TFT Panel of Serial Uart Interface

Application Note

V0

LCM-EU02401KL

(EzUIH0024D1)

洪泰科技（东莞）有限公司

深圳市鑫洪泰电子科技有限公司

ADD：广东省东莞市凤岗镇东深公路 208 号天安数码城 N3 栋 3 楼

1 Document revision history :

DOCUMENT REVISION	DATE	DESCRIPTION	PREPARED BY	APPROVED BY
0	2021-06-15	First Release.	Liu. YL	

1. General Feature:

EU02401KL is one of the EzUI series serial intelligent display modules. The module display is a 2.4-inch color TFT display, 240×320 dot matrix, 16-bit color depth; there is a 16M-byte Flash memory inside the module to store pictures, GIF animations, Information such as font library. The module is powered by a USB interface of 5.0V DC, and the external interface of the module is a serial UART (TTL level) interface. The interface is simple and easy to operate; convenient and simple interface operations can be carried out with various MCUs.

The biggest feature of EzUI series modules is easy to use and rich in functions, which can be simple or complex for users. The MCU program inside the module already contains the UART serial port protocol. The user can use the PC upper computer software (UI_Editor) to directly develop the product's UI display interface on the computer, and use the resource files generated by the tool software through the USB interface and the communication software (LT_VCOM_GUI)) Update the internal MCU program and SPI Flash data.

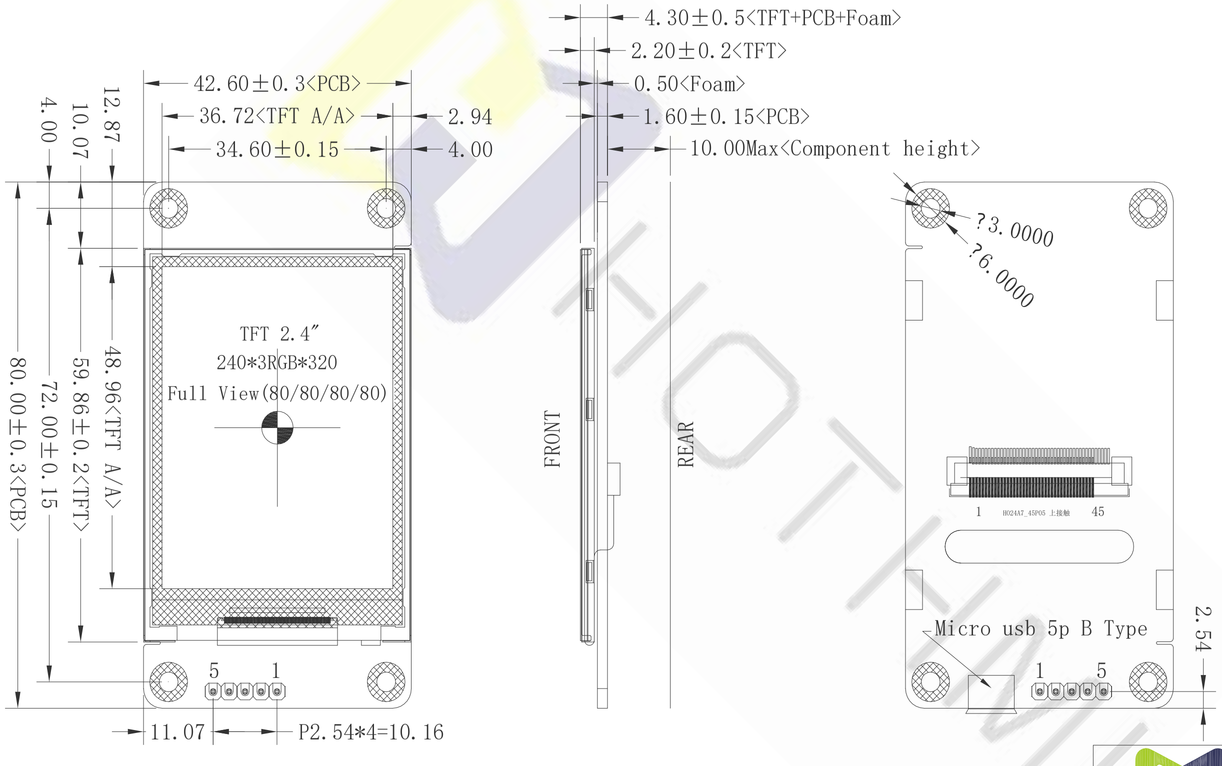
The built-in serial port command function of the module includes picture display, GIF animation display, cycle picture display, boot screen display, vertical bar display, text string display, two-dimensional code generation, and geometric graphics display such as drawing lines, drawing circles, Functions such as drawing triangles and rectangles

Item	Standard Value	Unit
LCD Display Size	2.4"	--
LCD Number of Pixels	240(H)x3(RGB)x320(V)	--
LCD Active Area	36.72(H) x 48.96(V)	mm
LCD Display Colors	262K (16-bit/565 format)	Color
Viewing Direction	ALL O'clock	-
LCM Outline Dimension	42.60(H)×80.00(V)×4.30(D)	mm
LCM Driver Voltage	+5.0V	V
Backlight	White LED	-
Touch type	No touch/button operation	-
Real Time Clock	Built-in	-
Operation Temperature	-20~70	℃
Storage Temperature	-30~80	℃

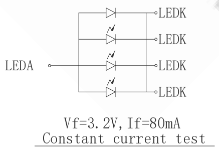
Note:Upgraded MCU data can be switched to 320*240 horizontal screen mode, the current default 240*320 vertical screen mode default

2. Outline Dimensions

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



USB_SIGNAL		Pin_SIGNAL	
1	DC5V	1	DC5V
2	USBDM	2	TXD
3	USBDP	3	RXD
4	NC	4	GND
5	GND	5	BUSY



Vf=3.2V,If=80mA
Constant current test

*1. LCD Display Type	TFT/Transmissive/Normally Black/262K Color		
*2. Viewing Diretion	Full View(U80/D80/R80/L80)	*6. Storage Temp	-30° C~80° C
*3. Interface	Uart(TTL)	*7. Driver IC	T.B.D
*4. Operating Voltage	VCI=5.0V (Typ)	*8. Backlight	WHITE Vf=3.2V If=80mA(Typ)
*5. Operating Temp	-20° C~70° C	*9. LCM Brightness	600 cd/m² (Typ)

HOTHMI

Dwg Title: LCM-EU02401KL		
Scale 1:1	Unit MM	Tol. ±0.3
Ver: V0	Drawn: Liu.YL	Date: 2020-04-26

