HM043WQ111ST-UR

4.3" Color TFT-LCD

FUNCTIONAL DRAFT SPECIFICATION

(This document is meant for customers' approval)

Release Date
30th Oct 2020

HIGHNESS MICROELECTRONICS PVT. LTD

URL: www.highnessmicro.com, Email:sales@highnessmicro.com

1 Introduction

1.1 Brief Introduction

The HM043WQ111ST-UR has been conceived as **TFT monitor** & **Touch controller.** It includes processor, control program, driver, flash memory,RS232/RS485 port, touch screen, power supply etc., so it is a whole display system based on the powerful & easy Command Set operating system, which can be controlled by Any MCU.

The HM043WQ111ST-UR can be used to perform all basic functions, such as text display, image display, curve display as well as touch function etc. The User Interface can be more abundant and various. And the flash memory can store your data, configuration files and images etc.

1.2 Warranty

All products purchased from our company are guaranteed to keep in good repair for 3 year
s. If quality problems (except human error) happen in guarantee
period, our company will maintain for free or replace the broken one
unconditionally.

1.3 Product Characteristics

- With Cortex CPU & Driving device
- Controlled by any MCU
- Display Picture/ Text /Curve
- 65536 colour TFT display
- With/without Touch Screen
- RS232/ RS485/ TTL UART Interface & USB port
- Wide voltage range
- Easy to use! Powerful function! Saving cost and time!

1.4 Application Area

Widely used in various industrial field
- Medical & Beauty Equipment
- Engineering Machinery and Vehicle Equipment
- Electronic Instrument
- Industrial Control System
- Electric Power Industry
- Civil Electronic Equipment
- Automation Equipment
- Traffic

Etc.

1.5 Working Principle

The Intelligent TFT-LCD Module communicates with the Customer's MCU via Command Set (HEX Code), and then the MCU would control its connected equipment to work according to the received commands.

