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-640480G2TNQW -TU0H-A
APPROVED BY	
DATE	

☐ Approved For Specifications
☐ Approved For Specifications & Sample

AMPIRE CO., LTD.
4F., No.114, Sec. 1, Xintai 5th Rd., Xizhi Dist., New Taipei City 221,Taiwan (R.O.C.)
新北市汐止區新台五路一段 **114** 號 **4** 樓**(**東方科學園區 **A** 棟**)**

APPROVED BY	CHECKED BY	ORGANIZED BY

RECORD OF REVISION

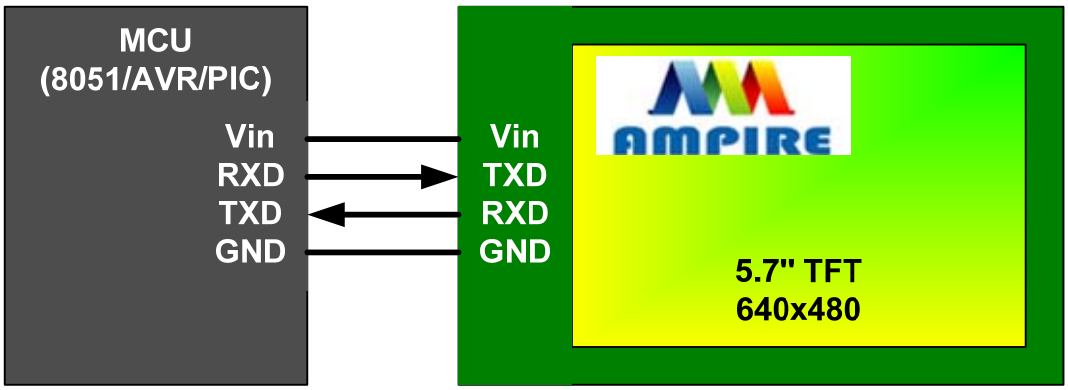
Revision Date	Page	Contents	Editor
2008/07/29	--	New Release	Kokai
2008/9/8		1. Add Touch Panel command & Control Touch Panel flow chart 2. Add Back-light dimming control command 3. Add Jumper setting	Kokai
2009/12/11	--	Change to use new Touch Panel Rename to new P/N : AM-640480G2TNQW-TU0H-A	Edward
2010/8/17	10	Add life time	Patrick
2011/1/18	13	Correct the connector pitch	Patrick
2011/8/26		Add maximum current consumption	Kokai
2011/9/2	7	Redefine Power Voltage Symbol	Rober
	8	Redefine Power Supply Current	

Index

1 Introduction

This is a color active matrix TFT-LCD that uses amorphous silicon TFT as a switching device. This model is composed of a 5.7inch TFT-LCD panel, a driving circuit and LED backlight system.This TFT-LCD has a high resolution (640(R.G.B) X 480) and can display up to 65K colors .
The TFT LCD module is a complete HMI module. It is able to handle the TFT panel by UART (Universal Asynchronous Receiver/Transmitter) interface. The user can easy use/control the TFT Panel by commands.
High Speed ASCI IC and rich function (Text support, draw pixel, draw Line, draw Arc, draw Rectangle) design will make the user's product success.

2 Application Diagram



Any MCU which is with UART interface can control the TFT module.

3 Physical Specifications

Item	Specifications	unit
Display resolution(dot)	640RGB (W) x 480(H)	dots
Display area	116.16 (W) x 87.12 (H)	mm
Pixel pitch	0.18 (W) x 0.18 (H)	mm
Color configuration	R.G.B Vertical stripe	
Overall dimension	144.0(W)x109.5(H)x22.0(D)---(Typ)	mm
Surface treatment	Antiglare , Hard-Coating(3H)	
Brightness	500	cd/m^2
Contrast ratio	250 : 1	
Backlight unit	LED	
Display color	65K	colors
Viewing Direction	12 o'clock (Gray Inversion)	
Display Mode	Normally White	

4 Electrical Specification

4.1 Absolute max. ratings

4.1.1 Electrical Absolute max. ratings

Item	Symbol	Condition	Min.	Max.	Unit	Remark
Power voltage	(Vin)	GND=0	-0.3	30.0	V	
Input voltege			-0.3	3.6	V	Note 1

Note1:RXD,TXD

4.1.2 Environmental Absolute max. ratings

Item	OPERATING		STORAGE		Remark
	MIN	MAX	MIN	MAX	
Temperature	-20	70	-30	80	
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℃

4.2 Electrical characteristics

4.2.1 DC Electrical characteristic

Typical operating conditions (GND=0V)

Item		Symbol	Min.	Typ.	Max.	Unit	Remark
Power supply		Vin	4.6	--	26.0	V	
Input Voltage for logic	H Level	V_{IH}	0.7 VDD	-	VDD	V	Note 1
	L Level	V_{IL}	0	-	0.3 VDD	V	
Power Supply current Vin =5.0V		I_{Vin}		480	580	mA	
Power Supply current Vin =12.0V		Ivin		250	350	mA	
Power Supply current		IDD	--	525	--	mA	Note 2
Power consumption		P		2.5		W	Note 3

Note1:RXD0,TXD0,RXD1,TXD1.

Note2: The internal power circuit will transfer the Vin to VDD=+3.3V for the module power, LED Backlight turn ON.

Note3: Black pattern, Vin = 5.0V, Ta=25℃

Item		Min.	Typ.	Max.	Unit	Remark
Processor performance	GPU	--	80	--	MHz	
	CPU	--	100	--	MHz	
	L1 Cache (8nS Register)		512		Bytes	
	L2 Cache (10nS SRAM)		8		KB	

Item	Min.	Typ.	Max.	Unit	Remark

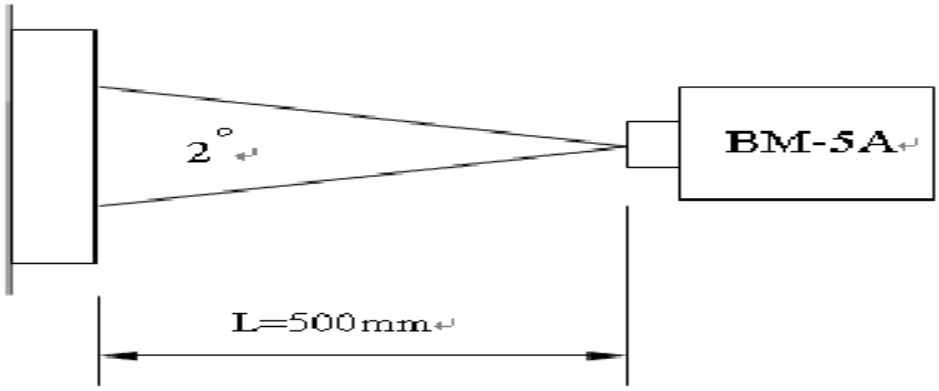
Storage	UART Buffer		32		KB	
	Display Buffer		1024		KB	
	Flash Memory:		128		MB	
	Flash Memory: Font		32		MB	
	Flash Memory: Picture	64		94	MB	
	Flash Memory: User Data	0		32	MB	
Uart Speed	COM1 (3.3V CMOS) User interface	1200		115200	bps	
	COM2 (3.3V CMOS) Function expand	--	115200	--	bps	
LCD Interface	CLOCK_H	10	25	33	MHz	
	CLOCK_L	1	5	10	MHz	
	VOH (IOH=-100uA)	0.9VLCD	--	--	V	
	VOL(IOL=100uA)	--	--	0.2 VLCD	V	
	IO	-20		+20	mA	
	Resolution		640*480	800*600	Dots	
	Color mode	16 bit, R5G6B5				
	ESD(HBM)		2		KV	

5 Optical Specification

Item		Symbol	Condition	Min.	Typ.	Max.	Unit	Note
Contrast ratio		CR	Point - 5 Θ=Φ=0°	200	250	--	--	(1)(2)(3)
Luminance		Lw		350	500	-	cd/m²	(1)(3)
Luminance Uniformity		ΔL		70	75	-	%	(1)(3)
Response Time (White – Black)		$T_r + T_f$		--	50	--	ms	(1)(3)(5)
Viewing Angle	Vertical	Θ	CR≧10 Point – 5	80	100	-	Deg.	(1)(2)(4)
	Horizontal	Φ		120	140	-		
Color chromaticity	Red	Rx	Point - 5 Θ=Φ=0°	0.566	0.616	0.666	--	(1)(3)
		Ry		0.302	0.352	0.402		
	Green	Gx		0.308	0.358	0.408		
		Gy		0.518	0.568	0.618		
	Blue	Bx		0.096	0.146	0.196		
		By		0.086	0.136	0.186		
	White	Wx		0.296	0.346	0.396		
		Wy		0.328	0.378	0.428		
Life time			25℃		30,000		Hr	(6)

NOTE :

(1) Measure conditions : 25℃±2℃ , 60±10%RH under 10Lux , in the dark room by BM-7TOPCON) ,viewing 2° , VCC=3.3V , VDD=3.3V



(2) Definition of Contrast Ratio :

Contrast Ratio (CR) = (White) Luminance of ON ÷ (Black) Luminance of OFF

(3) Definition of Luminance :

Definition of Luminance Uniformity

Measure white luminance on the point 5 as figure9-1

Measure white luminance on the point 1 ~ 9 as figure9-1

ΔL = [L(MIN) / L(MAX)] X 100%

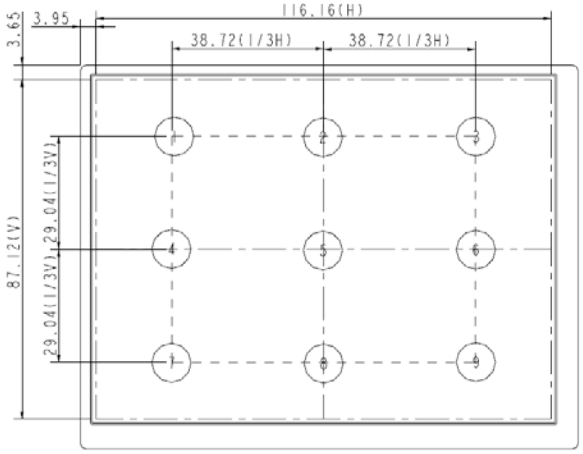


Fig9-1 Measuring point

(4) Definition of Viewing Angle(Θ, Φ), refer to Fig9-2 as below :

