



晶采光電科技股份有限公司  
AMPIRE CO., LTD.

## SPECIFICATIONS FOR LCD MODULE

<b>CUSTOMER</b>	
<b>CUSTOMER PART NO.</b>	
<b>AMPIRE PART NO.</b>	<b>AM-800600K7TMQW-U0H-A</b>
<b>APPROVED BY</b>	
<b>DATE</b>	

Preliminary Specification

Formal Specification

**AMPIRE CO., LTD.**

**4F., No.116, Sec. 1, Xintai 5th Rd., Xizhi Dist., New Taipei  
City221, Taiwan (R.O.C.)**

**新北市汐止區新台五路一段 116 號 4 樓(東方科學園區 A 棟)**

**TEL:886-2-26967269 , FAX:886-2-26967196 or 26967270**

Approved by	Checked by	Organized by
Patrick	Mark	Tank

\*This specification is subject to change without notice.

## RECORD OF REVISION

Revision Date	Page	Contents	Editor
2020/11/25	-	New Release	Tank

# 1. INSTRUCTION

Ampire 10.4" Display Module is a color active matrix TFT-LCD that uses amorphous silicon TFT as a switching device. This model is composed of a TFT-LCD panel, a driving circuit with LVDS interface. This TFT-LCD has a high resolution (800(R.G.B) X 600) and can display up to 262,144 colors.

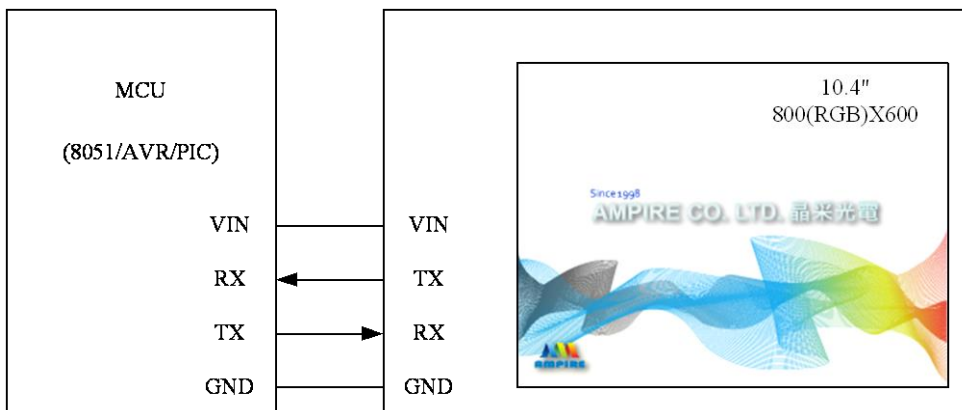
The TFT LCD module is a complete HMI module. It is able to handle the TFT panel by UART (Universal Asynchronous Receiver/Transmitter) interface. The user can easy use/control the TFT Panel by commands.

High Speed ASCII IC and rich function (Text support, draw pixel, draw Line, draw Arc, draw Rectangle) design will make the user's product success.

## 1.1 Features

- (1) Construction : a-Si TFT-LCD with driving system, White LED Backlight.
- (2) LCD type : Transmissive , Normally White
- (3) Number of the Colors : 262K colors (R,G,B 6 bit digital each)
- (4) [Change LED Driver to G5968C. PCB : 800600L-U0 Rev.D](#)

## 2. Application Diagram



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

### 3. PHYSICAL SPECIFICATIONS

Item	Specifications	unit
Display resolution	800RGB (W) x 600(H)	dots
Display size	10.4" (diagonal)	inch
Viewing Angle	6 o'clock (Gray Inversion)	-
Active area	211.2 (W) x 158.4(H)	mm
Pixel pitch	0.264 (W) x 0.264 (H)	um
Color configuration	R.G.B -stripe	-
Overall dimension	236.0(W) x 176.9(H) x 21.3(D)	mm
Backlight unit	LED	-
Brightness	500	cd/m2
Display color	262,144	colors
Contrast ratio	500	-

### 4. ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Condition	Min.	Max.	Unit	Remark
Power voltage	(Vin)	GND=0	-0.3	26.0	V	
Input voltage			-0.3	3.6	V	Note 1

Note1: RXD, TXD

#### 4.1. 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°C : 85% RH max

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

## 4.2 Electrical characteristics

### 4.2.1 DC Electrical characteristic

Typical operating conditions (GND=0V)

Item	Symbol	Min.	Typ.	Max.	Unit	Remark	
Power supply	V <sub>in</sub>	4.6	--	26.0	V		
Input Voltage for logic	H Level	V <sub>IH</sub>	0.7 VDD	-	VDD	V	Note 1
	L Level	V <sub>IL</sub>	0	-	0.3 VDD	V	
Power Supply current V <sub>in</sub> =5.0V	I <sub>vin</sub>		1050	1150	mA		
Power Supply current V <sub>in</sub> =12.0V	I <sub>vin</sub>		440	540	mA		
Power Supply current	IDD	--	1150	--	mA	Note 2	
Power consumption	P		5.25		W	Note 3	

Note1:RXD0, TXD0, RXD1, TXD1.

Note2: The internal power circuit will transfer the V<sub>in</sub> to VDD=+3.3V for the module power, LED Backlight turn ON.

Note3: Black pattern, V<sub>in</sub> = 5.0V, T<sub>a</sub>=25°C

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
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	