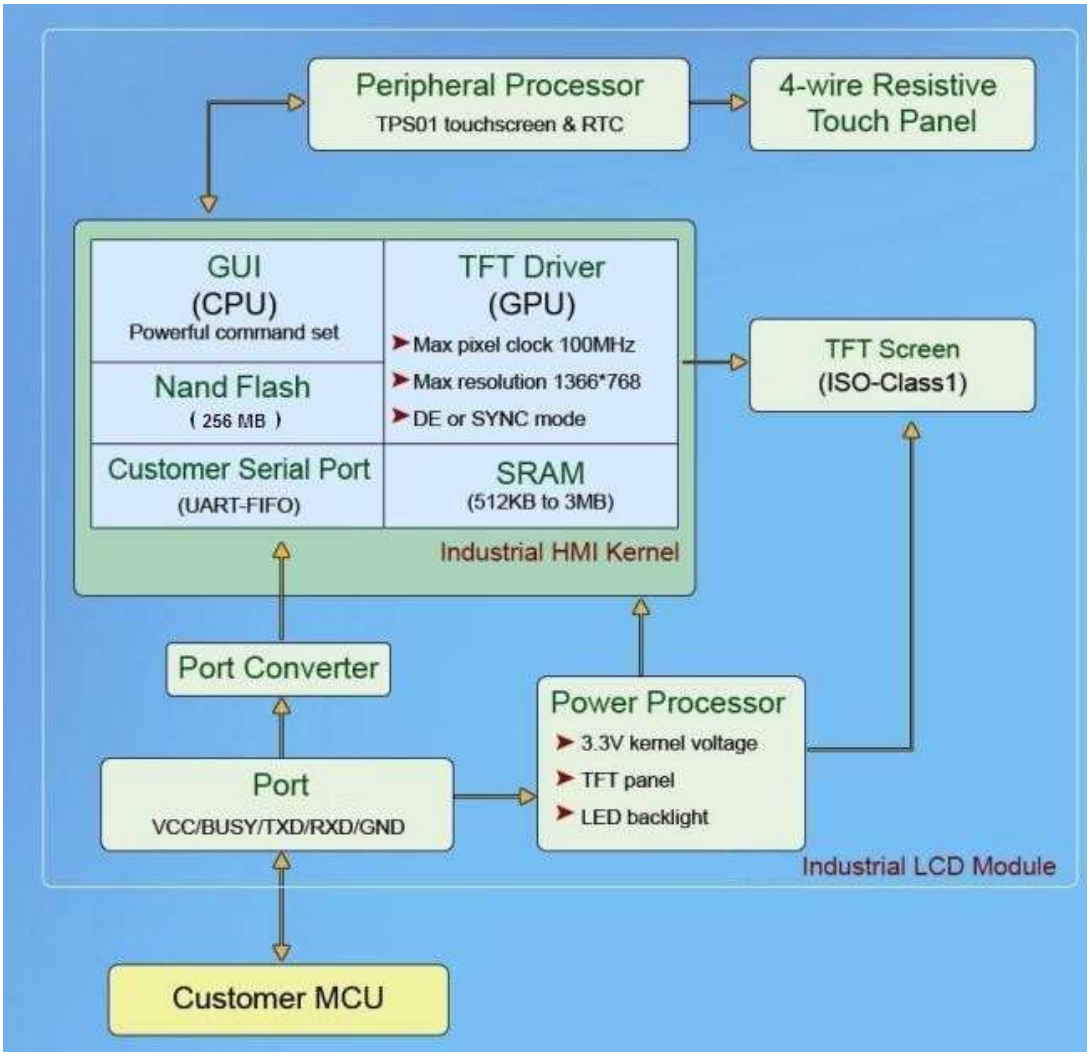


Figure 1.1-1 Configuration and process control phases

1.6 Operation Processing

Only **3 steps** to use our TFT-LCD Module:

1. Design a group of Beautiful “Graphical User Interface”. (Ref. Picture 1.2-1)
2. Connect with customer’s MCU through RS232, RS485 or TTL level directly. Plug and play.
3. Write a simple program for MCU to control the TFT-LCD Module via powerful “Command Set” (HEX Code).

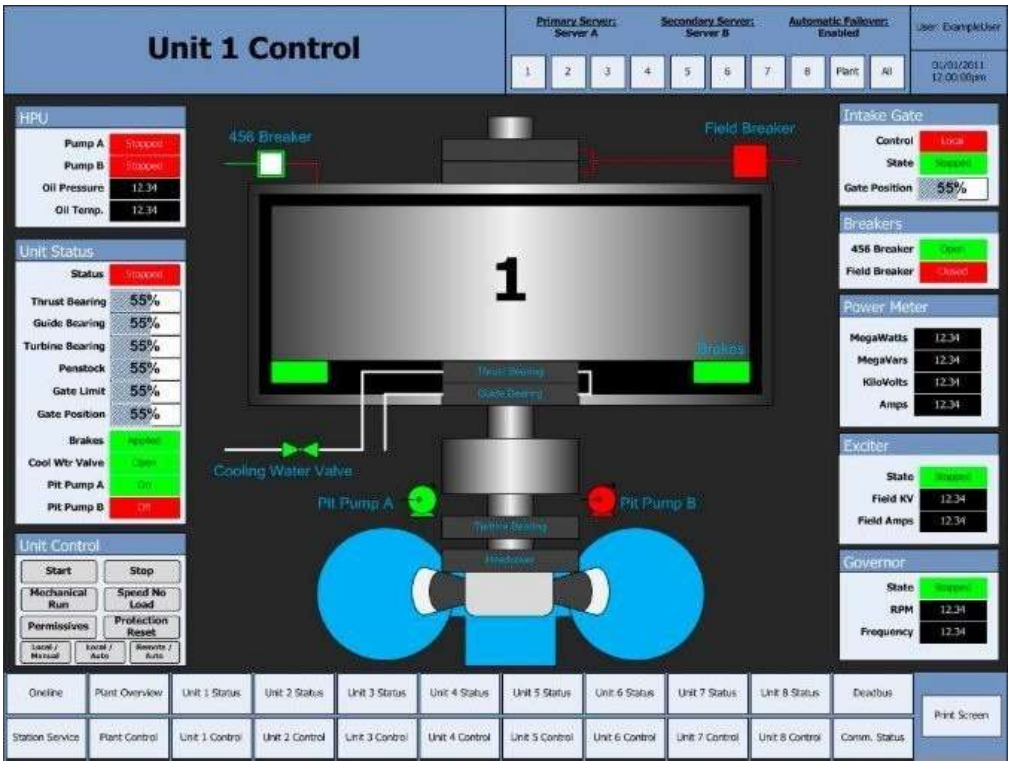
That’s all.

For example: Image Command 0x70.

Send: “0xAA + 0x70 + Pic_ID + 0xCC 0x33 0xC3 0x3C”

	Introduction	Example	Note
AA	**Frame head**		**Frame head of each Command**
0x70	**Command Code**		
Pic_ID	**Sequence number of the storage position.**	**0x00 0x01**	**Show Picture – 01**
CC 33 C3 3C	**Frame end**		**Frame end of each Command**

More information, please refer to the document of Command Set.