3. Pin Description

3.1 Pin Description

Pin NO.	Symbol	Description
1	DC5V	Power Supply
2	TXD	Module UART data transmission port
3	RXD	Module UART data receiving port
4	GND	Ground
5	BUSY	This pin is the busy status output of the serial screen, and the host can know whether the serial screen is busy from this signal. Before updating the LCM SPI Flash data or MCU program, this pin needs to be pulled low to enter the "USB_Update" mode. Then connect the board to the computer with a USB cable
- END -		

3.2 USB Pin Description

Pin NO.	Symbol	Description
1	DC5V	Power Supply
2	USBDM	The signal of the USB data port DM. The internal MCU program and the data in the external SPI Flash can be updated through the USB interface
3	USBDP	The signal of the USB data port DP. The internal MCU program and the data in the external SPI Flash can be updated through the USB interface
4	NC	Open this pin
5	GND	Ground
- END -		

4. Electrical Characteristics

4-1 TFT LCD Module Operating Conditions

Item	Condition	Min	Type	Max	Uint
Power supply	-	4.50	5.00	5.50	V
electrostatic discharge voltage	contact	-	-	8	KV
	air	-	-	8	KV
Working current	BL=600cd/㎡	-	170	-	mA
Sleep Mode	-	-	-	-	mA
UART Baudrate	-	-	115200	-	bps
Recommended Power supply: 5V 1A DC power supply					

4-2 LED back light specification (pera chip)

Item	Symbol	Condition	Min	Type	Max	Uint
Forward voltage	Vt	If=20mA /1-chip	2.8	3.2	3.4	V
Forward current	Ipn		-	80	-	mA
Luminance (With LCD)	Lv	If=80mA	-	600	-	cd/㎡
Luminous color	White					

5. Serial Uart TFT Panel Software and Hardware Architecture

TFT panel mainly depends on communicate via RS232, RS485 to the Host. If the distance between the host and TFT panel is close(within 30cm), user can directly connect the UART port of the host to TFT panel UART port, see as below figure:

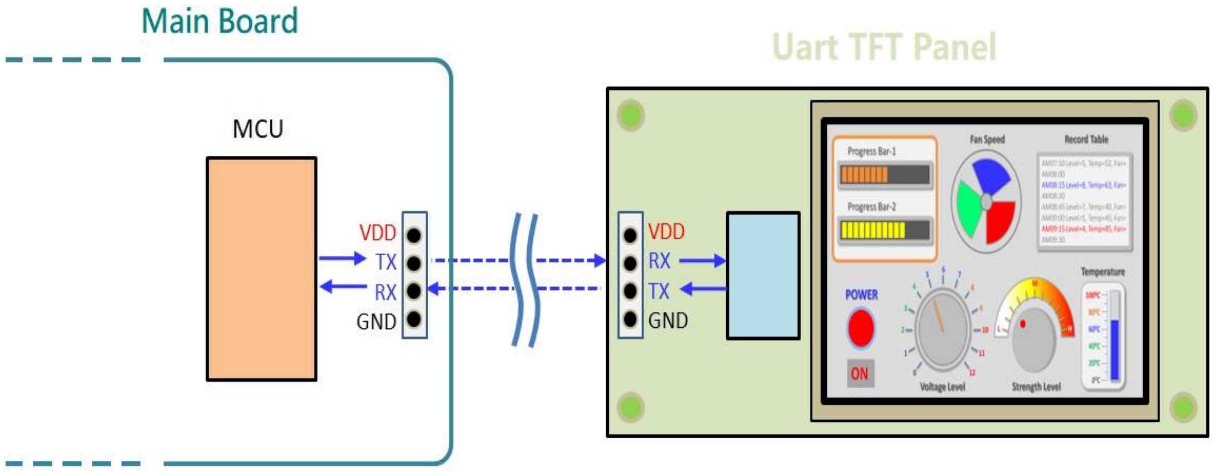


Figure 1-1: MCU Uart Connect with TFT Panel Uart

Before using the serial port screen, you must use the host computer software for development. We provide graphic and text integration compiler (UartTFT_Tool.exe) and graphical user interface editor (UI_Editor.exe) PC software to set TFT display and develop display functions. The software development of the host computer generates Bin files from the used pictures, text, GIF animation and other information. Developers can burn the Bin file to SPI Flash via USB, use the LT_VCOM_GUI_2.2.exe program, and then transfer the Uart (RS232) control line via USB to simulate the TFT serial screen, that is, pre-verify the TFT screen display screen.

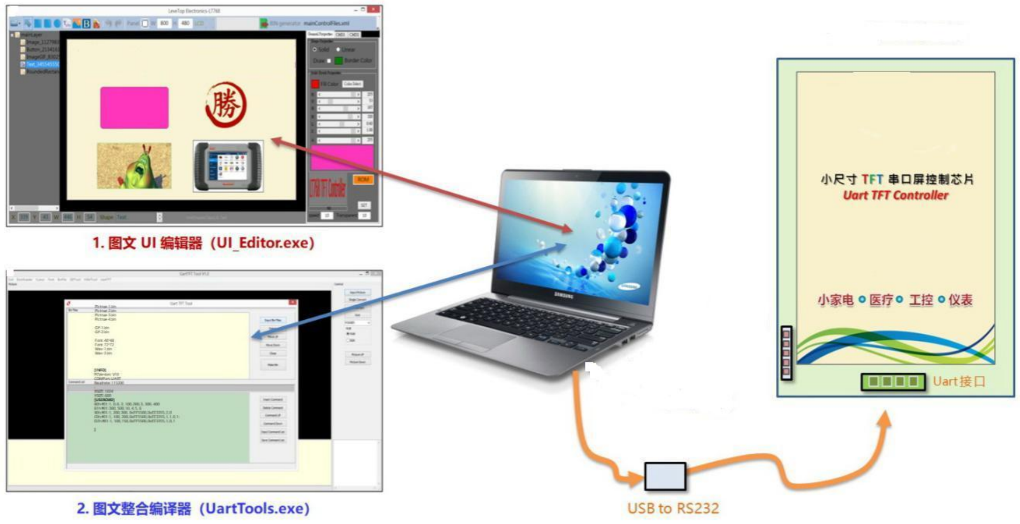

1. 图文 UI 编辑器（UI_Editor.exe）
2. 图文整合编译器（UartTools.exe）
USB to RS232
小尺寸 TFT 串口屏控制芯片
Uart TFT Controller
小家电 ○ 医疗 ○ 工控 ○ 仪表
Uart接口