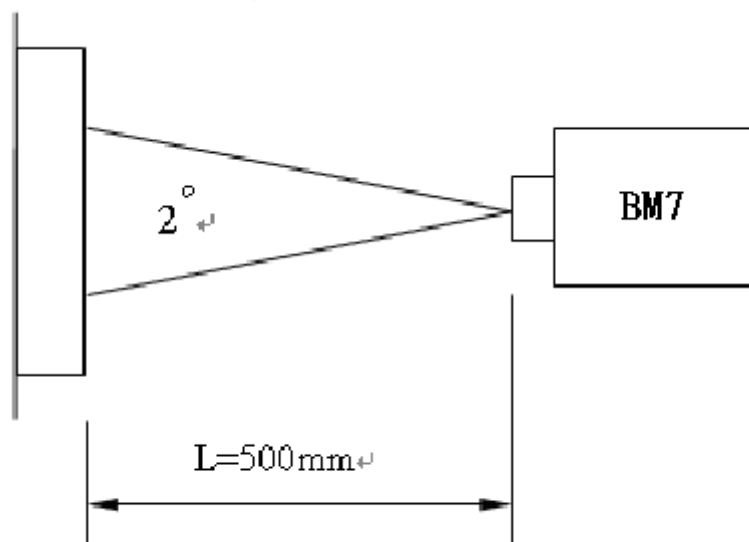
Item		Min.	Typ.	Max.	Unit	Remark
Uart Speed	COM1 (3.3V CMOS) User interface	1200		115200	bps	
	COM2 (3.3V CMOS) Function expand	--	115200	--	bps	
LCD Interface	CLOCK_H	10	25	33	MHz	
	CLOCK_L	1	5	10	MHz	
	VOH (IOH=-100uA)	0.9VLCD	--	--	V	
	VOL(IOL=100uA)	--	--	0.2 VLCD	V	
	IO	-20		+20	mA	
	Resolution		640*480	800*600	Dots	
	Color mode	16 bit, R5G6B5				
	ESD(HBM)		2		KV	

## 5. Optical Specification

Item	Symbol	Conditions	Min	Typ.	Max	Unit	Note	
Contrast Ration	CR	Viewing normal angle	-	500	-	-	(1)(2)(3)	
Response Time	TR	$\theta_x = \theta_y = 0$	-	5	10	ms	(1)(3)(5)	
	TF		-	15	20	ms		
Viewing Angle	Hor.	$\theta_{x+}$	Center CR>10	-	70	-	deg	(1)(2)(4)
		$\theta_{x-}$		-	70	-	deg	
	Ver.	$\theta_{y+}$		-	50	-	deg	
		$\theta_{y-}$		-	60	-	deg	
Module Chromaticity	Red	XR	Viewing normal angle $\theta_x = \theta_y = 0$	0.550	0.600	0.650	(1)(3)	
		YR		0.294	0.344	0.394		
	Green	XG		0.304	0.354	0.404		
		YG		0.529	0.579	0.629		
	Blue	XB		0.105	0.155	0.205		
		YB		0.076	0.126	0.176		
	White	XW		0.240	0.290	0.340		
		YW		0.268	0.318	0.368		
Brightness	-	$I_F=120\text{mA}$	400	500	-	cd/m <sup>2</sup>	(1)(3)	

NOTE:

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

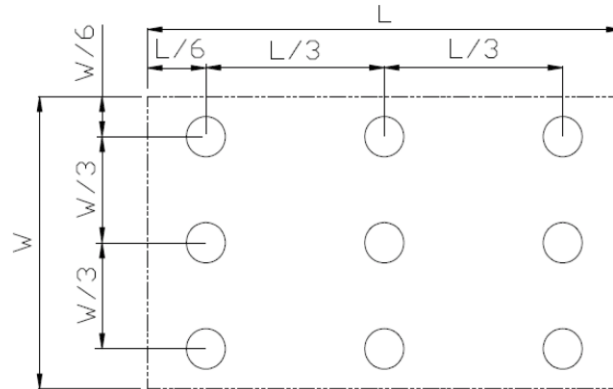


(2) Definition of Contrast Ratio:

**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

$$\Delta L = [ L(\text{MIN}) / L(\text{MAX}) ] \times 100\%$$



- (4) Definition of Viewing Angle( $\theta, \Phi$ ), refer to Fig9-2 as below :

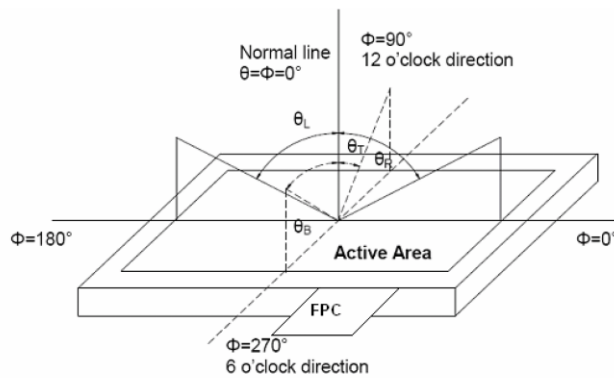


Fig. 1 Definition of viewing angle

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

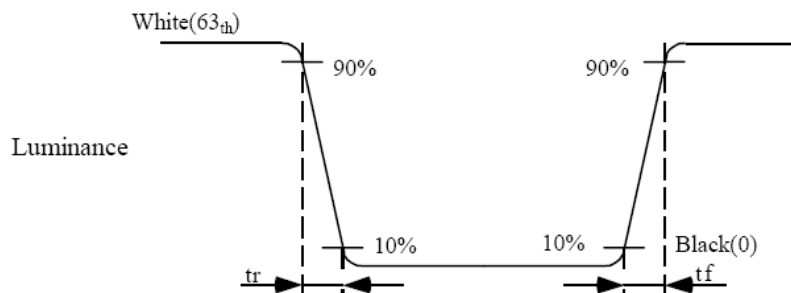
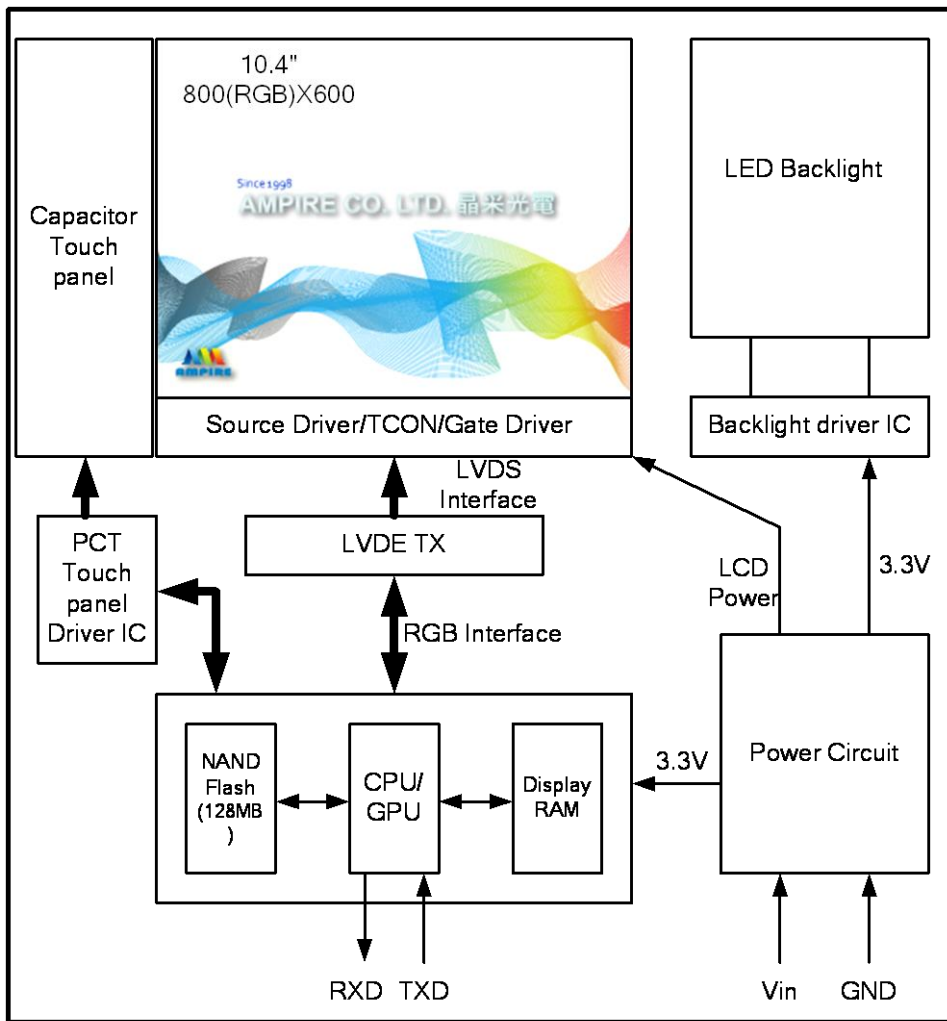