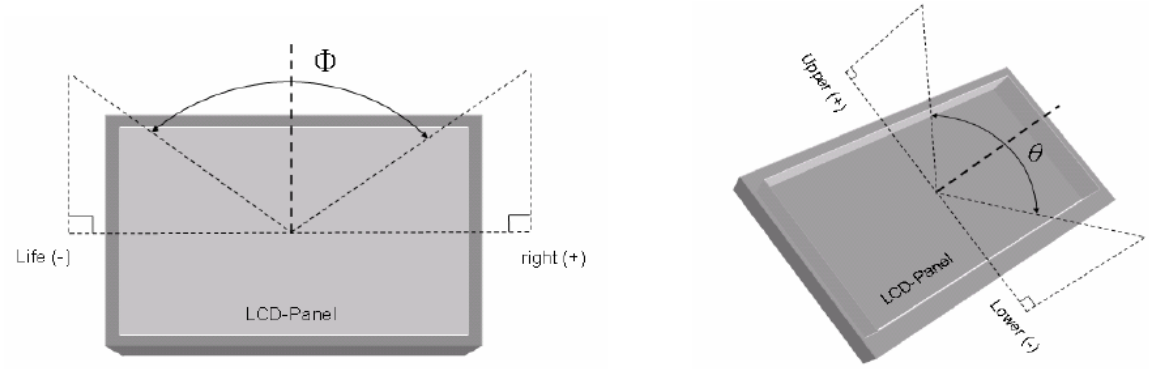


Fig9-2 Definition of Viewing Angle

(5) Definition of Response Time.(White – Black)

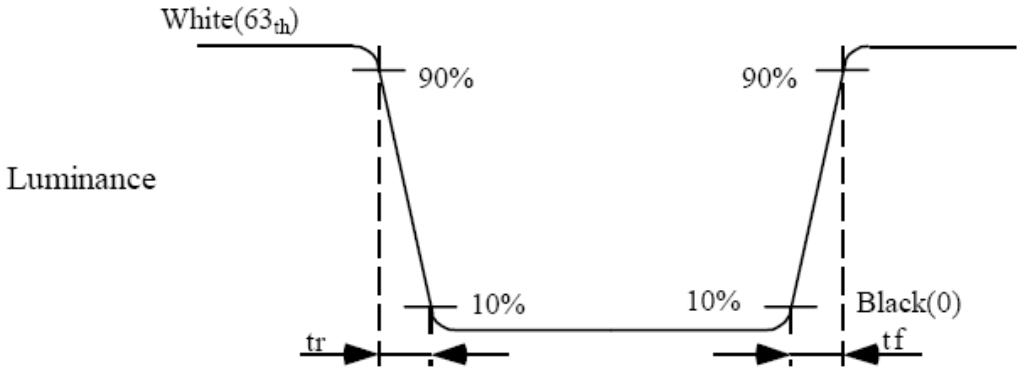
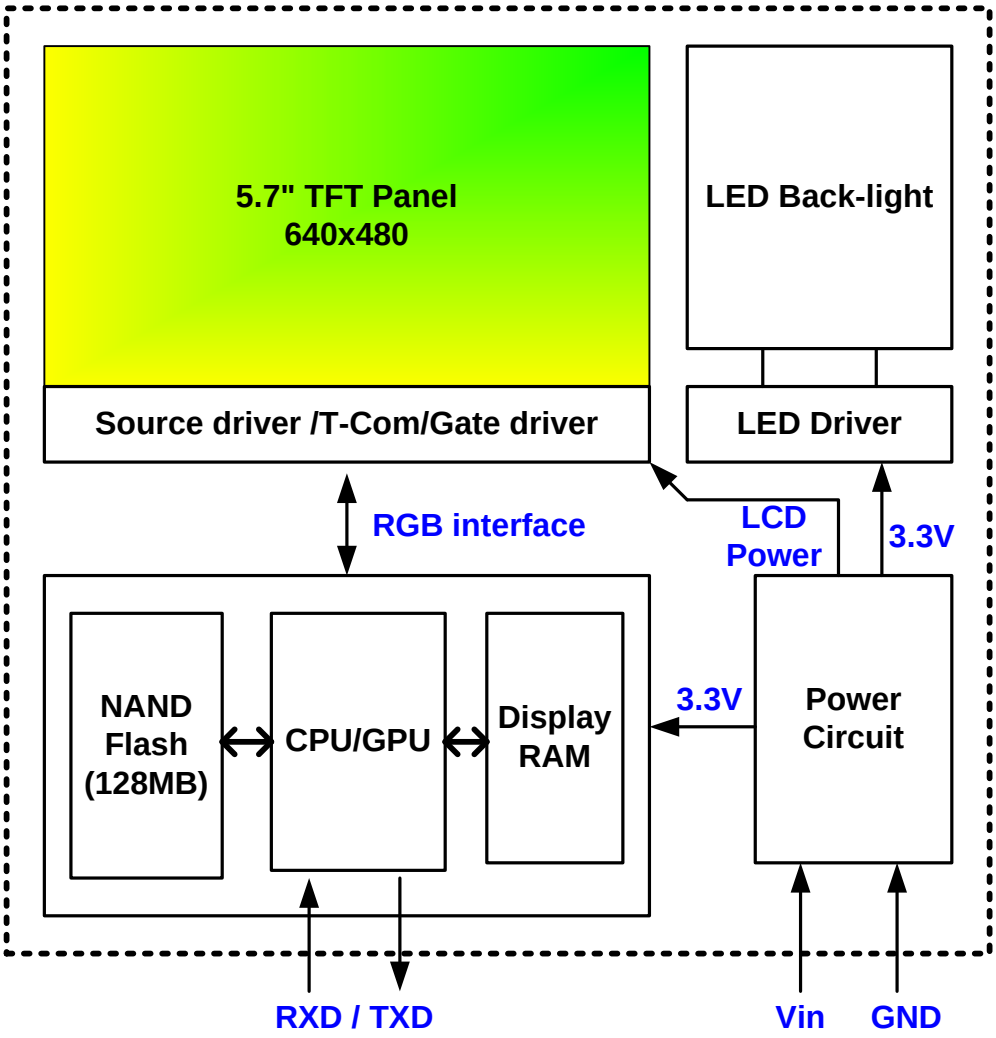


Fig9-3 Definition of Response Time(White-Black)

(6) By estimation. Luminance decreases to 50% of the min. value.

6 Block Diagram



6.1 Jumper Setting :

Operation mode	Jumper Setting			Remark
	RA1 or RA5	RA2or RA6	RA3or RA4	
UART Mode	Short	OPEN	OPEN	1200~115200 can be selected Touch Panel Workable
RS232 Mode	OPEN	Short	OPEN	1200~115200 can be selected Touch Panel Workable
USB Mode	OPEN	OPEN	Short	BPS: 921600 is fixed Touch Panel can not work

Note : The on board USB connector is USB B type

USB driver CP2102 : CP2102 USB to UART driver.rar

7 Interface specifications

8 Pins Pitch 2.5 connector				
PIN	Name	I/O	Description	Remark
1	Vin	P	Power input pin for the module (4.6~26V)	
2	Vin	P	Power input pin for the module (4.6~26V)	
3	/BUSY	O	Hi : Idle. The user can send the data to module. Lo: Busy. The data will be lost.	Note1
4	TXD	O	User UART serial signal output	
5	RXD	I	User UART serial signal input	
7	GND	P	Power and signal Ground	
8	GND	P	Power and signal Ground	

Note 1: The /BUSY=1 indicates that the internal GPU/CPU can accept at least a "**Command Frame**". The /BUSY=0 indicates that the interface board can't accept any "**Command Frame**". The user can ignore the /BUSY signal unless the user use the "**Command Frame End**" and send lots " **Command Frame** " over the UART buffer.

8 Software Command SET

8.1 Basic Agreement

8.1.1 UART default Parameters

The default parameters of UART are:

Baudrate: ..... 115200 bps (1200~115200 can be selected)

No of Stop Bits: 1

Parity: ..... Off

Handshake: ..... None

8.1.2 Serial port data frame structure (Hex mode)

The **"**Serial port data frame structure**"** includes **Start Character**, **Command**, **Data Pack** and **End Characters**.

The command Frame stat with **"Start Character 0xAA".** And the follow byte is **Command Byte, Data Pack.** The "**End Byte"** is 0XCC 0x33 0xC3 0X3C. The user can add "Delay time" instead of the End Byte. But if the "End Byte" has been send, the "Delay time " lose efficacy.

Structure	**Start Character**	**Command**	**Data Pack**	**End Characters**
Length(Byte)	**1**	**1**	**0-249**	**4**
	0XAA	One Byte (see command Set)	Relative Data (max 249 bytes)	0xCC 0x33 0xC3 0x3C Or add the Delay time

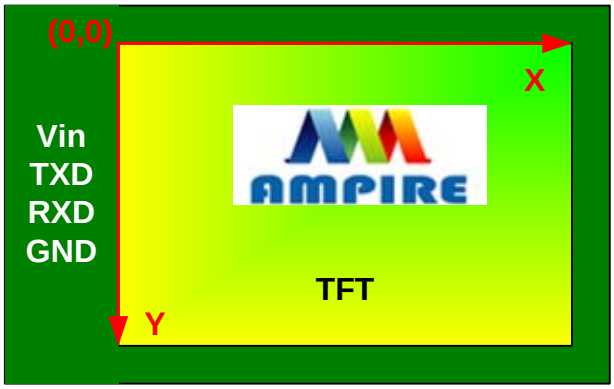
Baudrate_Set	Baudrate (bps)	Relative Delay Time
0x00	1200	12.5mS
0x01	2400	6.25mS
0X02	4800	3mS
0X03	9600	3mS
0X04	19200	2mS
0X05	38400	2mS
0X06	57600	1mS
0X07	115200	1mS

8.2 Command SET (Hex mode)

All the command description is hexadecimal format (HEX).