Figure 1-2: Develop Diagram using PC Software

The host computer software will generate the command format according to the order and manner in which the pictures appear. The above-mentioned simulation is to replace the Host through the USB to Uart control line to send commands, so that the developer can perform pre-verification on the PC software. The commands sent by the host computer software can be displayed on the TFT panel and achieve the desired effect of the developer. Figure 1-3 shows the connection between the main control board and the TFT panel:

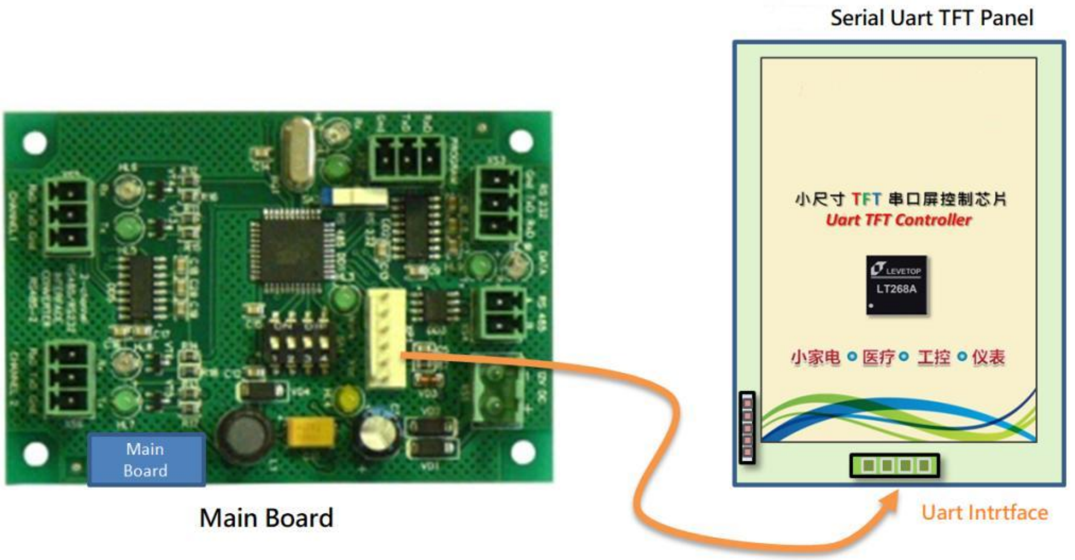


Figure 1-3: Host Mainboard Connection with TFT Panel

The UI_Editor has a fixed command for each display action of serial TFT panel, for example, 80h is the command to display pictures. UI_Editor will number the used pictures, and create Bin file for all pictures, texts, animation after compilation. After developers Programming the Bin file into SPI Flash,when PC send 80h, 00h, then will display on serial TFT panel will display the first picture ; when PC send 80h, 01h, the second picture will be displayed. Once commands sent by UI_Editor can be displayed and achieve desired effect, now we can connect Host with serial TFT panel as photo 1-5 shown. When Host MCU send out command 80h, 00h, 1Bh(CRC1), 98h(CRC2), Serial Uart TFT Panel will display the first picture and send back 80h, 00h(command executed), 1Bh(CRC1), 98h(CRC2) to Host to confirm the protocol is completed. Shown as below:

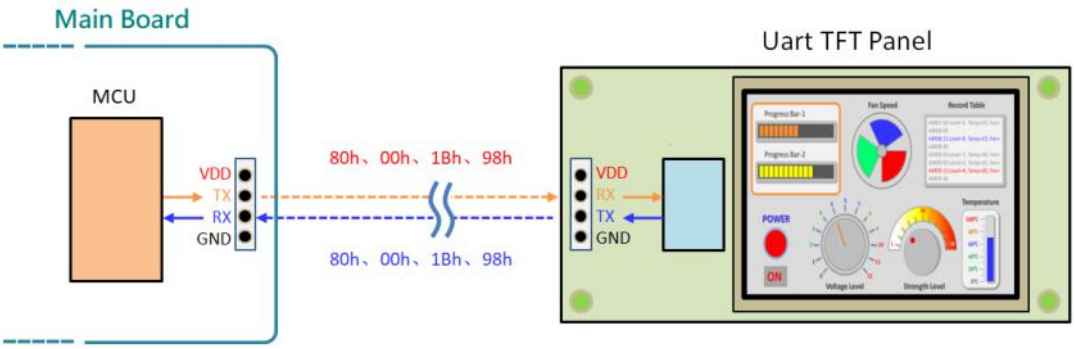


Figure 1-4: Command Protocol Example 1

When Host MCU send out command 80h, 01h, 0Bh(CRC1), B9h(CRC2), Serial Uart TFT Panel will display the second picture and send back 80h, 00h (command executed), 1Bh(CRC1) 98h(CRC2) to Host to confirm protocol is completed. As below figure shows:

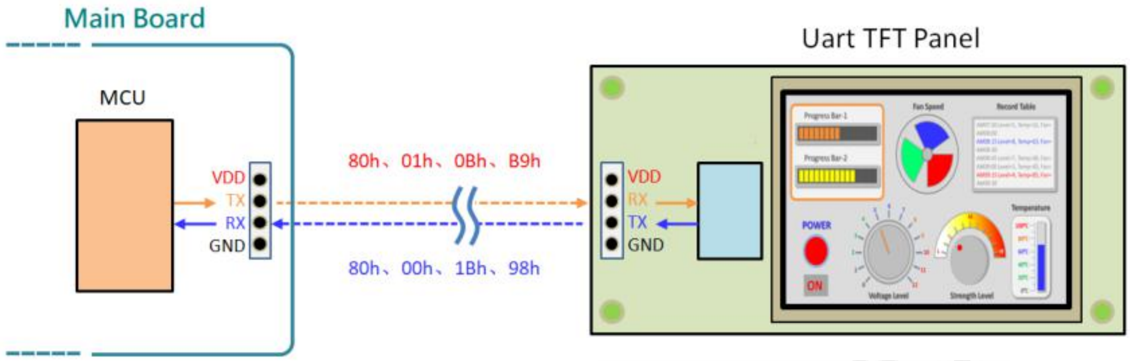


Figure 1-5: Command Protocol Example 2

In order to ensure the correct data transmission between Host and Serial Uart TFT Panel, the command sent by the Host MCU program must add a 1-byte initial code (fixed to 0xAA), a 2-byte CRC code, and a 4-byte end code (fixed to 0xE4, 0x1B, 0x11, 0xEE). After receiving the information or executing the command from the TFT panel, it returns the information to the host MCU. Please refer to section 6.2 for the protocol of command Host and TFT panel. To generate a 2-byte CRC, please refer to section 6.3.