Fig9-3 Definition of Response Time(White-Black)



## 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**



## 7 Interface specifications

8 Pins Pitch 2.54 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
<b>Length(Byte)</b>	<b>1</b>	<b>1</b>	<b>0-249</b>	<b>4</b>
	0XAA	One Byte (see command Set)	Relative Data (max 249 bytes)	0xCC 0x33 0xC3 0x3C Or add the Delay time

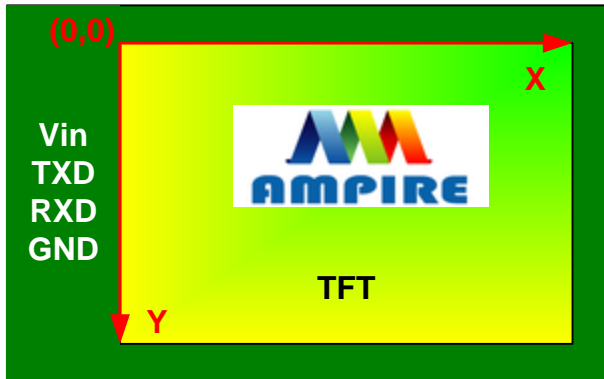
Baudrate_Set	Baudrate (bps)	Relative Delay Time
0x00	1200	12.5mS
0x01	2400	6.25mS
0X02	4800	3mS
0X03	9600	3mS
0X04	19200	2mS
0X05	38400	2mS
0X06	57600	1mS
0X07	115200	1mS

## 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, 224dec

TFT \_ ID:

TFT_ID	Resolution	Mode	Picture	Remark
0x04	800x480	SYNC	128	AM800480E
0x0A / 0x01	640x480	SYNC	153	AM640480G2
0x0D	800x600	SYNC	96	AM800600G2
0x0D	800x600	SYNC	96	TF800640-34-0

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 , Baud rate 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 Baud rate is 115200 N, 8 , 1 between power on 0~200mS , after 200mS the baud rate 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
<b>SET_COLOR 0x40</b>							
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
<b>SET_FONT_PITCH 0x41</b>							
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
<b>SET_BACKGROUND_COLOR By X,Y 0x42</b>							
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
<b>SET_FRONT_COLOR By X,Y 0x42</b>							
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
<b>SET_CURSOR_MODE 0x44</b>							
<b>Mode</b>							
<b>X MSB</b>							
<b>X MSB</b>							
<b>Y MSB</b>							
<b>Y MSB</b>							
<b>Cursor Wide</b>							
<b>Cursor High</b>							

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
<b>PRINT_STRING_8x8 0x53</b>							
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
<b>PRINT_STRING_16x16 0x54</b>							
<b>X MSB</b>							
<b>X MSB</b>							
<b>Y MSB</b>							
<b>Y MSB</b>							
<b>ASCII</b>							
<b>ASCII</b>							
~							
<b>ASCII</b>							
<b>ASCII</b>							

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
<b>PRINT_STRING_32x32 0x55</b>							
<b>X MSB</b>							
<b>X MSB</b>							
<b>Y MSB</b>							
<b>Y MSB</b>							
<b>ASCII</b>							
<b>ASCII</b>							
~							
<b>ASCII</b>							
<b>ASCII</b>							

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
<b>PRINT_STRING_12x12 0x6E</b>							
<b>X MSB</b>							
<b>X MSB</b>							
<b>Y MSB</b>							
<b>Y MSB</b>							
<b>ASCII</b>							
<b>ASCII</b>							
~							
<b>ASCII</b>							
<b>ASCII</b>							

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
<b>PRINT_STRING_24x24 0x6F</b>							
<b>X MSB</b>							
<b>X MSB</b>							
<b>Y MSB</b>							
<b>Y MSB</b>							
<b>ASCII</b>							
<b>ASCII</b>							
~							
<b>ASCII</b>							
<b>ASCII</b>							

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. <ul style="list-style-type: none"> <li>● Single area can be set as 12x12 or 16x16 (GBK encode)</li> <li>● The area can be combined for supper large font library.</li> </ul>	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
<b>PRINT_STRING_Custom Font</b>							
<b>X MSB</b>							
<b>X MSB</b>							
<b>Y MSB</b>							
<b>Y MSB</b>							
<b>Lib_ID</b>							
<b>C_mode</b>							
<b>C_dots</b>							
<b>Color</b>							
<b>Bcolor</b>							
<b>Font code</b>							
<b>Font code</b>							
~							
<b>Font code</b>							
<b>Font code</b>							