The X,Y coordinate express two bytes. High byte(MSB) + Low byte(LSB).

EX : X=10 (0x000A), The user must send the 0x00 0x0A.



8.2.1 Configuration Command

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0xE0**	**0x55 0xAA 0x5A 0xA5 +TFT_ID+ Baudrate_Set +0x00**	**0xCC 0x33 0xC3 0x3C** Or add the Delay time

Description: Set the TFT _ ID, UART Baudrate,

Class: Multi Byte Command

Code: E0hex, **224**dec

TFT _ ID:

TFT_ID	Resolution	Mode	Picture	Remark
0x04	800x480	SYNC	128	AM800480E
0x08	480x272	SYNC	255	AM480272C/D
0x0A / 0x01	640x480	SYNC	153	AM640480G2
0x0B	320x240	SYNC	153	AM320240N/N1

Baudrate_Set:

Baudrate_Set	Baudrate (bps)	Relative Delay Time
0x00	1200	12.5mS
0x01	2400	6.25mS
0X02	4800	3mS
0X03	9600	3mS
0X04	19200	2mS
0X05	38400	2mS
0X06	57600	1mS
0X07	115200	1mS

Example : AM320240N/N1 , Baudrate 115200bps.

RXD0 Input : **AA E0 55 AA 5A A5 0B 07 00**

TXD0 Output : **AA E0 0B 07 00 CC 33 C3 3C**

Note: The default Baudrate is 115200 N, 8 , 1 between power on 0~200mS , after 200mS the baudrate is set to user setting.

8.2.2 Hand Shake Command

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x00**	**None**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Check the interface board communication

Class: Single Byte Command

Code: 00hex, 00dec

Example : The Output information depend on configuration.

RXD0 Input : **AA 00**

TXD0 Output: **AA 00 " OK_V3.10" 0B 07 00 CC 33 C3 3C**

Note: The power on reset time is 0.5~2 seconds.

8.2.3 Display Mode Setting

8.2.3.1 SET_COLOR

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x40	Front Color MSB Front Color LSB BK Color MSB BK Color MSB	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Set Front Color and Back Color. The Color data is 16 bits (0x0000~0xFFFF). The default Front color is 0xFFFF(White) and the Background color is 0x001F (Blue).

Class: Multi Byte Command

Code: 40hex, 64dec

7	6	5	4	3	2	1	0
SET_COLOR 0x40							
FC R4	FC R3	FC R2	FC R1	FC R0	FC G5	FC G4	FC G3
FC G2	FC G1	FC G0	FC B4	FC B3	FC B2	FC B1	FC B0
BKC R4	BKC R3	BKC R2	BKC R1	BKC R0	BKC G5	BKC G4	BKC G3
BKC G2	BKC G1	BKC G0	BKC B4	BKC B3	BKC B2	BKC B1	BKC B0

Example : Set the Front color =0xFFFF(White) and Background color is 0x001F (Blue).

RXD0 Input : **AA 40 FF FF 00 1F**

TXD0 Output: **None**

8.2.3.2 SET_FONT_PITCH

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x41**	**PITCH_X** **PITCH_Y**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Set the character FONT display pitch. PITCH_X and PITCH_Y can't set over than 0x7F. The default PITCH_X= PITCH_Y=0

Class: Multi Byte Command

Code: 41hex, 65dec

7	6	5	4	3	2	1	0
SET_FONT_PITCH 0x41							
PITCH_X							
PITCH_Y							

Example : Set PCITH_X=1,PITCHY=2.

RXD0 Input : **AA 41 01 02**

TXD0 Output: **None**

8.2.3.3 SET_BACKGROUND_COLOR By X,Y

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x42**	**X MSB** **X LSB** **Y MSB** **Y MSB**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Get the (X,Y) display color data and set to the Background color.

Class: Multi Byte Command

Code: 42hex, 66dec

7	6	5	4	3	2	1	0
SET_BACKGROUND_COLOR By X,Y 0x42							
X_MSB							
X_LSB							
Y_MSB							
Y_LSB							

Example : Read the (X,Y)=(128,64) color data and set to the background color.

RXD0 Input : **AA 42 00 80 00 40**

TXD0 Output: **None**

8.2.3.4 SET_FRONT_COLOR By X,Y

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x43**	**X MSB** **X LSB** **Y MSB** **Y MSB**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Get the (X,Y) display color data and set to the Front color.

Class: Multi Byte Command

Code: 43hex, 67dec

7	6	5	4	3	2	1	0
SET_FRONT_COLOR By X,Y 0x42							
X_MSB							
X_LSB							
Y_MSB							
Y_LSB							

Example : Read the (X,Y)=(128,64) color data and set to the Front color.

RXD0 Input : **AA 43 00 80 00 40**

TXD0 Output: **None**

8.2.3.5 SET_CURSOR_MODE

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x44**	**Mode** **X MSB** **X LSB** **Y MSB** **Y LSB** **Cursor Wide** **Cursor High**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Set Cursor OFF (**Mode =0x00**) . Cursor ON (**Mode=0x01**). The (X,Y) is the left-up position of the cursor. Cursor Wide (0x01~0x1F) Cursor High (0x01~0x1F)

Class: Multi Byte Command

Code: 44hex, 68dec

7	6	5	4	3	2	1	0
SET_CURSOR_MODE 0x44							
Mode							
X MSB							
X MSB							
Y MSB							
Y MSB							
Cursor Wide							
Cursor High							

Example : Set Cursor ON, (X,Y)=(128,64) , Cursor Width=16 , Cursor High=4

RXD0 Input : **AA 44 01 00 80 00 40 10 04**

TXD0 Output: **None**

8.2.4 Text Display Command

8.2.4.1 PRINT_STRING_8x8 (Standard Font)

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x53**	**X MSB** **X LSB** **Y MSB** **Y LSB** **“String”**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Prints String. starting at (X,Y) Position

Class: Multi Byte Command

Code: 53hex, 83dec

7	6	5	4	3	2	1	0
PRINT_STRING_8x8 0x53							
X MSB							
X MSB							
Y MSB							
Y MSB							
ASCII							
ASCII							
~							
ASCII							
ASCII							

Example : Print 8x8 string “ LCD “ on (128,64).

RXD0 Input : **AA 53 00 80 00 40 4C 43 44**

```
PRINT_STRING_8x8 53 hex
XMSB 00 hex
XLSB 80 hex
YMSB 00 hex
YLSB 40 hex
'L' 4C hex
'C' 43 hex
'D' 44 hex
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y , SET_FRONT_COLOR By X,Y ,SET_FONT_PITCH

8.2.4.2 PRINT_STRING_16x16 (Standard Font)

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x54**	**X MSB** **X LSB** **Y MSB** **Y LSB** **“String”**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Prints String. start at (X,Y) Position

Class: Multi Byte Command

Code: 54hex, 84dec

7	6	5	4	3	2	1	0
PRINT_STRING _16x16 0x54							
X MSB							
X MSB							
Y MSB							
Y MSB							
ASCII							
ASCII							
~							
ASCII							
ASCII							

Example : Print 16x16 string “ LCD “ on (128,64).

RXD0 Input : **AA 54 00 80 00 40 4C 43 44**

```
PRINT_STRING_16x16 54 hex
XMSB 00 hex
XLSB 80 hex
YMSB 00 hex
YLSB 40 hex
'L' 4C hex
'C' 43 hex
'D' 44 hex
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y , SET_FRONT_COLOR By X,Y ,SET_FONT_PITCH

8.2.4.3 PRINT_STRING_32x32 (Standard Font)

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x55**	**X MSB** **X LSB** **Y MSB** **Y LSB** **“String”**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Prints String. starting at (X,Y) Position (Support GBK3212 FONT code)

Class: Multi Byte Command

Code: 55hex, 85dec

7	6	5	4	3	2	1	0
PRINT_STRING _32x32 0x55							
X MSB							
X MSB							
Y MSB							
Y MSB							
ASCII							
ASCII							
~							
ASCII							
ASCII							

Example : Print 32x32 string “ LCD “ on (128,64).

RXD0 Input : **AA 55 00 80 00 40 4C 43 44**

```
PRINT_STRING_32x32 55 hex
XMSB 00 hex
XLSB 80 hex
YMSB 00 hex
YLSB 40 hex
'L' 4C hex
'C' 43 hex
'D' 44 hex
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y , SET_FRONT_COLOR By X,Y ,SET_FONT_PITCH

8.2.4.4 PRINT_STRING_12x12(Standard Font)

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x6E	X MSB X LSB Y MSB Y LSB “String”	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Prints String. starting at (X,Y) Position

Class: Multi Byte Command

Code: 6Ehex, 110dec

7	6	5	4	3	2	1	0
PRINT_STRING _12x12 0x6E							
X MSB							
X MSB							
Y MSB							
Y MSB							
ASCII							
ASCII							
~							
ASCII							
ASCII							

Example : Print 12x12 string “ LCD “ on (128,64).

RXD0 Input : **AA 6E 00 80 00 40 4C 43 44**

```
PRINT_STRING_12x12 6E hex
XMSB 00 hex
XLSB 80 hex
YMSB 00 hex
YLSB 40 hex
'L' 4C hex
'C' 43 hex
'D' 44 hex
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y , SET_FRONT_COLOR By X,Y ,SET_FONT_PITCH

8.2.4.5 PRINT_STRING_24x24 (Standard Font)

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x6F**	**X MSB** **X LSB** **Y MSB** **Y LSB** **“String”**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Prints String. starting at (X,Y) Position

Class: Multi Byte Command

Code: 6Fhex, 111dec

7	6	5	4	3	2	1	0
PRINT_STRING _24x24 0x6F							
X MSB							
X MSB							
Y MSB							
Y MSB							
ASCII							
ASCII							
~							
ASCII							
ASCII							

Example : Print 24x24 string “ LCD “ on (128,64).