The TFT panel also supports USB interface. Users can use the USB interface to update or SPI Flash program, please refer to Chapter 7.

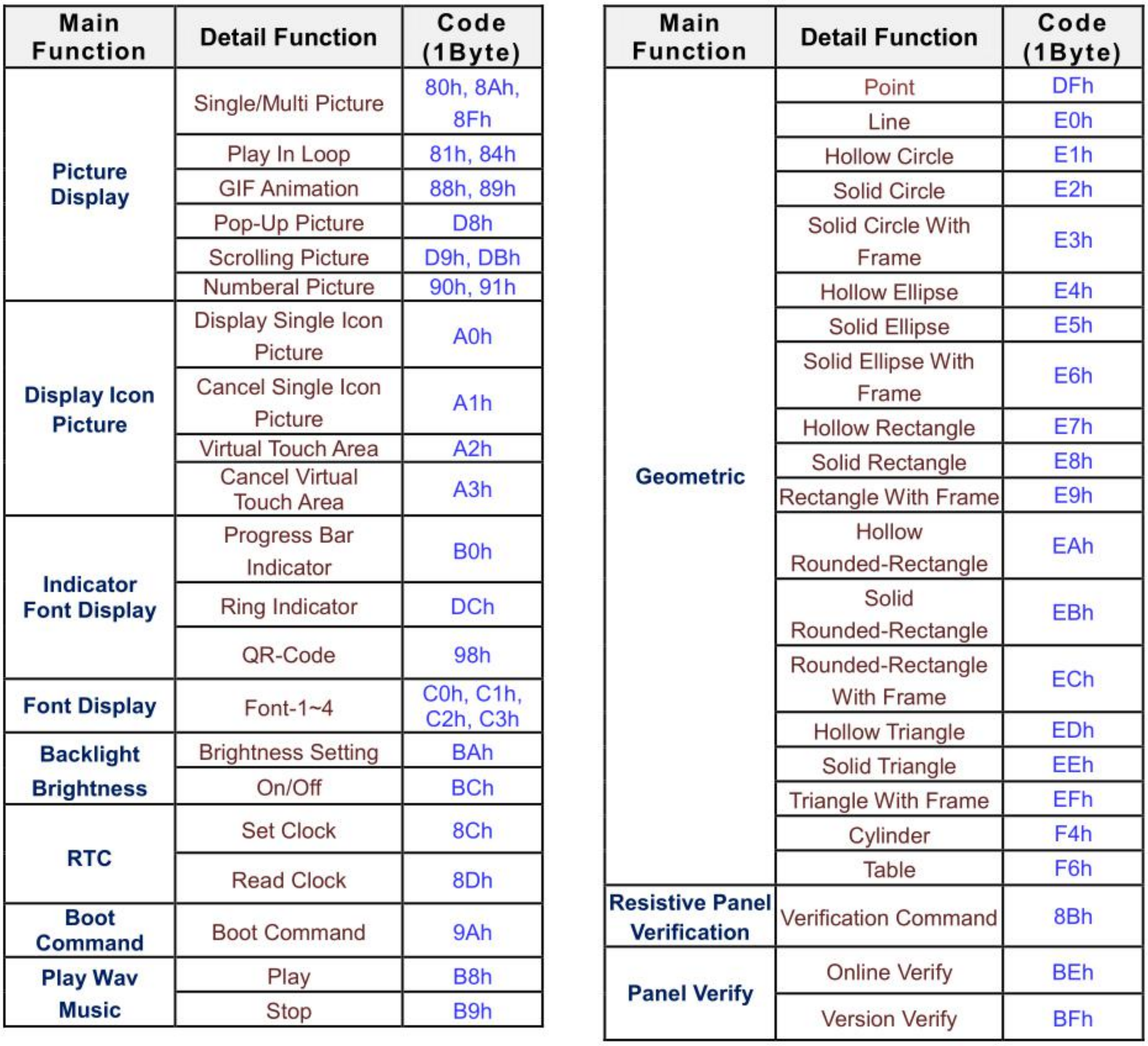
6. Serial Command

In order to allow the host to conveniently display pictures or information on the TFT panel through serial ports such as UART and SPI, a serial port command set is planned on it. Through the defined command code and parameter changes, the user can change the screen on the TFT panel. Our company provides two serial port panel development tools: UartTFT_Tool and UI_Editor. Users can choose one of these tools to develop TFT serial panels. And these two tools can simulate the display effect of TFT panel in real time and do preliminary verification.

6-1 Command Lists

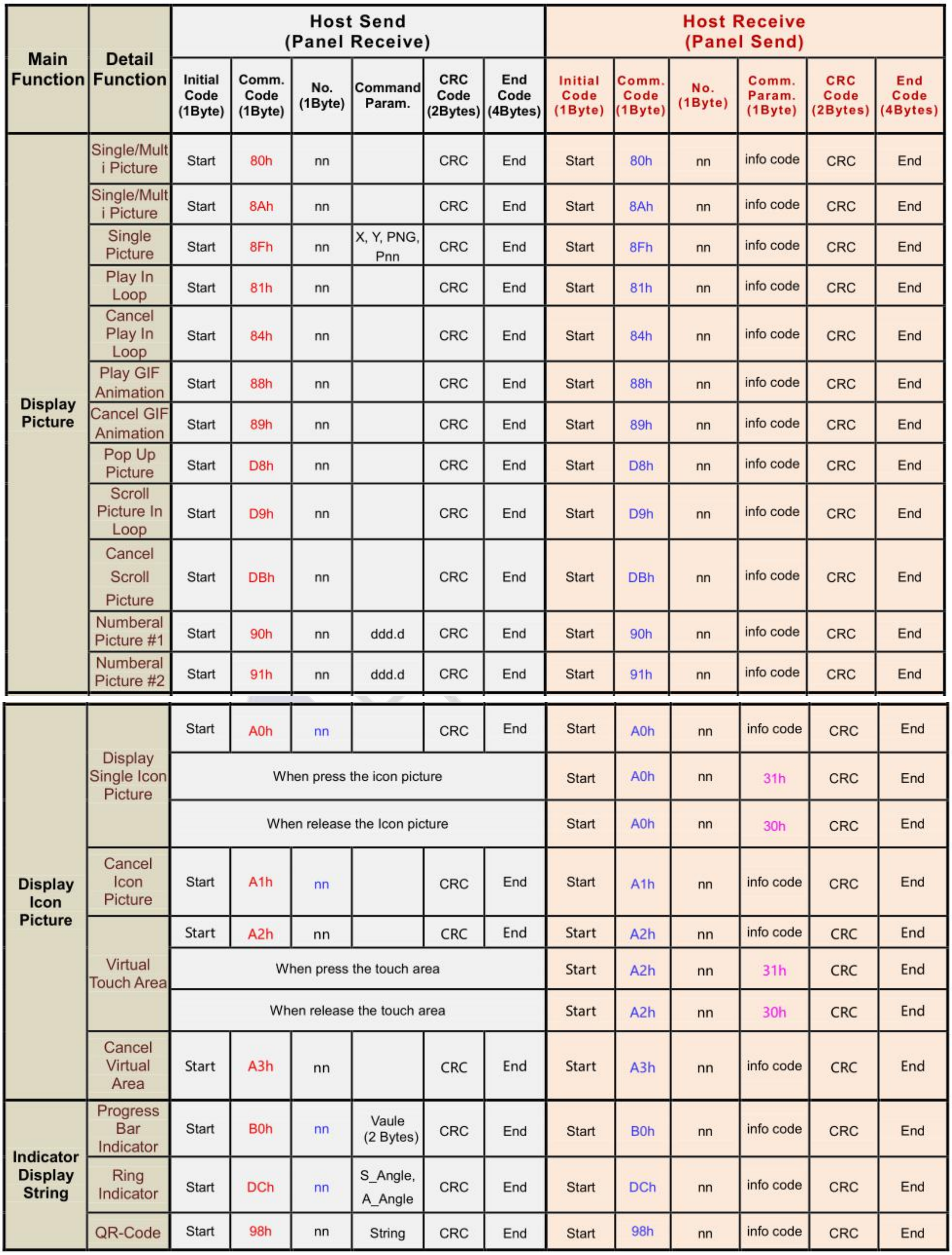
The commands supported by the TFT serial screen include static display of pictures, dynamic display of pictures, text display, geometric figures, etc., as shown in the following table 6-1