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 <sub>0</sub> MSB X <sub>0</sub> LSB Y <sub>0</sub> MSB Y <sub>0</sub> LSB X <sub>1</sub> MSB X <sub>1</sub> LSB Y <sub>1</sub> MSB Y <sub>1</sub> LSB ~~ X <sub>k</sub> MSB X <sub>k</sub> LSB Y <sub>k</sub> MSB Y <sub>k</sub> LSB	0xCC 0x33 0xC3 0x3C Or add the Delay time

**Description:** Put the background color to the point (X<sub>0</sub>,Y<sub>0</sub>) , (X<sub>1</sub>,Y<sub>1</sub>) ,... (X<sub>k</sub>,Y<sub>k</sub>)

**Class:** Multi Byte Command

**Code:** 50hex, 80dec

7	6	5	4	3	2	1	0
<b>Put Pixel by Background color</b>							
X <sub>0</sub> MSB							
X <sub>0</sub> LSB							
Y <sub>0</sub> MSB							
Y <sub>0</sub> LSB							
X <sub>1</sub> MSB							
X <sub>1</sub> LSB							
Y <sub>1</sub> MSB							
Y <sub>1</sub> LSB							
~~							
X <sub>k</sub> MSB							
X <sub>k</sub> LSB							
Y <sub>k</sub> MSB							
Y <sub>k</sub> LSB							

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 <sub>0</sub> MSB X <sub>0</sub> LSB Y <sub>0</sub> MSB Y <sub>0</sub> LSB X <sub>1</sub> MSB X <sub>1</sub> LSB Y <sub>1</sub> MSB Y <sub>1</sub> LSB ~~ X <sub>k</sub> MSB X <sub>k</sub> LSB Y <sub>k</sub> MSB Y <sub>k</sub> LSB	0xCC 0x33 0xC3 0x3C Or add the Delay time

**Description:** Put the front color to the point (X<sub>0</sub>,Y<sub>0</sub>) , (X<sub>1</sub>,Y<sub>1</sub>) ,... (X<sub>k</sub>,Y<sub>k</sub>)

**Class:** Multi Byte Command

**Code:** 51hex, 81dec

7	6	5	4	3	2	1	0
<b>Put Pixel by Front color</b>							
<b>X<sub>0</sub> MSB</b>							
<b>X<sub>0</sub> LSB</b>							
<b>Y<sub>0</sub> MSB</b>							
<b>Y<sub>0</sub> LSB</b>							
<b>X<sub>1</sub> MSB</b>							
<b>X<sub>1</sub> LSB</b>							
<b>Y<sub>1</sub> MSB</b>							
<b>Y<sub>1</sub> LSB</b>							
~~							
<b>X<sub>k</sub> MSB</b>							
<b>X<sub>k</sub> LSB</b>							
<b>Y<sub>k</sub> MSB</b>							
<b>Y<sub>k</sub> LSB</b>							

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	<b>X MSB</b> <b>X LSB</b> <b>Ys MSB</b> <b>Ys LSB</b> <b>YE MSB</b> <b>YE LSB</b> <b>Clear Color MSB</b> <b>Clear Color LSB</b> <b>Y1 MSB</b> <b>Y1 LSB</b> <b>Color(X,Y1) MSB</b> <b>Color(X,Y1) LSB</b> <b>Y2 MSB</b> <b>Y2 LSB</b> <b>Color(X,Y2) MSB</b> <b>Color(X,Y2) LSB</b> ~~ <b>Yk MSB</b> <b>Yk LSB</b> <b>Color(X,Yk) MSB</b> <b>Color(X,Yk) LSB</b>	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 ,Ys ) ( X ,YE) 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**

## 8.2.6 Draw 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 <sub>0</sub> MSB X <sub>0</sub> LSB Y <sub>0</sub> MSB Y <sub>0</sub> LSB X <sub>1</sub> MSB X <sub>1</sub> LSB Y <sub>1</sub> MSB Y <sub>1</sub> LSB ~~ X <sub>k</sub> MSB X <sub>k</sub> LSB Y <sub>k</sub> MSB Y <sub>k</sub> LSB	0xCC 0x33 0xC3 0x3C Or add the Delay time

**Description:** Draw the line (X<sub>0</sub>,Y<sub>0</sub>) ( X<sub>1</sub>,Y<sub>1</sub>) ....(X<sub>k</sub>,Y<sub>k</sub>)

**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 : (X<sub>0</sub>,Y<sub>0</sub>)=(128,64)

00 80 00 00 : (X<sub>1</sub>,Y<sub>1</sub>)=(128,0)

00 00 00 00 : (X<sub>2</sub>,Y<sub>2</sub>)=(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 <sub>0</sub> MSB X <sub>0</sub> LSB Y <sub>0</sub> MSB Y <sub>0</sub> LSB X <sub>1</sub> MSB X <sub>1</sub> LSB Y <sub>1</sub> MSB Y <sub>1</sub> LSB ~~ X <sub>k</sub> MSB X <sub>k</sub> LSB Y <sub>k</sub> MSB Y <sub>k</sub> LSB	0xCC 0x33 0xC3 0x3C Or add the Delay time

**Description:** Draw the line (X<sub>0</sub>,Y<sub>0</sub>) ( X<sub>1</sub>,Y<sub>1</sub>) ....(X<sub>k</sub>,Y<sub>k</sub>)

**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 CC 33 C3 3C**

AA: Start Byte

56: command Byte

00 00 00 00 : (X,Y)=(0,0)

00 80 00 40 : (X<sub>0</sub>,Y<sub>0</sub>)=(128,64)

00 80 00 00 : (X<sub>1</sub>,Y<sub>1</sub>)=(128,0)

00 00 00 00 : (X<sub>2</sub>,Y<sub>2</sub>)=(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 <sub>0</sub> MSB X <sub>0</sub> LSB Y <sub>0</sub> MSB Y <sub>0</sub> LSB R <sub>0</sub> ~~ TYPE X <sub>k</sub> MSB X <sub>k</sub> LSB Y <sub>k</sub> MSB Y <sub>k</sub> LSB R <sub>k</sub>	0xCC 0x33 0xC3 0x3C Or add the Delay time

