SPI_WriteComm(0xD52F);    SPI_WriteData(0x1E);
SPI_WriteComm(0xD530);    SPI_WriteData(0x03);
SPI_WriteComm(0xD531);    SPI_WriteData(0x55);
SPI_WriteComm(0xD532);    SPI_WriteData(0x03);
SPI_WriteComm(0xD533);    SPI_WriteData(0xFF);

//Gamma (B-)
SPI_WriteComm(0xD600);    SPI_WriteData(0x00);
SPI_WriteComm(0xD601);    SPI_WriteData(0x06);
SPI_WriteComm(0xD602);    SPI_WriteData(0x00);
SPI_WriteComm(0xD603);    SPI_WriteData(0x07);
SPI_WriteComm(0xD604);    SPI_WriteData(0x00);
SPI_WriteComm(0xD605);    SPI_WriteData(0x0E);
SPI_WriteComm(0xD606);    SPI_WriteData(0x00);
SPI_WriteComm(0xD607);    SPI_WriteData(0x22);
SPI_WriteComm(0xD608);    SPI_WriteData(0x00);
SPI_WriteComm(0xD609);    SPI_WriteData(0x3B);
SPI_WriteComm(0xD60A);    SPI_WriteData(0x00);
SPI_WriteComm(0xD60B);    SPI_WriteData(0x71);
SPI_WriteComm(0xD60C);    SPI_WriteData(0x00);
SPI_WriteComm(0xD60D);    SPI_WriteData(0x9F);
SPI_WriteComm(0xD60E);    SPI_WriteData(0x00);
SPI_WriteComm(0xD60F);    SPI_WriteData(0xE2);
SPI_WriteComm(0xD610);    SPI_WriteData(0x01);
SPI_WriteComm(0xD611);    SPI_WriteData(0x12);
```

```
SPI_WriteComm(0xD612);     SPI_WriteData(0x01);
SPI_WriteComm(0xD613);     SPI_WriteData(0x57);
SPI_WriteComm(0xD614);     SPI_WriteData(0x01);
SPI_WriteComm(0xD615);     SPI_WriteData(0x88);
SPI_WriteComm(0xD616);     SPI_WriteData(0x01);
SPI_WriteComm(0xD617);     SPI_WriteData(0xCE);
SPI_WriteComm(0xD618);     SPI_WriteData(0x02);
SPI_WriteComm(0xD619);     SPI_WriteData(0x07);
SPI_WriteComm(0xD61A);     SPI_WriteData(0x02);
SPI_WriteComm(0xD61B);     SPI_WriteData(0x08);
SPI_WriteComm(0xD61C);     SPI_WriteData(0x02);
SPI_WriteComm(0xD61D);     SPI_WriteData(0x39);
SPI_WriteComm(0xD61E);     SPI_WriteData(0x02);
SPI_WriteComm(0xD61F);     SPI_WriteData(0x6C);
SPI_WriteComm(0xD620);     SPI_WriteData(0x02);
SPI_WriteComm(0xD621);     SPI_WriteData(0x87);
SPI_WriteComm(0xD622);     SPI_WriteData(0x02);
SPI_WriteComm(0xD623);     SPI_WriteData(0xA6);
SPI_WriteComm(0xD624);     SPI_WriteData(0x02);
SPI_WriteComm(0xD625);     SPI_WriteData(0xBA);
SPI_WriteComm(0xD626);     SPI_WriteData(0x02);
SPI_WriteComm(0xD627);     SPI_WriteData(0xD2);
SPI_WriteComm(0xD628);     SPI_WriteData(0x02);
SPI_WriteComm(0xD629);     SPI_WriteData(0xE2);
SPI_WriteComm(0xD62A);     SPI_WriteData(0x02);
SPI_WriteComm(0xD62B);     SPI_WriteData(0xF7);
SPI_WriteComm(0xD62C);     SPI_WriteData(0x03);
SPI_WriteComm(0xD62D);     SPI_WriteData(0x06);
SPI_WriteComm(0xD62E);     SPI_WriteData(0x03);
SPI_WriteComm(0xD62F);     SPI_WriteData(0x1E);
SPI_WriteComm(0xD630);     SPI_WriteData(0x03);
SPI_WriteComm(0xD631);     SPI_WriteData(0x55);
SPI_WriteComm(0xD632);     SPI_WriteData(0x03);
SPI_WriteComm(0xD633);     SPI_WriteData(0xFF);