Table 6-1: Command List

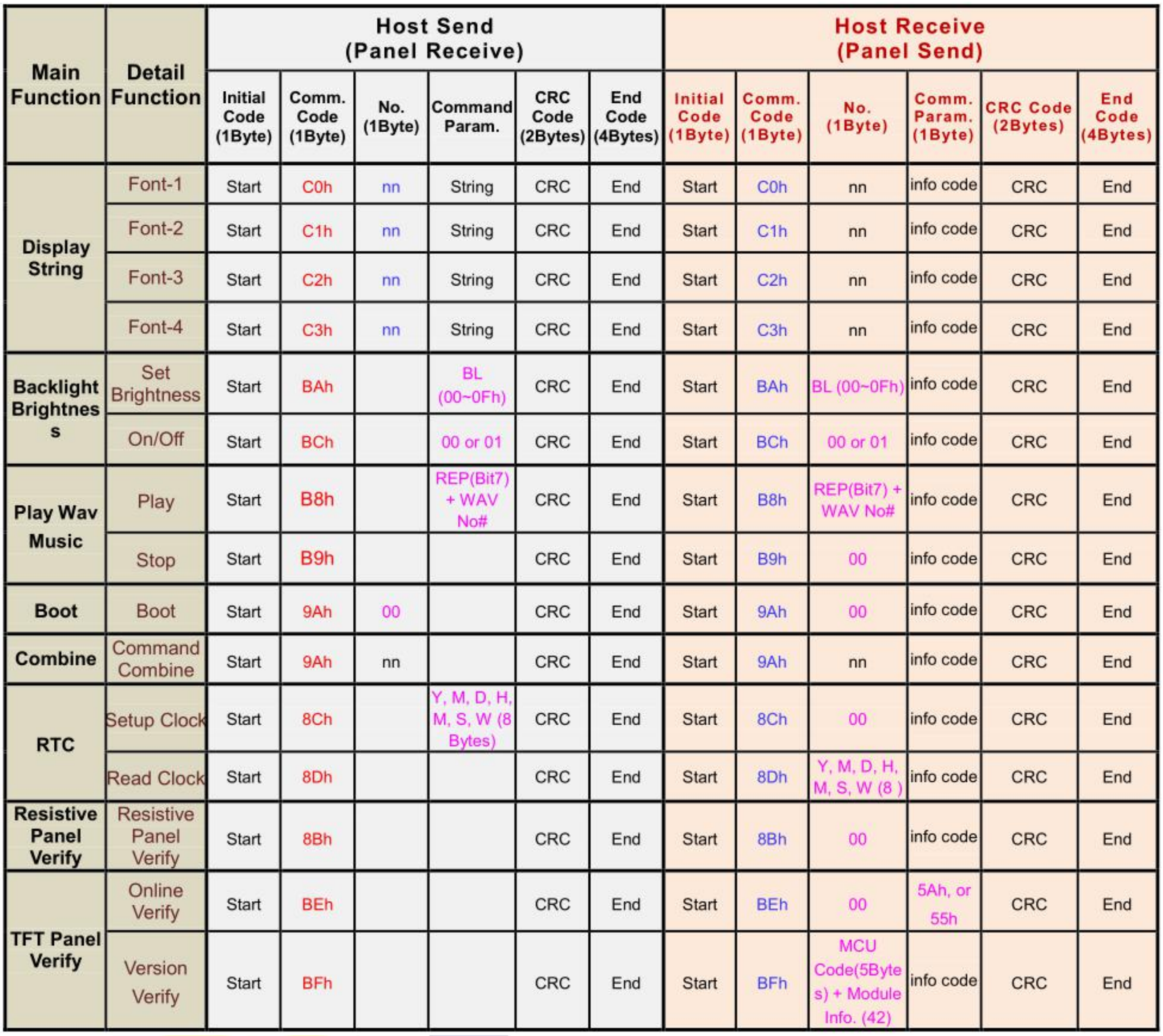


Main Function	Detail Function	Code (1Byte)
Picture Display	Single/Multi Picture	80h, 8Ah, 8Fh
	Play In Loop	81h, 84h
	GIF Animation	88h, 89h
	Pop-Up Picture	D8h
	Scrolling Picture	D9h, DBh
	Numberal Picture	90h, 91h
Display Icon Picture	Display Single Icon Picture	A0h
	Cancel Single Icon Picture	A1h
	Virtual Touch Area	A2h
	Cancel Virtual Touch Area	A3h
Indicator Font Display	Progress Bar Indicator	B0h
	Ring Indicator	DCh
	QR-Code	98h
Font Display	Font-1~4	C0h, C1h, C2h, C3h
Backlight Brightness	Brightness Setting	BAh
	On/Off	BCh
RTC	Set Clock	8Ch
	Read Clock	8Dh
Boot Command	Boot Command	9Ah
Play Wav Music	Play	B8h
	Stop	B9h
Geometric	Point	DFh
	Line	E0h
	Hollow Circle	E1h
	Solid Circle	E2h
	Solid Circle With Frame	E3h
	Hollow Ellipse	E4h
	Solid Ellipse	E5h
	Solid Ellipse With Frame	E6h
	Hollow Rectangle	E7h
	Solid Rectangle	E8h
	Rectangle With Frame	E9h
	Hollow Rounded-Rectangle	EAh
	Solid Rounded-Rectangle	EBh
	Rounded-Rectangle With Frame	ECh
	Hollow Triangle	EDh
	Solid Triangle	EEh
	Triangle With Frame	EFh
	Cylinder	F4h
	Table	F6h
Resistive Panel Verification	Verification Command	8Bh
Panel Verify	Online Verify	BEh
	Version Verify	BFh