RXD0 Input : **AA 6F 00 80 00 40 4C 43 44**

```
PRINT_STRING_24x24 6F hex
XMSB 00 hex
XLSB 80 hex
YMSB 00 hex
YLSB 40 hex
'L' 4C hex
'C' 43 hex
'D' 44 hex
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y , SET_FRONT_COLOR By X,Y ,SET_FONT_PITCH

8.2.4.6 PRINT_STRING (Custom Font Library)

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x98**	**X MSB** **X LSB** **Y MSB** **Y LSB** **Lib_ID** **C_mode** **C_dots** **Color** **Bcolor** **"String"**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Prints String custom font starting at (X,Y) Position

Class: Multi Byte Command

Code: 98hex, 152dec

Lib_ID：0x00~0x3B , Font library ID. The memory size of the font library is 32MB. The memory is arranged as 60 areas.

Lib_ID	Size	Description	Default
0x00~0x1F	128KB	There are 32 areas for small font library which size is 128KB. The user can download different style for ASCII code or small ICON	0x00=ASCII standard font library, please don't modify it. 0x01~0x1F: Free
0x20~0x3B	1MB	There are 28 areas for large font library which size is 1MB. ● Single area can be set as 12x12 or 16x16 (GBK encode) ● The area can be combined for supper large font library.	0x20=12x12 GBK 0x21=16x16 GBK 0x22=24x24 GB2312 0x23=32x32 GB2312 0x24~0x3B : Free

C_Mode：The mode of the character font:

Bit No	Bit7~Bit4	Bit3~Bit0
Definition	Display mode	Lib_Type(0x00~0x0F)
Description	Bit7=1: Front color is displayed Bit7=0: Front color is not displayed Bit6=1: Back ground color is displayed Bit6=0: Back ground color is not displayed Bit5 , Bit4 : don't care	Lib_Type define the encoding systems of the font : 0= 8bit 1=GB2312 2=GBK or HANGUL 3=BIG5 4=SJIS 5=UNICODE 6-F: undefined

C_dots：The display font size. (See the detail information: Custom Font Definition)

C_dots	C_Mode (Bit3~Bit0)		
	0 or 5	1 , 2 , 3, 4	
		ASCII	Chinese/Korea/Japan
0x00	8x8	6x12	12x12
0x01	6x12	8x16	16x16
0x02	8x16	12x24	24x24
0x03	12x24	16x32	32x32
0x04	16x32	20x40	40x40
0x05	20x40	24x48	48x48
0x06	24x48	28x56	56x56
0x07	28x58	32x64	64x64
0x08	32x64		
0x09	12x12		
0x0A	16x16		
0x0B	24x24		
0x0C	32x32		
0x0D	40x40		
0x0E	48x48		
0x0F	56x56		
0x10	64x64		
0x11	40x80		
0x12	48x96		
0x13	56x112		
0x14	64x128		
0x15	80x80		
0x16	96x96		
0x17	112x112		
0x18	128x128		

Color ：The color of the font.

Bcolor：The background color of the font.

String：The encoding system is defined by C_Mode (Bit3~Bit0). If C_Mode=0,1,2,3and 4, the code identify as “ ASCII “ and showed by Lib_ID= 0.

7	6	5	4	3	2	1	0
PRINT_STRING _Custom Font							
X MSB							
X MSB							
Y MSB							
Y MSB							
Lib_ID							
C_mode							
C_dots							
Color							
Bcolor							
Font code							
Font code							
~							
Font code							

Example : Print string “ AMPIRE晶采光電 “ on (128,64).

RXD0 Input :

AA 98 00 80 00 40 24 C3 02 FF FF 00 00 41 4D 50 49 52 45 B4 B9 AA F6 A5 FA B9 71 CC 33 C3 3C

AA: Start Byte

98: command Byte

00 80 : X position

00 40 : Y position

24 : Lib_ID (The user must to download the BIG5_24x24 to 0x24)

C3 : C_mode Front color is displayed , Back ground color is displayed , BIG5

02 : C_dots , ASCII size :12x24 , Chinese : 24x24

FF FF : Color is white

00 00 : Bcolor , background color is black.

41 4D 50 49 52 45: 'AMPIRE'

B4 B9 AA F6 A5 FA B9 71: '晶采光電'

CC 33 C3 3C: End Byte

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y , SET_FRONT_COLOR By X,Y ,SET_FONT_PITCH

8.2.5 Put pixel Command

8.2.5.1 Put Pixel by Background Color

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x50	X_0 MSB X_0 LSB Y_0 MSB Y_0 LSB X_1 MSB X_1 LSB Y_1 MSB Y_1 LSB ~~ X_k MSB X_k LSB Y_k MSB Y_k LSB	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Put the background color to the point (X_0,Y_0) , (X_1,Y_1) ,…(X_k,Y_k)

Class: Multi Byte Command

Code: 50hex, 80dec

7	6	5	4	3	2	1	0
Put Pixel by Background color							
X_0 MSB							
X_0 MSB							
Y_0 MSB							
Y_0 MSB							
X_1 MSB							
X_1 MSB							
Y_1 MSB							
Y_1 MSB							
~~							
X_k MSB							
X_k MSB							
Y_k MSB							
Y_k MSB							

Example : Put pixel by background color on (0,0) (1,1) (2,2)

RXD0 Input : **AA 50 00 00 00 00 00 01 00 01 00 02 00 02 CC 33 C3 3C**

```
AA: Start Byte
50: command Byte
00 00 00 00 : (X,Y)=(0,0)
00 01 00 01 : (X,Y)=(1,1)
00 02 00 02 : (X,Y)=(2,2)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y

8.2.5.2 Put Pixel by Front Color

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x51**	**X_0 MSB** **X_0 LSB** **Y_0 MSB** **Y_0 LSB** **X_1 MSB** **X_1 LSB** **Y_1 MSB** **Y_1 LSB** **~~** **X_k MSB** **X_k LSB** **Y_k MSB** **Y_k LSB**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Put the front color to the point (X_0,Y_0) , (X_1,Y_1) ,…(X_k,Y_k)

Class: Multi Byte Command

Code: 51hex, 81dec

7	6	5	4	3	2	1	0
Put Pixel by Front color							
X_0 MSB							
X_0 MSB							
Y_0 MSB							
Y_0 MSB							
X_1 MSB							
X_1 MSB							
Y_1 MSB							
Y_1 MSB							
~~							
X_k MSB							
X_k MSB							
Y_k MSB							
Y_k MSB							

Example : Put pixel by front color on (0,0) (1,1) (2,2)

RXD0 Input : **AA 51 00 00 00 00 00 01 00 01 00 02 00 02 CC 33 C3 3C**

```
AA: Start Byte
50: command Byte
00 00 00 00 : (X,Y)=(0,0)
00 01 00 01 : (X,Y)=(1,1)
00 02 00 02 : (X,Y)=(2,2)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_FRONT_COLOR By X,Y

8.2.5.3 Put Pixel and Clear vertical line by specify color

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x74**	**X MSB** **X LSB** **Y_S MSB** **Y_S LSB** **Y_E MSB** **Y_E LSB** **Clear Color MSB** **Clear Color LSB** **Y_1 MSB** **Y_1 LSB** **Color(X,Y_1) MSB** **Color(X,Y_1) LSB** **Y_2 MSB** **Y_2 LSB** **Color(X,Y_2) MSB** **Color(X,Y_2) LSB** **~~** **Y_k MSB** **Y_k LSB** **Color(X,Y_k) MSB** **Color(X,Y_k) LSB**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: This command is for motion curve application. This command include two actions.

- Draw a vertical line on (**X ,Y_S**) (**X ,Y_E**) by **Clear Color**
- Put **(X,Y1)** by **Color (X,Y1)** , Put **(X,Y2)** by **Color (X,Y2)** ……and Put **(X,Yk)** by **Color (X,Yk).**

Class: Multi Byte Command

Code: 74hex, 116 dec

Example :

Clear line (5,0) (5,255) by Black 0x0000

Put (5, 0) by Red 0xF800

Put (5, 16) by Green 0x07E0

Put (5, 32) by Blue 0x001F

Put (5, 48) by White 0xFFFF

RXD0 Input : **AA 74 00 05 00 00 00 FF 00 00 00 00 F8 00 00 10 07 E0 00 20 00 1F 00 30 FF FF CC 33 C3 3C**

`AA: Start Byte`

```
74: command Byte
00 05 : X = 5
00 00 : YS= 0
00 FF : YE= 255
00 00 : Clear Color = Black 0x0000
00 00 : Y1 = 0
F8 00 : (X,Y1) color = Red 0xF800
00 10 : Y2 = 16
07 E0 : (X,Y2) color = Green 0x07E0
00 20 : Y3 = 32
00 1F : (X,Y3) color = Blue 0x001F
00 30 : Y4 = 48
00 1F : (X,Y4) color = White 0xFFFF
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

8.2.6Draw Line Command

8.2.6.1 Draw Front Color Lines by the specify points

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x56	X_0 MSB X_0 LSB Y_0 MSB Y_0 LSB X_1 MSB X_1 LSB Y_1 MSB Y_1 LSB ~~ X_k MSB X_k LSB Y_k MSB Y_k LSB	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Draw the line (X_0,Y_0) (X_1,Y_1) ….(Xk,Yk)