//PAGE0
SPI_WriteComm(0xF000);     SPI_WriteData(0x55);
SPI_WriteComm(0xF001);     SPI_WriteData(0xAA);
SPI_WriteComm(0xF002);     SPI_WriteData(0x52);
SPI_WriteComm(0xF003);     SPI_WriteData(0x08);
```

```
SPI_WriteComm(0xF004);      SPI_WriteData(0x00);

//480x800
SPI_WriteComm(0xB500);      SPI_WriteData(0x50);
//SPI_WriteComm(0x2C00);      SPI_WriteData(0x06); //8BIT 6-6-6?

//Dispay control
SPI_WriteComm(0xB100);      SPI_WriteData(0xCC);
SPI_WriteComm(0xB101);      SPI_WriteData(0x00); // S1->S1440:00;S1440->S1:02

//Source hold time (Nova non-used)
SPI_WriteComm(0xB600);      SPI_WriteData(0x05);

//Gate EQ control       (Nova non-used)
SPI_WriteComm(0xB700);      SPI_WriteData(0x70);
SPI_WriteComm(0xB701);      SPI_WriteData(0x70);

//Source EQ control (Nova non-used)
SPI_WriteComm(0xB800);      SPI_WriteData(0x01);
SPI_WriteComm(0xB801);      SPI_WriteData(0x03);
SPI_WriteComm(0xB802);      SPI_WriteData(0x03);
SPI_WriteComm(0xB803);      SPI_WriteData(0x03);

//Inversion mode: column
SPI_WriteComm(0xBC00);      SPI_WriteData(0x02);   //00: column
SPI_WriteComm(0xBC01);      SPI_WriteData(0x00);   //01:1dot
SPI_WriteComm(0xBC02);      SPI_WriteData(0x00);

//Frame rate (Nova non-used)
SPI_WriteComm(0xBD00);      SPI_WriteData(0x01);
SPI_WriteComm(0xBD01);      SPI_WriteData(0x84);
SPI_WriteComm(0xBD02);      SPI_WriteData(0x1C);
SPI_WriteComm(0xBD03);      SPI_WriteData(0x1C);
SPI_WriteComm(0xBD04);      SPI_WriteData(0x00);

//LGD timing control(4H/4-Delay)
SPI_WriteComm(0xC900);      SPI_WriteData(0xD0);
SPI_WriteComm(0xC901);      SPI_WriteData(0x02);
SPI_WriteComm(0xC902);      SPI_WriteData(0x50);
SPI_WriteComm(0xC903);      SPI_WriteData(0x50);
```

```
SPI_WriteComm(0xC904);    SPI_WriteData(0x50);

SPI_WriteComm(0x3600);    SPI_WriteData(0x00);
SPI_WriteComm(0x3500);    SPI_WriteData(0x00);
SPI_WriteComm(0x3A00);    SPI_WriteData(0x66);

//Sleep out
SPI_WriteComm(0x1100);
Delayms(120);

//Display on
SPI_WriteComm(0x2900);
}
```

```
#define ROW    800              //显示的行、列数
#define COL    480
void BlockWrite(unsigned int Xstart,unsigned int Xend,unsigned int Ystart,unsigned int Yend)
{
   WriteComm(0x2a);
   WriteData(Xstart>>8);
   WriteData(Xstart&0xff);
   WriteData(Xend>>8);
   WriteData(Xend&0xff);
   WriteComm(0x2b);
   WriteData(Ystart>>8);
   WriteData(Ystart&0xff);
   WriteData(Yend>>8);
   WriteData(Yend&0xff);
   WriteComm(0x2c);
}
void DispColor(unsigned int color)
{
   unsigned int i,j;
   BlockWrite(0,COL-1,0,ROW-1);
      CLKSEL = 0x03;
   //CS0=0;
   //RD0=1;
   RS=1;
   DBH=color>>8;
   DBL=color;
   for(i=0;i<ROW;i++)
   {
         for(j=0;j<COL;j++)
         {
               WR0=0;
               WR0=1;
         }
   }
   //CLKSEL = 0x00;
   //CS0=1;
   }
```

--END--