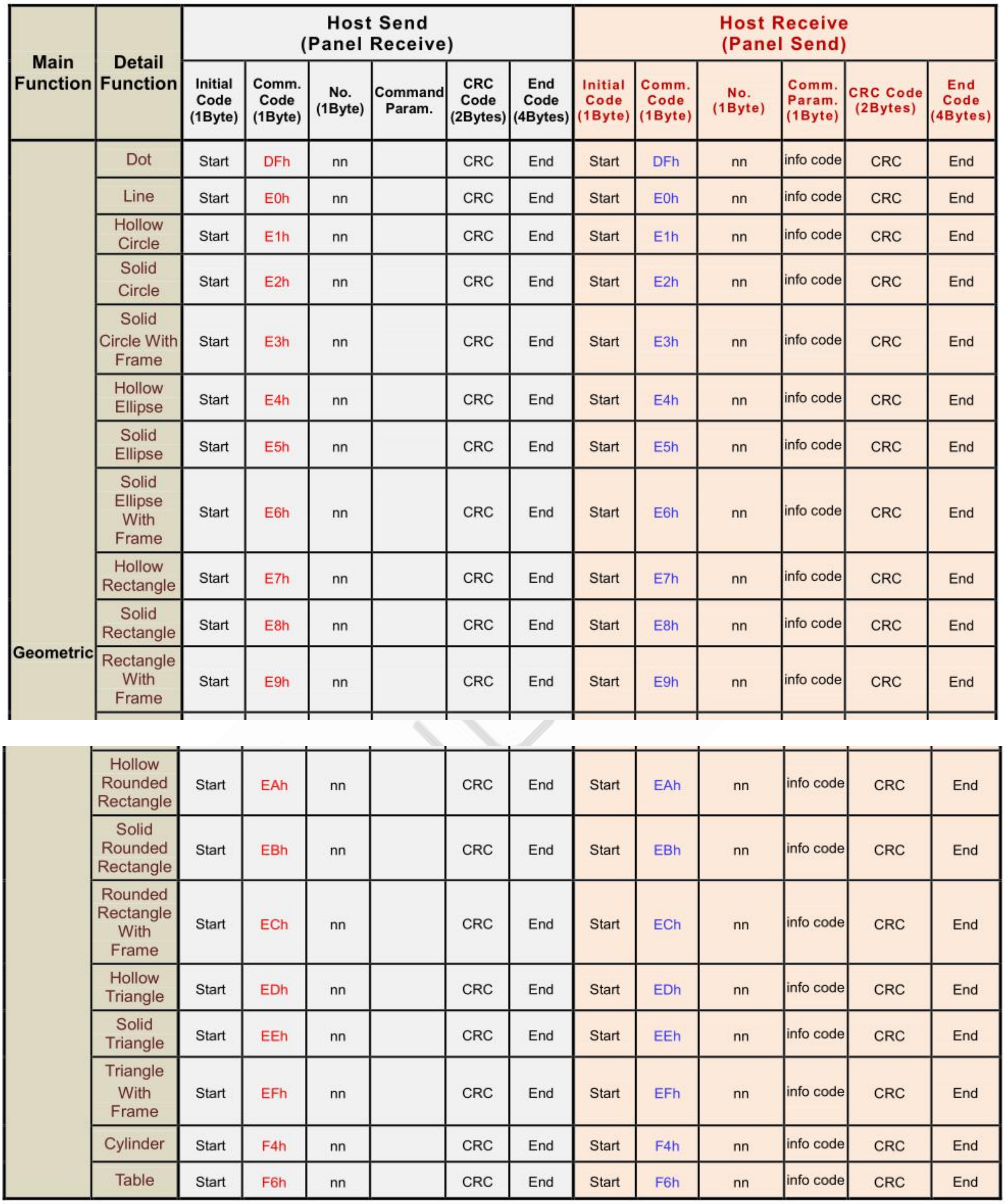
6-2 Host and Serial Uart Panel Protocol



Main Function	Detail Function	Host Send (Panel Receive)						Host Receive (Panel Send)					
		Initial Code (1Byte)	Comm. Code (1Byte)	No. (1Byte)	Command Param.	CRC Code (2Bytes)	End Code (4Bytes)	Initial Code (1Byte)	Comm. Code (1Byte)	No. (1Byte)	Comm. Param. (1Byte)	CRC Code (2Bytes)	End Code (4Bytes)
Display Picture	Single/Multi Picture	Start	80h	nn		CRC	End	Start	80h	nn	info code	CRC	End
	Single/Multi Picture	Start	8Ah	nn		CRC	End	Start	8Ah	nn	info code	CRC	End
	Single Picture	Start	8Fh	nn	X, Y, PNG, Pnn	CRC	End	Start	8Fh	nn	info code	CRC	End
	Play In Loop	Start	81h	nn		CRC	End	Start	81h	nn	info code	CRC	End
	Cancel Play In Loop	Start	84h	nn		CRC	End	Start	84h	nn	info code	CRC	End
	Play GIF Animation	Start	88h	nn		CRC	End	Start	88h	nn	info code	CRC	End
	Cancel GIF Animation	Start	89h	nn		CRC	End	Start	89h	nn	info code	CRC	End
	Pop Up Picture	Start	D8h	nn		CRC	End	Start	D8h	nn	info code	CRC	End
	Scroll Picture In Loop	Start	D9h	nn		CRC	End	Start	D9h	nn	info code	CRC	End
	Cancel Scroll Picture	Start	DBh	nn		CRC	End	Start	DBh	nn	info code	CRC	End
	Numberal Picture #1	Start	90h	nn	ddd.d	CRC	End	Start	90h	nn	info code	CRC	End
	Numberal Picture #2	Start	91h	nn	ddd.d	CRC	End	Start	91h	nn	info code	CRC	End
Display Icon Picture	Display Single Icon Picture	Start	A0h	nn		CRC	End	Start	A0h	nn	info code	CRC	End
		When press the icon picture						Start	A0h	nn	31h	CRC	End
		When release the Icon picture						Start	A0h	nn	30h	CRC	End
	Cancel Icon Picture	Start	A1h	nn		CRC	End	Start	A1h	nn	info code	CRC	End
	Virtual Touch Area	Start	A2h	nn		CRC	End	Start	A2h	nn	info code	CRC	End
		When press the touch area						Start	A2h	nn	31h	CRC	End
		When release the touch area						Start	A2h	nn	30h	CRC	End
	Cancel Virtual Area	Start	A3h	nn		CRC	End	Start	A3h	nn	info code	CRC	End
Indicator Display String	Progress Bar Indicator	Start	B0h	nn	Vaule (2 Bytes)	CRC	End	Start	B0h	nn	info code	CRC	End
	Ring Indicator	Start	DCh	nn	S_Angle, A_Angle	CRC	End	Start	DCh	nn	info code	CRC	End
	QR-Code	Start	98h	nn	String	CRC	End	Start	98h	nn	info code	CRC	End