Class: Multi Byte Command

Code: 56hex, 86dec

Example : To draw a triangle by front color . (0,0) (128,64) (128,0) (0,0)

RXD0 Input : **AA 56 00 00 00 00 00 80 00 40 00 80 00 00 00 00 00 00 CC 33 C3 3C**

```
AA: Start Byte
56: command Byte
00 00 00 00 : (X,Y)=(0,0)
00 80 00 40 : (X0,Y0)=(128,64)
00 80 00 00 : (X1,Y1)=(128,0)
00 00 00 00 : (X2,Y2)=(0,0)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_FRONT_COLOR By X,Y

8.2.6.2 Draw Background Color Lines by the specify points

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x5D**	**X_0 MSB** **X_0 LSB** **Y_0 MSB** **Y_0 LSB** **X_1 MSB** **X_1 LSB** **Y_1 MSB** **Y_1 LSB** **~~** **X_k MSB** **X_k LSB** **Y_k MSB** **Y_k LSB**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Draw the line (X_0,Y_0) (X_1,Y_1) ….(Xk,Yk)

Class: Multi Byte Command

Code: 56hex, 86dec

Example : To draw a triangle by background color. (0,0) (128,64) (128,0) (0,0)

RXD0 Input : **AA 5D 00 00 00 00 00 80 00 40 00 80 00 00 00 00 00 00 CC 33 C3 3C**

```
AA: Start Byte
56: command Byte
00 00 00 00 : (X,Y)=(0,0)
00 80 00 40 : (X0,Y0)=(128,64)
00 80 00 00 : (X1,Y1)=(128,0)
00 00 00 00 : (X2,Y2)=(0,0)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y

8.2.7 Draw Circle Command

8.2.7.1 Draw a Circle by the specified the radius and center points.

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x57**	**TYPE** **X_0 MSB** **X_0 LSB** **Y_0 MSB** **Y_0 LSB** **R_0** **~~** **TYPE** **X_k MSB** **X_k LSB** **Y_k MSB** **Y_k LSB** **R_k**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Draw the Circle by the specify center (X_0,Y_0) , Radius R0 .

Class: Multi Byte Command

Code: 56hex, 86dec

Type : Circle display mode .

Type=0 Reverse the color of the Circle

Type=1 Display the front color to the Circle

Type=2 Reverse the color of the full circle area

Type=3 Display the front color to the full circle area

Example : Reverse the color of the circle which radius = 100 and center (200,250)

RXD0 Input : **AA 57 02 00 C8 00 FA 64 CC 33 C3 3C**

```
AA: Start Byte
56: command Byte
00 00 00 00 : (X,Y)=(0,0)
00 80 00 40 : (X0,Y0)=(128,64)
00 80 00 00 : (X1,Y1)=(128,0)
00 00 00 00 : (X2,Y2)=(0,0)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_FRONT_COLOR By X,Y

8.2.8Draw Rectangle Command

8.2.8.1 Draw a front color Rectangle by the specified the left-up and right-down points.

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x59**	**X_{S0} MSB** **X_{S0} LSB** **Y_{S0} MSB** **Y_{S0} LSB** **X_{E0} MSB** **X_{E0} LSB** **Y_{E0} MSB** **Y_{E0} LSB** **~~** **X_{Sk} MSB** **X_{Sk} LSB** **Y_{Sk} MSB** **Y_{Sk} LSB** **X_{Ek} MSB** **X_{Ek} LSB** **Y_{Ek} MSB** **Y_{Ek} LSB**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Draw the front color Rectangle by the specify the left-up point (X_{S0},Y_{S0}) , Right-Down points(X_{E0},Y_{E0}) .

Class: Multi Byte Command

Code: 59hex, 89dec

Example : Draw the two Rectangle. One is the left-up point (64,64) and Right-Down points(128,128) . The other one is the left-up point (10,10) and Right-Down points (20,20) .

RXD0 Input : **AA 59 00 40 00 40 00 80 00 80 00 0A 00 0A 00 14 00 14 CC 33 C3 3C**

```
AA: Start Byte
59: command Byte
00 40 00 40 : (XS0,YS0)=(64,64)
00 80 00 80 : (XE0,YE0)=(128,64)
00 0A 00 0A : (XS1,YS1)=(10,10)
00 14 00 14 : (XS1,YS1)=(20,20)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_FRONT_COLOR By X,Y

8.2.8.2 Draw a background color Rectangle by the specified the left-up and right-down points.

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x69**	**X_{S0} MSB** **X_{S0} LSB** **Y_{S0} MSB** **Y_{S0} LSB** **X_{E0} MSB** **X_{E0} LSB** **Y_{E0} MSB** **Y_{E0} LSB** **~~** **X_{Sk} MSB** **X_{Sk} LSB** **Y_{Sk} MSB** **Y_{Sk} LSB** **X_{Ek} MSB** **X_{Ek} LSB** **Y_{Ek} MSB** **Y_{Ek} LSB**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Draw the background color Rectangle by the specify the left-up point (X_{S0},Y_{S0}) , Right-Down points(X_{E0},Y_{E0}) .

Class: Multi Byte Command

Code: 69hex, 105dec

Example : Draw the two Rectangle. One is the left-up point (64,64) and Right-Down points(128,128) . The other one is the left-up point (10,10) and Right-Down points (20,20) .

RXD0 Input : **AA 69 00 40 00 40 00 80 00 80 00 0A 00 0A 00 14 00 14 CC 33 C3 3C**

```
AA: Start Byte
69: command Byte
00 40 00 40 : (XS0,YS0)=(64,64)
00 80 00 80 : (XE0,YE0)=(128,64)
00 0A 00 0A : (XS1,YS1)=(10,10)
00 14 00 14 : (XS1,YS1)=(20,20)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y

8.2.9 Full Area Command

8.2.9.1 Clear Screen

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x52	None	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Clear the LCD with the background color.

Class: None Byte Command

Code: 52hex, 82dec

Example : Clear the LCD with the background color.

RXD0 Input : **AA 52 CC 33 C3 3C**

```
AA: Start Byte
52: command Byte
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y

8.2.9.2 Full area with Background color

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x5A**	**X_{S0} MSB** **X_{S0} LSB** **Y_{S0} MSB** **Y_{S0} LSB** **X_{E0} MSB** **X_{E0} LSB** **Y_{E0} MSB** **Y_{E0} LSB** **~~** **X_{Sk} MSB** **X_{Sk} LSB** **Y_{Sk} MSB** **Y_{Sk} LSB** **X_{Ek} MSB** **X_{Ek} LSB** **Y_{Ek} MSB** **Y_{Ek} LSB**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Full the Rectangle with background color by the specify the left-up point (X_{S0},Y_{S0}) , Right-Down points(X_{E0},Y_{E0}) .

Class: Multi Byte Command

Code: 5Ahex, 90dec

Example : Full two Rectangle with background color . One is left-up point (0,0) and Right-Down points (64,64). The other one is left-up point (160,128) and Right-Down points (240,160).

RXD0 Input : **AA 5A 00 00 00 00 00 40 00 40 00 A0 00 80 00 F0 00 A0 CC 33 C3 3C**

```
AA: Start Byte
5A: command Byte
00 00 00 00 : left-up point (0,0)
00 40 00 40 : Right-down point (64,64)
00 A0 00 80 : left-up point (160,128)
00 F0 00 A0 : Right-down point (240,160)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y

8.2.9.3 Full area with Front color

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x5B**	**X_{S0} MSB** **X_{S0} LSB** **Y_{S0} MSB** **Y_{S0} LSB** **X_{E0} MSB** **X_{E0} LSB** **Y_{E0} MSB** **Y_{E0} LSB** **~~** **X_{Sk} MSB** **X_{Sk} LSB** **Y_{Sk} MSB** **Y_{Sk} LSB** **X_{Ek} MSB** **X_{Ek} LSB** **Y_{Ek} MSB** **Y_{Ek} LSB**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Full the Rectangle with front color by the specify the left-up point (X_{S0},Y_{S0}) , Right-Down points(X_{E0},Y_{E0}) .

Class: Multi Byte Command

Code: 5Bhex, 91dec

Example : Full two Rectangle with front color . One is left-up point (0,0) and Right-Down points (64,64). The other one is left-up point (160,128) and Right-Down points (240,160).

RXD0 Input : **AA 5B 00 00 00 00 00 40 00 40 00 A0 00 80 00 F0 00 A0 CC 33 C3 3C**

```
AA: Start Byte
5B: command Byte
00 00 00 00 : left-up point (0,0)
00 40 00 40 : Right-down point (64,64)
00 A0 00 80 : left-up point (160,128)
00 F0 00 A0 : Right-down point (240,160)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_FRONT_COLOR By X,Y

8.2.9.4 Reverse area color

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x5C**	**X_{S0} MSB** **X_{S0} LSB** **Y_{S0} MSB** **Y_{S0} LSB** **X_{E0} MSB** **X_{E0} LSB** **Y_{E0} MSB** **Y_{E0} LSB** **~~** **X_{Sk} MSB** **X_{Sk} LSB** **Y_{Sk} MSB** **Y_{Sk} LSB** **X_{Ek} MSB** **X_{Ek} LSB** **Y_{Ek} MSB** **Y_{Ek} LSB**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Reverse the area color by the specify the left-up point (X_{S0},Y_{S0}) , Right-Down points(X_{E0},Y_{E0}) .

Class: Multi Byte Command

Code: 5Chex, 92dec

Example : Reverse two areas color. One is left-up point (0,0) and Right-Down points (64,64). The other one is left-up point (160,128) and Right-Down points (240,160).