Picture 1.2-1

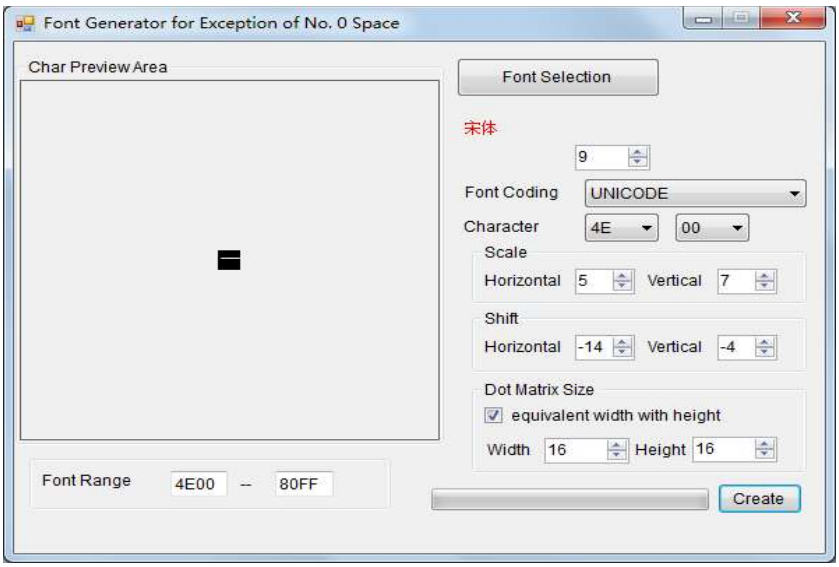
1.7 Software Operation

We will offer simple software “Toolbox” to help you to work with Intelligent TFT-LCD Module on computer.

1. Development Assistant: Show how to send Commands to control the TFT-LCD Module.
2. Download Assistant: Download the images and configuration files into TFT-LCD Module.
3. Update firmware: For updating firmware.
4. Touch control: Set many buttons on the User Interface, and create a touch configuration file.
5. Dial control: Create the dial configuration file.

Picture 1.3-1

6. Font software: Create font configuration file.



Picture 1.3-2

2 Technical Parameters

This chapter contains technical data on:

- **Physical Parameters:**
  Physical Parameters
  Display

- **Hardware Parameters:**
  Processor
  Memory
  Interface
  Power Supply

- **Storage & Test**
  Electrical Characteristics
  Ambient Conditions
  Noise Immunity
  Radio Interference

- **Support Device**
  Support Device

Physical Parameter	
Size	4.3 inch
Resolution	480×RGB×272
Pixel Spacing	0.1905 mm × 0.0635 mm (H×V)
Color	65536 colors (16 bit)
Viewing Area	95 mm × 53.9 mm
Display Dimension	97.1 mm × 56 mm
Overall Dimension	120.4 mm × 68.6 mm × 12.3 mm(N)/ 13.9 mm(T)
Net Weight	115 g(N)/ 155 g(T)

Display	
Backlight Type	LED
Brightness	800 cd/m^2 (Brightness can be adjustable in 100 levels)
Contrast	500:1
Backlight life	50,000 hours
Viewing Angle	70°/ 70°/ 50°/ 70°(L/ R/ U/ D)
TFT Panel	A+ Class Industry Panel
Touch Screen	Industry Level 4 wire resistance Or without touch screen is optional.
Screen Mode：	Digital

Processor	
CPU	CortexM3 STM32F103 70MHz
LCD Controller	CPLD EPM240
Refresh Rate	60Hz
Update Speed of per frame	19 ms/frame (53 images/s)

Memory	
Flash Memory	Standard 128MB, Extension 1GB
Memory Amount for picture	416 images, Extension 4000

Interface	
Interface	RS232/ RS485/ TTL UART Interface
Image downloading	MiniUSB2.0 (12Mbps) & TF card

Power Supply	
Rated voltage	+12 V DC
Permissible voltage range	+4.5...+20.0 V DC (or Extend to +42V)
Max. permissible transients	+24 V (500 msec)
Time between two transients	50 sec minimum
Fuse, internal	Electronic
Power consumption	2.0 W

Electrical Characteristics					
Parameter		**Condition**	**Min**	**Type**	**Max**
Supply Current		VIN=5V		400mA	
		VIN=12V		160mA	
		VIN=20V		100mA	
Signal Input Voltage	TTL level	VIH	2.1V		
		VIL			0.9V
	RS232 level	V range	-15V		+15V
	RS485 level	Differential Threshold	-0.2V		0.2V
Signal Output Voltage	TTL level	VOH	3V		3.3V
		VOL	0V		0.1V
	RS232 level	V range	-15V		+15V
	RS485 level	Differential Driver			5V
Baud Rate			1200 bps		921600 bps
Note: As the brightness lower, the current will also reduce.					