**Description:** Draw the Circle by the specify center (X<sub>0</sub>,Y<sub>0</sub>) , Radius R<sub>0</sub> .

**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 : (X<sub>0</sub>,Y<sub>0</sub>)=(128,64)

00 80 00 00 : (X<sub>1</sub>,Y<sub>1</sub>)=(128,0)

00 00 00 00 : (X<sub>2</sub>,Y<sub>2</sub>)=(0,0)

CC 33 C3 3C: End Byte

TXD0 Output: **None**

**See also:** [SET COLOR](#) , [SET FRONT COLOR By X,Y](#)



## 8.2.8 Draw 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	<b>X<sub>S0</sub> MSB</b> <b>X<sub>S0</sub> LSB</b> <b>Y<sub>S0</sub> MSB</b> <b>Y<sub>S0</sub> LSB</b> <b>X<sub>E0</sub> MSB</b> <b>X<sub>E0</sub> LSB</b> <b>Y<sub>E0</sub> MSB</b> <b>Y<sub>E0</sub> LSB</b> ~~ <b>X<sub>Sk</sub> MSB</b> <b>X<sub>Sk</sub> LSB</b> <b>Y<sub>Sk</sub> MSB</b> <b>Y<sub>Sk</sub> LSB</b> <b>X<sub>Ek</sub> MSB</b> <b>X<sub>Ek</sub> LSB</b> <b>Y<sub>Ek</sub> MSB</b> <b>Y<sub>Ek</sub> LSB</b>	0xCC 0x33 0xC3 0x3C Or add the Delay time

**Description:** Draw the front color Rectangle by the specify the left-up point (X<sub>S0</sub>,Y<sub>S0</sub>) , Right-Down points(X<sub>E0</sub>,Y<sub>E0</sub>) .

**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 : (X<sub>S0</sub>,Y<sub>S0</sub>)=(64,64)

00 80 00 80 : (X<sub>E0</sub>,Y<sub>E0</sub>)=(128,64)

00 0A 00 0A : (X<sub>S1</sub>,Y<sub>S1</sub>)=(10,10)

00 14 00 14 : (X<sub>S1</sub>,Y<sub>S1</sub>)=(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	<b>X<sub>S0</sub> MSB</b> <b>X<sub>S0</sub> LSB</b> <b>Y<sub>S0</sub> MSB</b> <b>Y<sub>S0</sub> LSB</b> <b>X<sub>E0</sub> MSB</b> <b>X<sub>E0</sub> LSB</b> <b>Y<sub>E0</sub> MSB</b> <b>Y<sub>E0</sub> LSB</b> ~~ <b>X<sub>Sk</sub> MSB</b> <b>X<sub>Sk</sub> LSB</b> <b>Y<sub>Sk</sub> MSB</b> <b>Y<sub>Sk</sub> LSB</b> <b>X<sub>EK</sub> MSB</b> <b>X<sub>EK</sub> LSB</b> <b>Y<sub>EK</sub> MSB</b> <b>Y<sub>EK</sub> LSB</b>	0xCC 0x33 0xC3 0x3C Or add the Delay time

**Description:** Draw the background color Rectangle by the specify the left-up point (X<sub>S0</sub>,Y<sub>S0</sub>) , Right-Down points(X<sub>E0</sub>,Y<sub>E0</sub>) .

**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 : (X<sub>S0</sub>,Y<sub>S0</sub>)=(64,64)

00 80 00 80 : (X<sub>E0</sub>,Y<sub>E0</sub>)=(128,64)

00 0A 00 0A : (X<sub>S1</sub>,Y<sub>S1</sub>)=(10,10)

00 14 00 14 : (X<sub>S1</sub>,Y<sub>S1</sub>)=(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 <sub>S0</sub> MSB X <sub>S0</sub> LSB Y <sub>S0</sub> MSB Y <sub>S0</sub> LSB X <sub>E0</sub> MSB X <sub>E0</sub> LSB Y <sub>E0</sub> MSB Y <sub>E0</sub> LSB ~~ X <sub>Sk</sub> MSB X <sub>Sk</sub> LSB Y <sub>Sk</sub> MSB Y <sub>Sk</sub> LSB X <sub>Ek</sub> MSB X <sub>Ek</sub> LSB Y <sub>Ek</sub> MSB Y <sub>Ek</sub> 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<sub>S0</sub>,Y<sub>S0</sub>) , Right-Down points(X<sub>E0</sub>,Y<sub>E0</sub>) .

**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	<b>X<sub>S0</sub> MSB</b> <b>X<sub>S0</sub> LSB</b> <b>Y<sub>S0</sub> MSB</b> <b>Y<sub>S0</sub> LSB</b> <b>X<sub>E0</sub> MSB</b> <b>X<sub>E0</sub> LSB</b> <b>Y<sub>E0</sub> MSB</b> <b>Y<sub>E0</sub> LSB</b> ~~ <b>X<sub>Sk</sub> MSB</b> <b>X<sub>Sk</sub> LSB</b> <b>Y<sub>Sk</sub> MSB</b> <b>Y<sub>Sk</sub> LSB</b> <b>X<sub>Ek</sub> MSB</b> <b>X<sub>Ek</sub> LSB</b> <b>Y<sub>Ek</sub> MSB</b> <b>Y<sub>Ek</sub> LSB</b>	0xCC 0x33 0xC3 0x3C Or add the Delay time

**Description:** Full the Rectangle with front color by the specify the left-up point (X<sub>S0</sub>,Y<sub>S0</sub>) , Right-Down points(X<sub>E0</sub>,Y<sub>E0</sub>) .

**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	<b>X<sub>S0</sub> MSB</b> <b>X<sub>S0</sub> LSB</b> <b>Y<sub>S0</sub> MSB</b> <b>Y<sub>S0</sub> LSB</b> <b>X<sub>E0</sub> MSB</b> <b>X<sub>E0</sub> LSB</b> <b>Y<sub>E0</sub> MSB</b> <b>Y<sub>E0</sub> LSB</b> ~~ <b>X<sub>Sk</sub> MSB</b> <b>X<sub>Sk</sub> LSB</b> <b>Y<sub>Sk</sub> MSB</b> <b>Y<sub>Sk</sub> LSB</b> <b>X<sub>EK</sub> MSB</b> <b>X<sub>EK</sub> LSB</b> <b>Y<sub>EK</sub> MSB</b> <b>Y<sub>EK</sub> LSB</b>	0xCC 0x33 0xC3 0x3C Or add the Delay time

**Description:** Reverse the area color by the specify the left-up point (X<sub>S0</sub>,Y<sub>S0</sub>) , Right-Down points(X<sub>E0</sub>,Y<sub>E0</sub>) .