RXD0 Input : **AA 5C 00 00 00 00 00 40 00 40 00 A0 00 80 00 F0 00 A0 CC 33 C3 3C**

```
AA: Start Byte
5C: command Byte
00 00 00 00 : left-up point (0,0)
00 40 00 40 : Right-down point (64,64)
00 A0 00 80 : left-up point (160,128)
00 F0 00 A0 : Right-down point (240,160)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

8.2.10 Move Area Command

8.2.10.1 Move area data to left N pixel by the specified the left-up and right-down points.

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x60	X_{S0} MSB X_{S0} LSB Y_{S0} MSB Y_{S0} LSB X_{E0} MSB X_{E0} LSB Y_{E0} MSB Y_{E0} LSB N_0 (N <=16) ~~ X_{Sk} MSB X_{Sk} LSB Y_{Sk} MSB Y_{Sk} LSB X_{Ek} MSB X_{Ek} LSB Y_{Ek} MSB Y_{Ek} LSB N_k (N <=16)	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Move area data to left N pixel by the specify the left-up point (X_{S0},Y_{S0}) , Right-Down points(X_{E0},Y_{E0}) .

Class: Multi Byte Command

Code: 60hex, 96dec

Example : Move two areas data to left 16 pixels. One is the left-up point (64,64) and Right-Down points(128,128) . The other one is the left-up point (16,16) and Right-Down points (48,48) .

RXD0 Input :

AA 60 00 40 00 40 00 80 00 80 10 00 10 00 10 00 30 00 30 10 CC 33 C3 3C

```
AA: Start Byte
60: command Byte
00 40 00 40 : (XS0,YS0)=(64,64)
00 80 00 80 : (XE0,YE0)=(128,64)
00 10 00 10 : (XS1,YS1)=(16,16)
00 30 00 30 : (XS1,YS1)=(48,48)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

8.2.10.2 Move area data to right N pixel by the specified the left-up and right-down points.

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x61**	**X_{S0} MSB** **X_{S0} LSB** **Y_{S0} MSB** **Y_{S0} LSB** **X_{E0} MSB** **X_{E0} LSB** **Y_{E0} MSB** **Y_{E0} LSB** **N_0 (N <=16)** **~~** **X_{Sk} MSB** **X_{Sk} LSB** **Y_{Sk} MSB** **Y_{Sk} LSB** **X_{Ek} MSB** **X_{Ek} LSB** **Y_{Ek} MSB** **Y_{Ek} LSB** **N_k (N <=16)**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Move area data to right N pixel by the specify the left-up point (X_{S0},Y_{S0}) , Right-Down points(X_{E0},Y_{E0}) .

Class: Multi Byte Command

Code: 61hex, 97dec

Example : Move two areas data to right 16 pixels. One is the left-up point (64,64) and Right-Down points(128,128) . The other one is the left-up point (16,16) and Right-Down points (48,48) .

RXD0 Input :

AA 61 00 40 00 40 00 80 00 80 10 00 10 00 10 00 30 00 30 10 CC 33 C3 3C

```
AA: Start Byte
61: command Byte
00 40 00 40 : (XS0,YS0)=(64,64)
00 80 00 80 : (XE0,YE0)=(128,64)
00 10 00 10 : (XS1,YS1)=(16,16)
00 30 00 30 : (XS1,YS1)=(48,48)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

8.2.10.3 Move area data to left N pixel by the specified the left-up and right-down points. And full the right N pixel with background color

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x62	X_{S0} MSB X_{S0} LSB Y_{S0} MSB Y_{S0} LSB X_{E0} MSB X_{E0} LSB Y_{E0} MSB Y_{E0} LSB N_0 (N <=16) ~~ X_{Sk} MSB X_{Sk} LSB Y_{Sk} MSB Y_{Sk} LSB X_{Ek} MSB X_{Ek} LSB Y_{Ek} MSB Y_{Ek} LSB N_k (N <=16)	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Move area data to left N pixel by the specify the left-up point (X_{S0},Y_{S0}) , Right-Down points(X_{E0},Y_{E0}) . And full the right N pixel with background color.

Class: Multi Byte Command

Code: 62hex, 98dec

Example : Move two areas data to left 16 pixels. One is the left-up point (64,64) and Right-Down points(128,128) . The other one is the left-up point (16,16) and Right-Down points (48,48) .And full the right side 16 pixels with background color.

RXD0 Input :

AA 62 00 40 00 40 00 80 00 80 10 00 10 00 10 00 30 00 30 10 CC 33 C3 3C

```
AA: Start Byte
62: command Byte
00 40 00 40 : (XS0,YS0)=(64,64)
00 80 00 80 : (XE0,YE0)=(128,64)
00 10 00 10 : (XS1,YS1)=(16,16)
00 30 00 30 : (XS1,YS1)=(48,48)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y

8.2.10.4 Move area data to right N pixel by the specified the left-up and right-down points. And full the left N pixel with background color

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x63**	**X_{S0} MSB** **X_{S0} LSB** **Y_{S0} MSB** **Y_{S0} LSB** **X_{E0} MSB** **X_{E0} LSB** **Y_{E0} MSB** **Y_{E0} LSB** **N_0 (N <=16)** **~~** **X_{Sk} MSB** **X_{Sk} LSB** **Y_{Sk} MSB** **Y_{Sk} LSB** **X_{Ek} MSB** **X_{Ek} LSB** **Y_{Ek} MSB** **Y_{Ek} LSB** **N_k (N <=16)**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Move area data to right N pixel by the specify the left-up point (X_{S0},Y_{S0}) , Right-Down points(X_{E0},Y_{E0}) . And full the left N pixel with background color.

Class: Multi Byte Command

Code: 63hex, 99dec

Example : Move two areas data to right 16 pixels. One is the left-up point (64,64) and Right-Down points(128,128) . The other one is the left-up point (16,16) and Right-Down points (48,48) .And full the left side 16 pixels with background color.

RXD0 Input :

AA 63 00 40 00 40 00 80 00 80 10 00 10 00 10 00 30 00 30 10 CC 33 C3 3C

```
AA: Start Byte
63: command Byte
00 40 00 40 : (XS0,YS0)=(64,64)
00 80 00 80 : (XE0,YE0)=(128,64)
00 10 00 10 : (XS1,YS1)=(16,16)
00 30 00 30 : (XS1,YS1)=(48,48)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

See also: SET_COLOR , SET_BACKGROUND_COLOR By X,Y

8.2.11 Picture Command

8.2.11.1 Show the full picture by specify the PICNUM

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x70**	**PICNUM**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: The command will show the full picture which save in the picture flash memory by specify PICNUM.

Class: Single Byte Command

Code: 70hex, 112dec

Example : Show the picture which PICNUM=1.

RXD0 Input : **AA 70 01 CC 33 C3 3C**

```
AA: Start Byte
70: command Byte
01: PICNUM
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

8.2.11.2 Show the area of picture in position X,Y by specify the PICNUM , left-up and right-down points , X and Y.

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x71	PICNUM X_{S0} MSB X_{S0} LSB Y_{S0} MSB Y_{S0} LSB X_{E0} MSB X_{E0} LSB Y_{E0} MSB Y_{E0} LSB X MSB X LSB Y MSB Y LSB	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: The command will show the area of the picture which left-up point (X_{S0},Y_{S0}) , Right-Down points(X_{E0},Y_{E0}) and PICNU in position X,Y.

Class: Multi Byte Command

Code: 71hex, 113dec

Example : Show the area of the picture which PICNUM=2 , left-up point (100,100) , Right-Down points(200,200) in the position (0,0).

.

RXD0 Input : **AA 71 02 00 64 00 64 00 C8 00 C8 00 00 00 00 CC 33 C3 3C**

```
AA: Start Byte
71: command Byte
02: PICNUM
00 64 00 64: left-up point (100,100)
00 C8 00 C8: right-down point (200,200)
00 00 00 00: Show on position (0,0)
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

8.2.11.3 Save the screen data to flash memory.

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0xE2	PICNUM	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: The command will save all the screen data to flash memory. The picture address is PICNUM.

Class: Single Byte Command

Code: E2hex, 226dec

Example : Save the screen data to PICNUM=2.

RXD0 Input : **AA E2 02 CC 33 C3 3C**

```
AA: Start Byte
E2: command Byte
02: save picture address PICNUM
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

8.2.12 Display RAM Command

8.2.12.1 Write data to the display RAM.

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0x72**	**ADR Bit23~Bit16** **ADR Bit15~Bit8** **ADR Bit7~Bit0** **DATA0** **DATA1** **~** **DATAn (n<=248)**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: The user can write the display RAM directly. The address range is 0x00000~0x7FFFF. The width of the data is 16 bits.

Address = Resolution_X * Position_Y+ Position_X.