Main Function	Detail Function	Host Send (Panel Receive)						Host Receive (Panel Send)					
		Initial Code (1Byte)	Comm. Code (1Byte)	No. (1Byte)	Command Param.	CRC Code (2Bytes)	End Code (4Bytes)	Initial Code (1Byte)	Comm. Code (1Byte)	No. (1Byte)	Comm. Param. (1Byte)	CRC Code (2Bytes)	End Code (4Bytes)
Display String	Font-1	Start	C0h	nn	String	CRC	End	Start	C0h	nn	info code	CRC	End
	Font-2	Start	C1h	nn	String	CRC	End	Start	C1h	nn	info code	CRC	End
	Font-3	Start	C2h	nn	String	CRC	End	Start	C2h	nn	info code	CRC	End
	Font-4	Start	C3h	nn	String	CRC	End	Start	C3h	nn	info code	CRC	End
Backlight Brightness s	Set Brightness	Start	BAh		BL (00~0Fh)	CRC	End	Start	BAh	BL (00~0Fh)	info code	CRC	End
	On/Off	Start	BCh		00 or 01	CRC	End	Start	BCh	00 or 01	info code	CRC	End
Play Wav Music	Play	Start	B8h		REP(Bit7) + WAV No#	CRC	End	Start	B8h	REP(Bit7) + WAV No#	info code	CRC	End
	Stop	Start	B9h			CRC	End	Start	B9h	00	info code	CRC	End
Boot	Boot	Start	9Ah	00		CRC	End	Start	9Ah	00	info code	CRC	End
Combine	Command Combine	Start	9Ah	nn		CRC	End	Start	9Ah	nn	info code	CRC	End
RTC	Setup Clock	Start	8Ch		Y, M, D, H, M, S, W (8 Bytes)	CRC	End	Start	8Ch	00	info code	CRC	End
	Read Clock	Start	8Dh			CRC	End	Start	8Dh	Y, M, D, H, M, S, W (8)	info code	CRC	End
Resistive Panel Verify	Resistive Panel Verify	Start	8Bh			CRC	End	Start	8Bh	00	info code	CRC	End
TFT Panel Verify	Online Verify	Start	BEh			CRC	End	Start	BEh	00	5Ah, or 55h	CRC	End
	Version Verify	Start	BFh			CRC	End	Start	BFh	MCU Code(5Bytes) + Module Info. (42)	info code	CRC	End



Main Function	Detail Function	Host Send (Panel Receive)						Host Receive (Panel Send)					
		Initial Code (1Byte)	Comm. Code (1Byte)	No. (1Byte)	Command Param.	CRC Code (2Bytes)	End Code (4Bytes)	Initial Code (1Byte)	Comm. Code (1Byte)	No. (1Byte)	Comm. Param. (1Byte)	CRC Code (2Bytes)	End Code (4Bytes)
Geometric	Dot	Start	DFh	nn		CRC	End	Start	DFh	nn	info code	CRC	End
	Line	Start	E0h	nn		CRC	End	Start	E0h	nn	info code	CRC	End
	Hollow Circle	Start	E1h	nn		CRC	End	Start	E1h	nn	info code	CRC	End
	Solid Circle	Start	E2h	nn		CRC	End	Start	E2h	nn	info code	CRC	End
	Solid Circle With Frame	Start	E3h	nn		CRC	End	Start	E3h	nn	info code	CRC	End
	Hollow Ellipse	Start	E4h	nn		CRC	End	Start	E4h	nn	info code	CRC	End
	Solid Ellipse	Start	E5h	nn		CRC	End	Start	E5h	nn	info code	CRC	End
	Solid Ellipse With Frame	Start	E6h	nn		CRC	End	Start	E6h	nn	info code	CRC	End
	Hollow Rectangle	Start	E7h	nn		CRC	End	Start	E7h	nn	info code	CRC	End
	Solid Rectangle	Start	E8h	nn		CRC	End	Start	E8h	nn	info code	CRC	End
	Rectangle With Frame	Start	E9h	nn		CRC	End	Start	E9h	nn	info code	CRC	End
	Hollow Rounded Rectangle	Start	EAh	nn		CRC	End	Start	EAh	nn	info code	CRC	End
	Solid Rounded Rectangle	Start	EBh	nn		CRC	End	Start	EBh	nn	info code	CRC	End
	Rounded Rectangle With Frame	Start	ECh	nn		CRC	End	Start	ECh	nn	info code	CRC	End
	Hollow Triangle	Start	EDh	nn		CRC	End	Start	EDh	nn	info code	CRC	End
	Solid Triangle	Start	EEh	nn		CRC	End	Start	EEh	nn	info code	CRC	End
	Triangle With Frame	Start	EFh	nn		CRC	End	Start	EFh	nn	info code	CRC	End
	Cylinder	Start	F4h	nn		CRC	End	Start	F4h	nn	info code	CRC	End
	Table	Start	F6h	nn		CRC	End	Start	F6h	nn	info code	CRC	End

6-3 RS-232(UART) Communication Protocol

When the main system transmits a display command to the TFT panel through the UART serial port, it includes Command Code, Code Number, and Command Parameters, 1 Byte Start Code (fixed to 0xAA) and 2 Bytes, CRC Code, 4 Byte End Code (fixed to 0xE4, 0x1B, 0x11, 0xEE), the command information is as below:

Initial Code	Command Code	Serial Code No.	Parameter	CRC Code	End Code
0xAA (1 Byte)	1 Byte	1 Byte	n Bytes	2 Bytes	**0xE4, 0x1B, 0x11, 0xEE** (4 Bytes)

CRC protocol as below:

```
chkSum = Rx_CRC_CCITT(txBuf,txLen);
txBuf[txLen++] = (chkSum>>8)&0xFF;
txBuf[txLen++] = chkSum&0xFF;

unsigned int Rx_CRC_CCITT(unsigned char *puchMsg, unsigned int usDataLen)
{
unsigned char i = 0;
unsigned short wCRCin = 0x0000;
unsigned short wCPoly = 0x1021;
unsigned char wChar = 0;
while (usDataLen--)
    {
    wChar = *(puchMsg++);
    wCRCin ^= (wChar << 8);
    for(i = 0; i < 8; i++)
        {
        if (wCRCin & 0x8000)
           wCRCin = (wCRCin << 1) ^ wCPoly;
        else
            wCRCin = wCRCin << 1;
        }
    }
return (wCRCin);
}
```

After receiving the host command, the TFT panel will normally respond to 10 bytes of information, including initial code, command code, information code, CRC code, and end code. The first Byte is the initial code, then the received command, the third is the serial number. Then the fourth is information code that returns the executed result. The fifth and sixth are the CRC code, and the last is the end code of 4 Bytes:

Initial Code	Command Code	Serial Number	Information Code	CRC Code	End Code
0xAA (1 Byte)	1 Byte	Normal Command (1 Byte) BFh Command (47 Bytes)	1Byte 0x00: Command executed 0x01: Command parameter error 0x02: Non-exist command 0x03: Command's configuration Overflow 0x04: CRC correction error 0x05: Flash Data Exception BEh Command: 0x5A: Ready 0x55: Not Ready A0h Command: 0x31: Press Icon 0x30: Release Icon	2 Bytes	**0xE4, 0x1B, 0x11, 0XEE** (4 Bytes)

In the information structure of Serial Uart TFT Panel feedback, serial numbers also represent different meanings in some instructions. For example, the brightness command - BAh its serial number represents the backlight brightness. The Version Check command - BFh has 47 Bytes representing Serial Uart TFT Panel information.

6-4 A Program Example of Host Sends Command

The following is an example of a program in which the host passes a display command to the serial TFTpanel through the UART port. This program takes the transmission of the first picture (80h, 00h) as an example. The program will automatically add 0xAA start code, 2 Byte CRC codes and 4Byte end codes.

```
int main (void)
{
char c[] = "80 00"; //Send 80h command and 00 serial number
Send(c);
while(1);
}

unsigned short Rx_CRC_CCITT(unsigned char *puchMsg, unsigned int usDataLen) // generate CRC
{
unsigned char i = 0;
unsigned short wCRCin = 0x0000;
unsigned short wCPoly = 0x1021;
unsigned char wChar = 0;
while (usDataLen--)
    {
     wChar = *(puchMsg++);
     wCRCin ^= (wChar << 8);
     for(i = 0; i < 8; i++)
     {
         if (wCRCin & 0x8000)
         wCRCin = (wCRCin << 1) ^ wCPoly;
     else
         wCRCin = wCRCin << 1;
     }
    }
return (wCRCin);
}
```

```
    void Send(char *c) //Send command function
    {
        unsigned char Sendbuff[100]={0};
        unsigned short Send_CRC = 0;
        unsigned char C_flag = 0;
        int i = 0, j = 0;
    if(((c[0]>=0x30 && c[0]<=0x39) || (c[0]>=0x41 && c[0]<=0x5A)) || ((c[1]>=0x30
&& c[1]<=0x39) || (c[1]>=0x41 && c[1]<=0x5A)))
        {
        while(c[i] != '\0')
            {
            if(c[i] != ' ')
                {

                if(c[i] == '"') ，

                    {
                    C_flag++;
                    i++;
                    }
                if(C_flag == 1)
                    {
                    if(c[i] != '"')
                        {
                        Sendbuff[j] = c[i]; //ASCII direct output
                        i++;
                        j++;
                        }
                    }
                else if(C_flag == 2)
                    {
                    C_flag = 0;
                    i++;
                    }
                if(C_flag == 0)
                    {
                    if(c[i] == '/') break;
                    if(c[i]>=0x30 && c[i]<=0x39) //0~9
                        {
                        Sendbuff[j] = ((c[i] - 0x30)<< 4 );
                        i++;
                    if(c[i]>=0x30 && c[i]<=0x39)
                        {
                        Sendbuff[j] += (c[i] - 0x30);
                        i++;
                        j++;
                        }
                    else if(c[i]>=0x41 && c[i]<=0x5A)
                        {
                        Sendbuff[j] += (c[i] - 0x37);
                        i++;
                        j++;
                        }
                    }
            else if(c[i]>=0x41 && c[i]<=0x5A) //A~Z
                {
                Sendbuff[j] = ((c[i] - 0x37)<< 4 );
                i++;
                if(c[i]>=0x30 && c[i]<=0x39)
                    {
                    Sendbuff[j] += (c[i] - 0x30);
                    i++;
                    j++;
                    }
                else if(c[i]>=0x41 && c[i]<=0x5A)
                    {
                    Sendbuff[j] += (c[i] - 0x37);
                    i++;
                    j++;
                    }
                }
            }
        }
        else i++;
    }
Sendbuff[j] = '\0';
//  printf("%s\r\n",Sendbuff);
/************************ CRC, header and end ***********************/
Send_CRC = Rx_CRC_CCITT(Sendbuff,j);
Sendbuff[j] = Send_CRC>>8 & 0xff;
Sendbuff[j+1] = Send_CRC & 0xff;
```

```
for(i = 0;i<j+2;i++)
Sendbuff[j+2-i]=Sendbuff[j+1-i];
Sendbuff[0] = 0xAA;
Sendbuff[j+3] = 0xE4;
Sendbuff[j+4] = 0x1B;
Sendbuff[j+5] = 0x11;
Sendbuff[j+6] = 0xEE;
j+=7;
for(i = 0; i < j; i++)
{
USART_DATA(USART0) = (uint8_t) Sendbuff[i];
while(usart_flag_get(USART0, USART_FLAG_TBE)==0){};
}
}
}
```

7、Production name

液晶模组		简易串口 Easy uart	尺寸	款号	TP 类型	功能类型	版本号
LCM	-	EU	024	01	K	L	V0
					K:机械按键	L:基本型（不带触摸等附加功能）	
					R:RTP	H: 增强型（带触摸等附加功能）	
					C:CTP		
					N:无触无按键		

- - END - -