**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	<b>X<sub>S0</sub> MSB</b> <b>X<sub>S0</sub> LSB</b> <b>Y<sub>S0</sub> MSB</b> <b>Y<sub>S0</sub> LSB</b> <b>X<sub>E0</sub> MSB</b> <b>X<sub>E0</sub> LSB</b> <b>Y<sub>E0</sub> MSB</b> <b>Y<sub>E0</sub> LSB</b> <b>N<sub>0</sub> ( N &lt;=16)</b> ~~ <b>X<sub>Sk</sub> MSB</b> <b>X<sub>Sk</sub> LSB</b> <b>Y<sub>Sk</sub> MSB</b> <b>Y<sub>Sk</sub> LSB</b> <b>X<sub>E<sub>k</sub></sub> MSB</b> <b>X<sub>E<sub>k</sub></sub> LSB</b> <b>Y<sub>E<sub>k</sub></sub> MSB</b> <b>Y<sub>E<sub>k</sub></sub> LSB</b> <b>N<sub>k</sub> ( N &lt;=16)</b>	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<sub>S0</sub>,Y<sub>S0</sub>) , Right-Down points(X<sub>E0</sub>,Y<sub>E0</sub>) .

**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 : (X<sub>S0</sub>,Y<sub>S0</sub>)=(64,64)

00 80 00 80 : (X<sub>E0</sub>,Y<sub>E0</sub>)=(128,64)

00 10 00 10 : (X<sub>S1</sub>,Y<sub>S1</sub>)=(16,16)

00 30 00 30 : (X<sub>S1</sub>,Y<sub>S1</sub>)=(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	<b>X<sub>S0</sub> MSB</b> <b>X<sub>S0</sub> LSB</b> <b>Y<sub>S0</sub> MSB</b> <b>Y<sub>S0</sub> LSB</b> <b>X<sub>E0</sub> MSB</b> <b>X<sub>E0</sub> LSB</b> <b>Y<sub>E0</sub> MSB</b> <b>Y<sub>E0</sub> LSB</b> <b>N<sub>0</sub> ( N &lt;=16)</b> ~~ <b>X<sub>Sk</sub> MSB</b> <b>X<sub>Sk</sub> LSB</b> <b>Y<sub>Sk</sub> MSB</b> <b>Y<sub>Sk</sub> LSB</b> <b>X<sub>Ek</sub> MSB</b> <b>X<sub>Ek</sub> LSB</b> <b>Y<sub>Ek</sub> MSB</b> <b>Y<sub>Ek</sub> LSB</b> <b>N<sub>k</sub> ( N &lt;=16)</b>	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<sub>S0</sub>,Y<sub>S0</sub>) , Right-Down points(X<sub>E0</sub>,Y<sub>E0</sub>) .

**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 : (X<sub>S0</sub>,Y<sub>S0</sub>)=(64,64)

00 80 00 80 : (X<sub>E0</sub>,Y<sub>E0</sub>)=(128,64)

00 10 00 10 : (X<sub>S1</sub>,Y<sub>S1</sub>)=(16,16)

00 30 00 30 : (X<sub>S1</sub>,Y<sub>S1</sub>)=(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	<b>X<sub>S0</sub> MSB</b> <b>X<sub>S0</sub> LSB</b> <b>Y<sub>S0</sub> MSB</b> <b>Y<sub>S0</sub> LSB</b> <b>X<sub>E0</sub> MSB</b> <b>X<sub>E0</sub> LSB</b> <b>Y<sub>E0</sub> MSB</b> <b>Y<sub>E0</sub> LSB</b> <b>N<sub>0</sub> ( N &lt;=16)</b> ~~ <b>X<sub>Sk</sub> MSB</b> <b>X<sub>Sk</sub> LSB</b> <b>Y<sub>Sk</sub> MSB</b> <b>Y<sub>Sk</sub> LSB</b> <b>X<sub>Ek</sub> MSB</b> <b>X<sub>Ek</sub> LSB</b> <b>Y<sub>Ek</sub> MSB</b> <b>Y<sub>Ek</sub> LSB</b> <b>N<sub>k</sub> ( N &lt;=16)</b>	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<sub>S0</sub>,Y<sub>S0</sub>) , Right-Down points(X<sub>E0</sub>,Y<sub>E0</sub>) . 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 : (X<sub>S0</sub>,Y<sub>S0</sub>)=(64,64)

00 80 00 80 : (X<sub>E0</sub>,Y<sub>E0</sub>)=(128,64)

00 10 00 10 : (X<sub>S1</sub>,Y<sub>S1</sub>)=(16,16)

00 30 00 30 : (X<sub>S1</sub>,Y<sub>S1</sub>)=(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	<b>X<sub>S0</sub> MSB</b> <b>X<sub>S0</sub> LSB</b> <b>Y<sub>S0</sub> MSB</b> <b>Y<sub>S0</sub> LSB</b> <b>X<sub>E0</sub> MSB</b> <b>X<sub>E0</sub> LSB</b> <b>Y<sub>E0</sub> MSB</b> <b>Y<sub>E0</sub> LSB</b> <b>N<sub>0</sub> ( N &lt;=16)</b> ~~ <b>X<sub>Sk</sub> MSB</b> <b>X<sub>Sk</sub> LSB</b> <b>Y<sub>Sk</sub> MSB</b> <b>Y<sub>Sk</sub> LSB</b> <b>X<sub>Ek</sub> MSB</b> <b>X<sub>Ek</sub> LSB</b> <b>Y<sub>Ek</sub> MSB</b> <b>Y<sub>Ek</sub> LSB</b> <b>N<sub>k</sub> ( N &lt;=16)</b>	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<sub>S0</sub>,Y<sub>S0</sub>) , Right-Down points(X<sub>E0</sub>,Y<sub>E0</sub>) . 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 : (X<sub>S0</sub>,Y<sub>S0</sub>)=(64,64)