EX: If the Display Resolution is 640x480. The address of position (80,100) is 640*100+80=64080(dec) = 0x0FA50(hex)

Class: Multi Byte Command

Code: 72hex, 114dec

Example : Write the Red(0xf800) color to the position (80,100) , Green (0x07E0) color to the position (81,100)

RXD0 Input :

AA 72 00 FA 50 F8 00 07 E0 CC 33 C3 3C

```
AA: Start Byte
72: command Byte
00 FA 50 : Display RAM Address.
F8 00 : Red data
07 E0 : Green data
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

8.2.13 IO control Command

8.2.13.1 SET I/O1 to Low

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x5E	None	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: The user can set the I/O1 to Low by the command. When I/O1 connect to LED backlight driving circuit , the B/L will turn OFF.

Class: Single Byte Command

Code: 5Eex, 94dec

Example : Set I/O1=Low

RXD0 Input : **AA 5E CC 33 C3 3C**

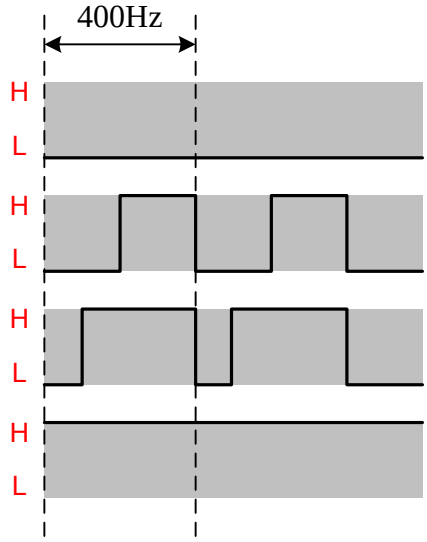
```
AA: Start Byte
5E: command Byte
CC 33 C3 3C: End Byte
```

TXD0 Output: **None**

8.2.13.2 Back-light Dimming control I/O1

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x5F	PWM_DUTY (0x00-0x40)	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: The LED Back-light dimming circuit connect to the I/O1. The PWM_DUTY(0x00~0x40) can be selected the brightness of the LED Back-light.



BL OFF
AA 5F 00 CC 33 C3 3C

BL 50% Brightness
AA 5F 20 CC 33 C3 3C

BL 75% Brightness
AA 5F 30 CC 33 C3 3C

BL 100% Brightness
AA 5F 40 CC 33 C3 3C

Class: Multi Byte Command

Code: 5Fex, 94dec

Example : Set the LED brightness = 75%

RXD0 Input : **AA 5F 30 CC 33 C3 3C**

AA: Start Byte

5F: command Byte

30: PWM_DUTY

CC 33 C3 3C: End Byte

TXD0 Output: **None**

8.2.14 Touch Panel Command

8.2.14.1 Touch Panel Calibration

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	**0xE4**	**55 AA 5A A5**	0xCC 0x33 0xC3 0x3C Or add the Delay time

Description: Touch Panel calibration command.

Class: Multi Byte Command

Code: E4ex, 228dec

Example : Set I/O1=Low

RXD0 Input : **AA E4 55 AA 5A A5 CC 33 C3 3C**

AA: Start Byte

E4: command Byte

55 AA 5A A5 : Data byte (Fixed)

CC 33 C3 3C: End Byte

TXD0 Output: **None**

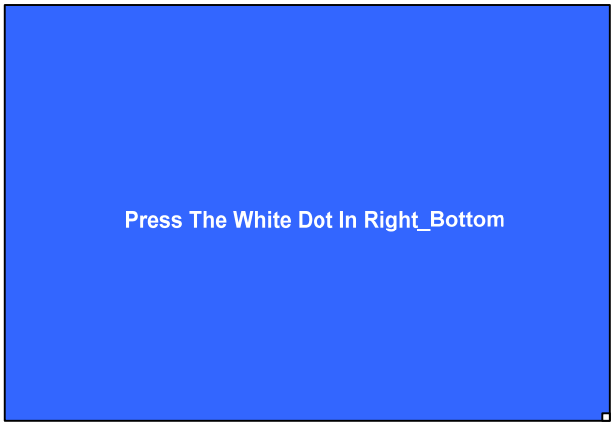
Control Touch Panel flow chart:

Step 1: RXD0 Input : **AA E4 55 AA 5A A5 CC 33 C3 3C**

Step 2: The LCD will show the following message. Please touch the LCD(0,0) LCD Left_Top point.

Step 3: The LCD will show the following message. Please touch the LCD(Max_X,0) LCD Right_Top point.

Step 4: The LCD will show the following message. Please touch the LCD(Max_X,MaxY) LCD Right_Bottom point.



Step 5:

After Touch Panel calibration command, the Touch panel position will be sent out by UART TXD0 pin. The Data Format :

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x73	Press_LCD_TP_X Press_LCD_TP_Y	0xCC 0x33 0xC3 0x3C

Example : If the user **press** the Position LCD_TP(128,64)

TXD0 Output: **0xAA 0x73 00 80 00 40 CC 33 C3 3C**

Start Byte	Command Byte	Data Bytes	End Byte
0XAA	0x72	Release_LCD_TP_X Release_LCD_TP_Y	0xCC 0x33 0xC3 0x3C

Example : If the user **press** the Position LCD_TP(128,64) ,then move to LCD_TP (128,65) and release on LCD_TP(128,66)

TXD0 Output:

0xAA 0x73 00 80 00 40 CC 33 C3 3C **0xAA 0x73 00 80 00 41 CC 33 C3 3C**

0xAA 0x72 00 80 00 42 CC 33 C3 3C

Note : The touch panel can not work on USB mode.

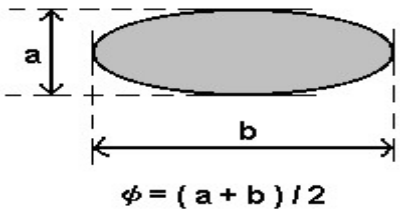
9 INCOMING INSPECTION STANDARD FOR TFT-LCD PANEL

DEFECT TYPE			LIMIT						Note
VISUAL DEFECT	INTERNAL	SPOT	φ < 0.15mm			Ignore			Note1
			0.15mm ≦ φ ≦ 0.5mm			N≦4			
			0.5mm < φ			N=0			
		FIBER	0.03mm<W≦0.1mm, L≦ 5mm			N≦3			Note1
			1.0mm < W, 1.5mm < L			N=0			
		POLARIZER BUBBLE	φ < 0.15mm			Ignore			Note1
			0.15mm ≦ φ ≦ 0.5mm			N≦2			
			0.5mm < φ			N=0			
		Mura	It' OK if mura is slight visible through 6%ND filter						
ELECTRICAL DEFECT	BRIGHT DOT		A Grade			B Grade			
			C Area	O Area	Total	C Area	O Area	Total	Note3
			N≦0	N≦2	N≦2	N≦2	N≦3	N≦5	Note2
	DARK DOT		N≦2	N≦3	N≦3	N≦3	N≦5	N≦8	
	TOTAL DOT		N≦4			N≦5	N≦6	N≦8	Note2
	TWO ADJACENT DOT		N≦0	N≦1 pair	N≦1 pair	N≦1 pair	N≦1 pair	N≦1 pair	Note4
	THREE OR MORE ADJACENT DOT		NOT ALLOWED						
	LINE DEFECT		NOT ALLOWED						

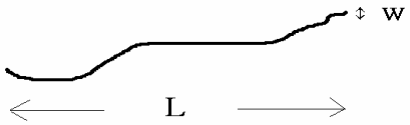
(1) One pixel consists of 3 sub-pixels, including R,G, and B dot.(Sub-pixel = Dot)

(2) LITTLE BRIGHT DOT ACCEPTITABLE UNDER 6 % ND-Filter

[Note1] W : Width[mm], L : Length[mm], N : Number, φ : Average Diameter

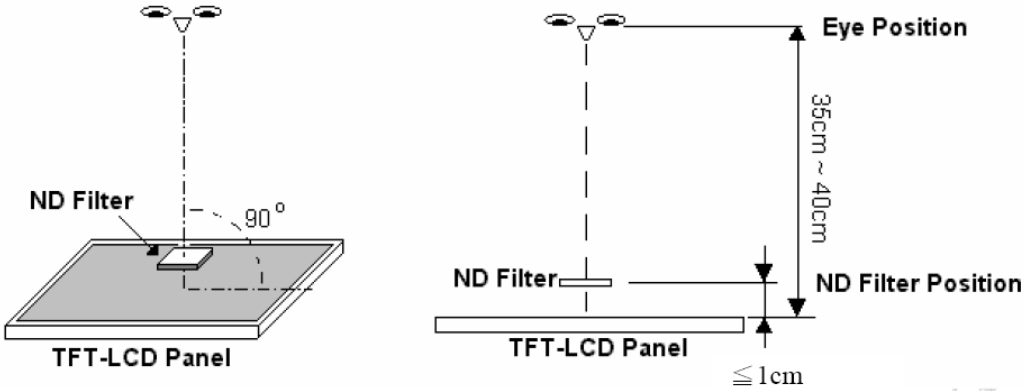


1. (White, black) Spot
2. Polarizer Bubble

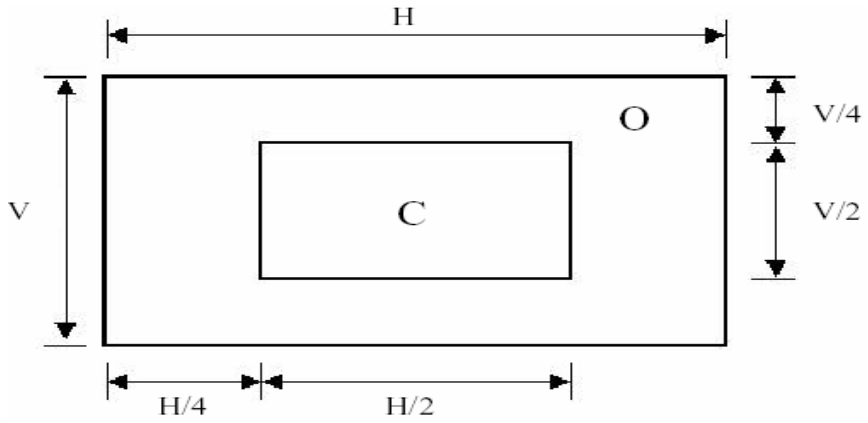


1. fiber

[Note2] Bright dot is defined through 6% transmission ND Filter as following.



[Note3]

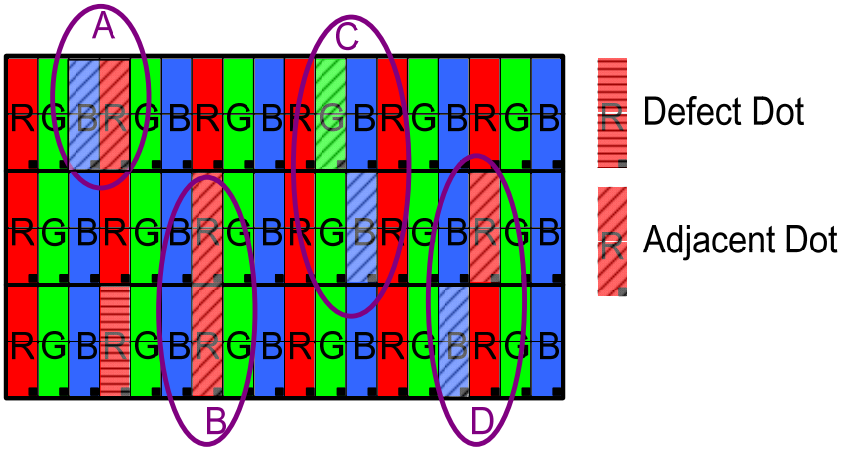


C Area: Center of display area

C Area: Outer of display area

[Note4]

Judge defect dot and adjacent dot as following. Allow below (as A, B, C and D status) adjacent defect dots, including bright and dart adjacent dot. And they will be counted 2 defect dots in total quantity.



(1) The defects that are not defined above and considered to be problem shall be reviewed and discussed by both parties.

(2) Defects on the Black Matrix, out of Display area, are not considered as a defect or counted.

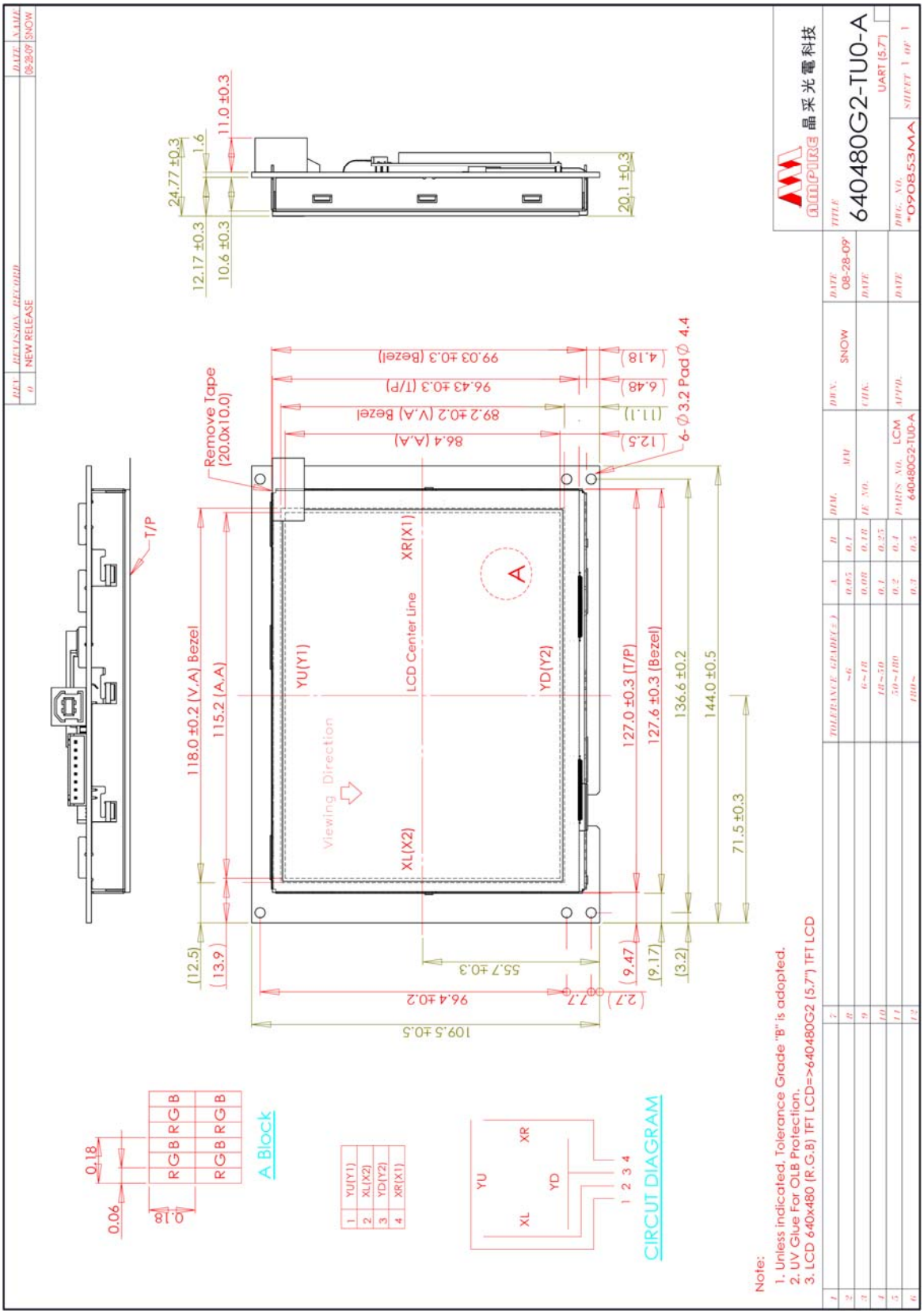
10 RELIABILITY TEST CONDITIONS

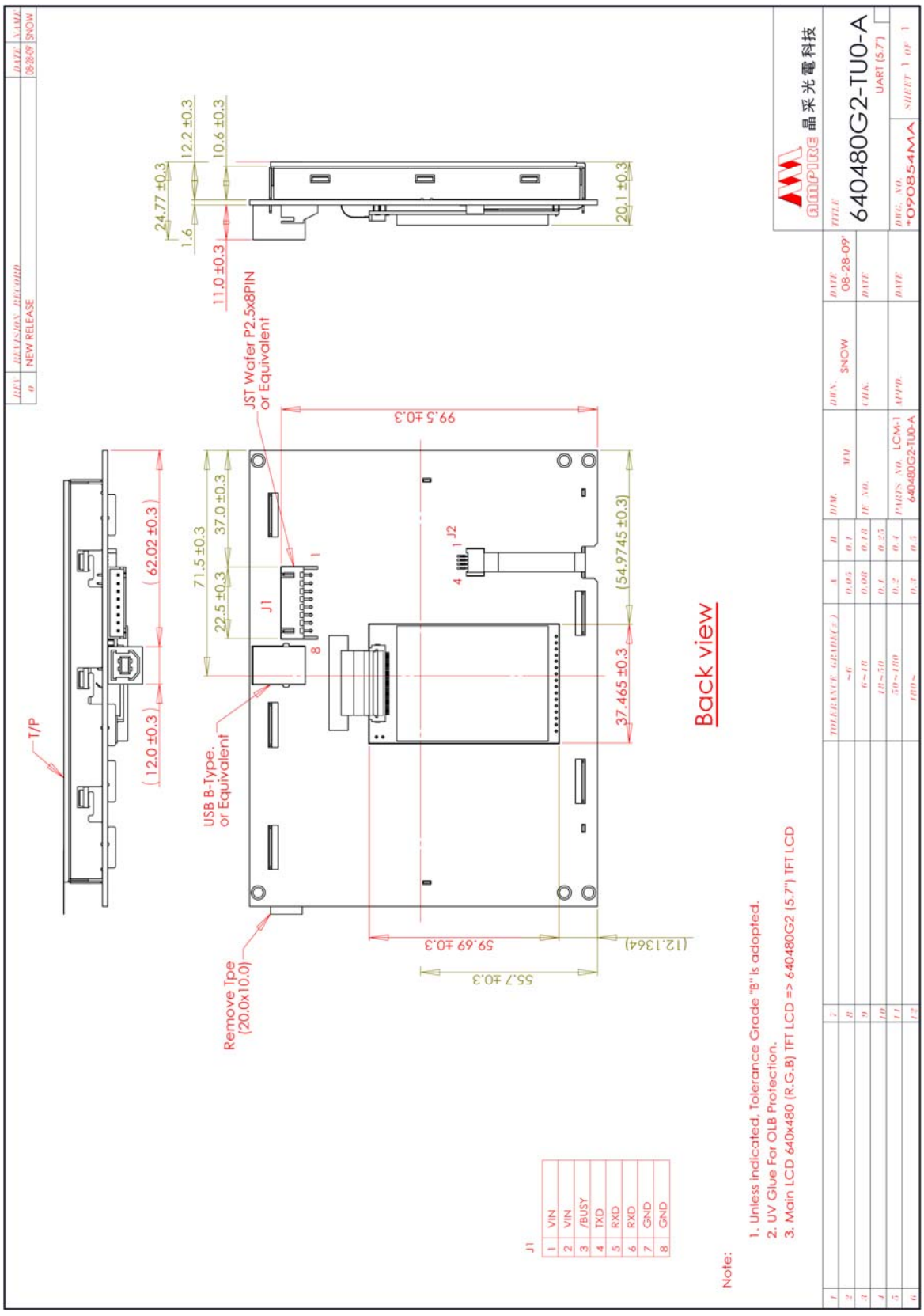
ITEM	CONDITIONS
HIGH TEMPERATURE OPERATION	70℃ , 240Hrs
HIGH TEMPERATURE AND HIGH HUMIDITY OPERATION	60℃ , 90%RH , 240Hrs
HIGH TEMPERATURE STORAGE	80℃ , 240Hrs
LOW TEMPERATURE OPERATION	-20℃ , 240Hrs
LOW TEMPERATURE STORAGE	-30℃ , 240Hrs
THERMAL SHOCK	-30℃(0.5Hr) ~80℃(0.5Hr) 200Cycle

11 OTHERS

AMIPRE will provide one year warranty for all products and three months warrantee for all repairing products.

12 OUTLINE DIMENSION




Back view
JST Wafer P2.5x8PIN or Equivalent
USB B-Type, or Equivalent
Remove Tpe (20.0x10.0)
T/P
J1
J2
Note:
1. Unless indicated, Tolerance Grade "B" is adopted.
2. UV Glue For OLB Protection.
3. Main LCD 640x480 (R.G.B) TFT LCD => 640480G2 (5.7") TFT LCD
640480G2-TU0-A
UART (5.7")
08-28-09
SNOW