00 80 00 80 : (X<sub>E0</sub>,Y<sub>E0</sub>)=(128,64)

00 10 00 10 : (X<sub>S1</sub>,Y<sub>S1</sub>)=(16,16)

00 30 00 30 : (X<sub>S1</sub>,Y<sub>S1</sub>)=(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	<b>PICNUM</b> <b>X<sub>S0</sub> MSB</b> <b>X<sub>S0</sub> LSB</b> <b>Y<sub>S0</sub> MSB</b> <b>Y<sub>S0</sub> LSB</b> <b>X<sub>E0</sub> MSB</b> <b>X<sub>E0</sub> LSB</b> <b>Y<sub>E0</sub> MSB</b> <b>Y<sub>E0</sub> LSB</b> <b>X MSB</b> <b>X LSB</b> <b>Y MSB</b> <b>Y LSB</b>	0xCC 0x33 0xC3 0x3C Or add the Delay time

**Description:** The command will show the area of the picture which left-up point (X<sub>S0</sub>,Y<sub>S0</sub>) , Right-Down points(X<sub>E0</sub>,Y<sub>E0</sub>) 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  
 $640 * 100 + 80 = 64080(\text{dec}) = 0x0FA50(\text{hex})$

**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

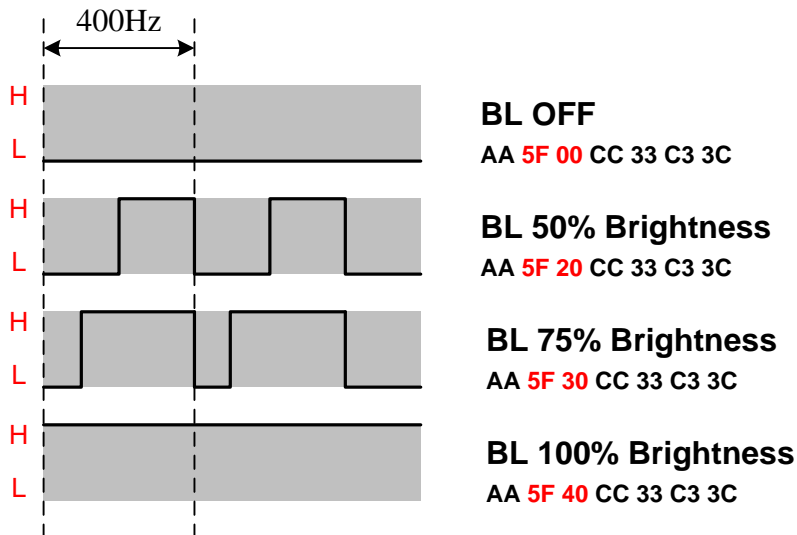
TXD0 Output: None

### 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.



**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**



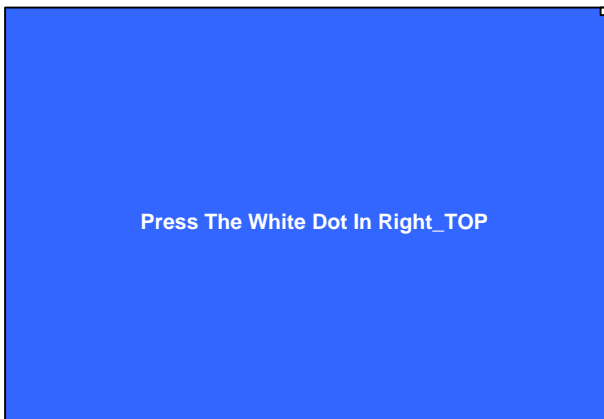
Control Touch Panel flow chart:

**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.**

## 11. QUALITY AND RELIABILITY

Test Item	Test Conditions	Note
High Temperature Operation	60±3°C , t=240 hrs	
Low Temperature Operation	-5±3°C , t=240 hrs	
High Temperature Storage	70±3°C , t=240 hrs	1,2
Low Temperature Storage	-30±3°C , t=240 hrs	1,2
Thermal Shock Test	-20°C ~ 25°C ~ 70°C 30 m in. 5 min. 30 min. ( 1 cycle ) Total 5 cycle	1,2
Storage Humidity Test	60 °C, Humidity 90%, 240 hrs	1,2
Vibration Test (Packing)	Sweep frequency : 10 ~ 55 ~ 10 Hz/1min Amplitude : 0.75mm Test direction : X.Y.Z/3 axis Duration : 30min/each axis	2

Note 1 : Condensation of water is not permitted on the module.

Note 2 : The module should be inspected after 1 hour storage in normal conditions  
(15-35°C , 45-65%RH).

Definitions of life end point :

- Current drain should be smaller than the specific value.
- Function of the module should be maintained.
- Appearance and display quality should not have degraded noticeably.
- Contrast ratio should be greater than 50% of the initial value.

## **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.

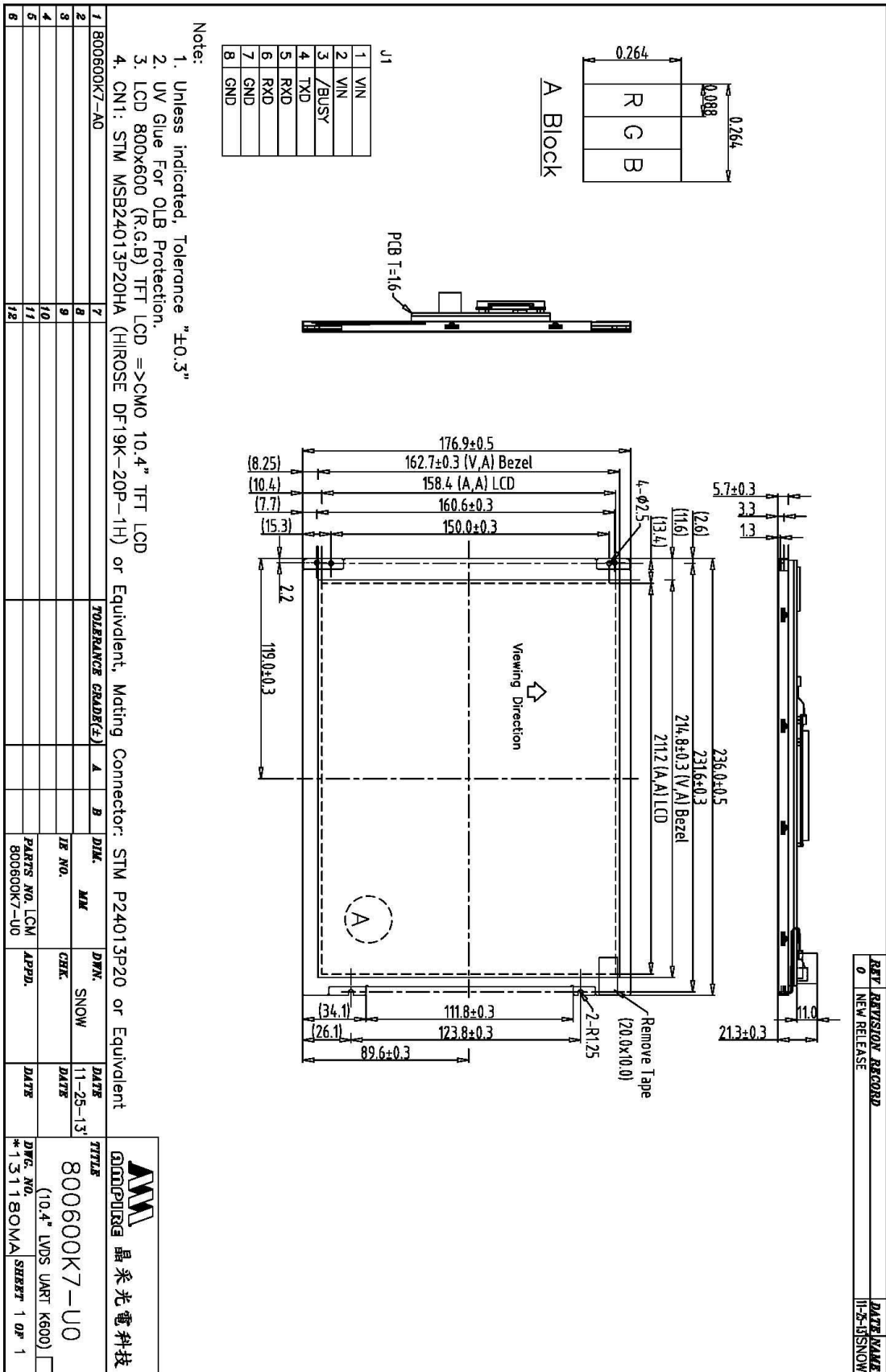
#### **12.4 Operating precautions**

- 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 drive 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: 0.2V<sub>dd</sub> or less and H level: 0.8V<sub>dd</sub> 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.

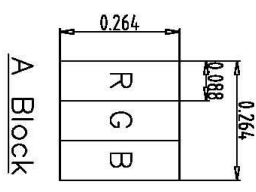
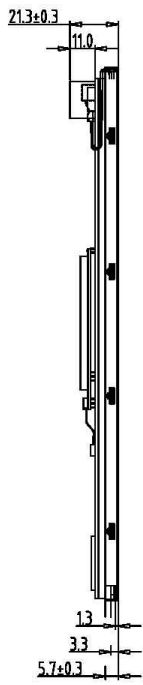
#### **12.5 Other**

- 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 warranty for all products and three months warrantee for all repairing products.

# 13. OUTLINE DIMENSION

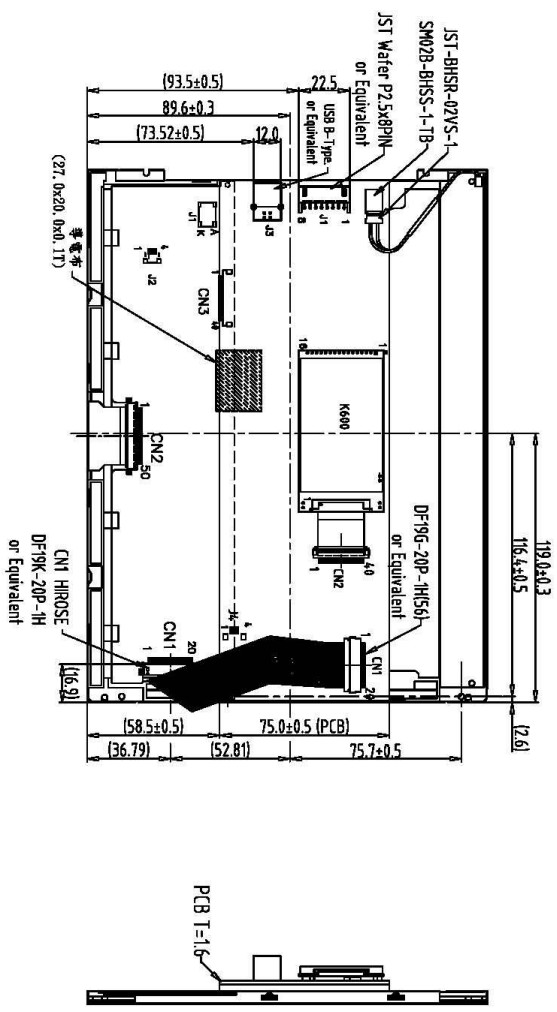


REV	REVISION RECORD	DATE	NAME
0	NEW RELEASE	11-5-13	SNOW



J1	1	VIN
	2	VIN
	3	/BUSY
	4	TXD
	5	RXD
	6	RXD
	7	GND
	8	GND

Note:  
 1. Unless indicated, Tolerance "±0.3"  
 2. UV Glue For OLB Protection.  
 3. LCD 800x600 (R,G,B) TFT LCD =>CMO 10.4" TFT LCD  
 4. CN1: STM MSB24013P20HA (HIROSE DF19K-20P-1H) or Equivalent; Mating Connector: STM P24013P20 or Equivalent



Back View

1	800600K7-A0	7		TOLERANCE GRADIENT	A	B	DIK.	MM	DWN.	SNOW	DATE	11-25-13	TITLE	800600K7-U0	DMG. NO.	*131181MA	SHEET	1 OF 1
2		8							CHEK.		DATE		AMM	晶采光電科技				
3		9											COMPUING					
4		10											(10.4" LVDS UART K600)					
5		11							APPD.		DATE							
6		12																