Ambient Conditions	
Max. permissible ambient temperature	
Operation	-30℃ ～ +80℃
Storage	-30℃ ～ +85℃
Relative humidity	
Operation	55℃, 85%
Storage	60℃, 90%
Shock loading	
Operation	15 g/ 11 msec
Storage	25 g/ 6 msec
Vibration	
Operation	0.035 mm (10 - 58 Hz)/ 1 g (58 - 500 Hz)
Storage	3.5 mm (5 - 8,5 Hz)/ 1 g (8.5 - 500 Hz)
Barometric pressure	
Operation	706 to 1030 hPa
Storage	581 to 1030 hPa

Noise Immunity	
Static discharge (contact discharge/air discharge)	EN 61000-4-2 6 kV/ 8 kV
RF irradiation	EN 61000-4-3 10 V/m, 80% AM 1 kHz
Pulse modulation	ENV 50204 900 MHz ±5 MHz 10 V/meff., 50% ED, 200 Hz
RF conduction	EN 61000-4-6 150 kHz - 80 MHz 10 V, 80% AM, 1 kHz
Burst interference Supply lines Process data lines Signal lines	EN 61000-4-4 2kV 2kV 1kV

Radio Interference	
Radio interference level complying to EN 55011	Class A

Support Device	
Buzzer	Support
RTC	Support
Key-board Interface	Support User can set dot matrix for 8*8 or 4*4
16 IO Ports for Users	Support
USB port For Downloading Pictures & Font File	Support (Download one 800×600 picture only take 0.5S）
Touch Screen	4 Wire Resistance/ Capacitive
Default Font	8×8 / 16×16/ 16×32/ 12×12 / 12×24 (Dot Matrix)
Picture	Support BMP, JPG Format
Storage Data	Support
Command Set	Unified Simplified Command Sets

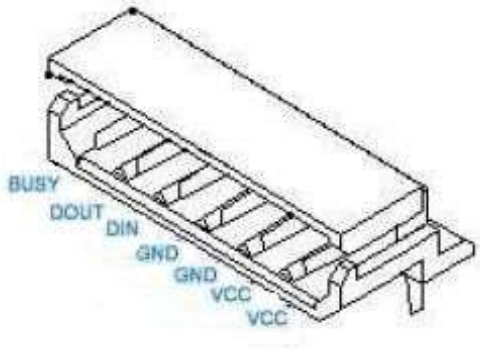
3 Interface Description

This chapter contains the description of the interfaces:

- VVC
- GND
- DIN
- DOUT
- BUSY
- Baud Rate

Please notify the interface type before ordering. RS232/ RS485/ TTL level interface.

Communication Interface Definition：



Pin Name	Pin NO.	Pin Type	Interpret
VCC	1,2	P	Power Supply Input
GND	3,4	P	Power Ground
DIN	5,	I	Data Input
DOUT	6	O	Data Output
BUSY	7	O	Issue a directive as Full Signal of UART Buffer

I: Input O: Output P: Power

Note A: 1. Adopting the 7 Pin 2mm spacing socket. (Socket Model: Molex 5023520700)

2. Direction of the signal was defined with TFT-LCD Module;
   "I" refers to the signal from the user's system transmitted to the TFT-LCD Module.
3. Pins with the same definition are connected together in the module inside.
4. User can set the RS232 or 3.3 V TTL/ CMOS level by soldering pad on the PCB. Default is "OFF" (RS232 Level). If change to "ON" (TTL Level), only need to short the soldering pad of J21, J22, J23 and take out R61, R62, R63.
5. The RS485 and other interface can be customized which need to point out in the order.
6. "Busy" signal doesn't need to be considered as communication, Frame and Command never be lost, (70MHz CPU, 1M bit Command Buffer), "BUSY" signal can be accessed to data input pin or left vacant.

Note B: The selection of Baud rate for the serial interface：

Baud rate（bps）	1200	2400	4800	9600	19200	38600	57600	115200
Bode_Set	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07

4 Physical Dimensions

This chapter contains the information of Physical Dimensions.

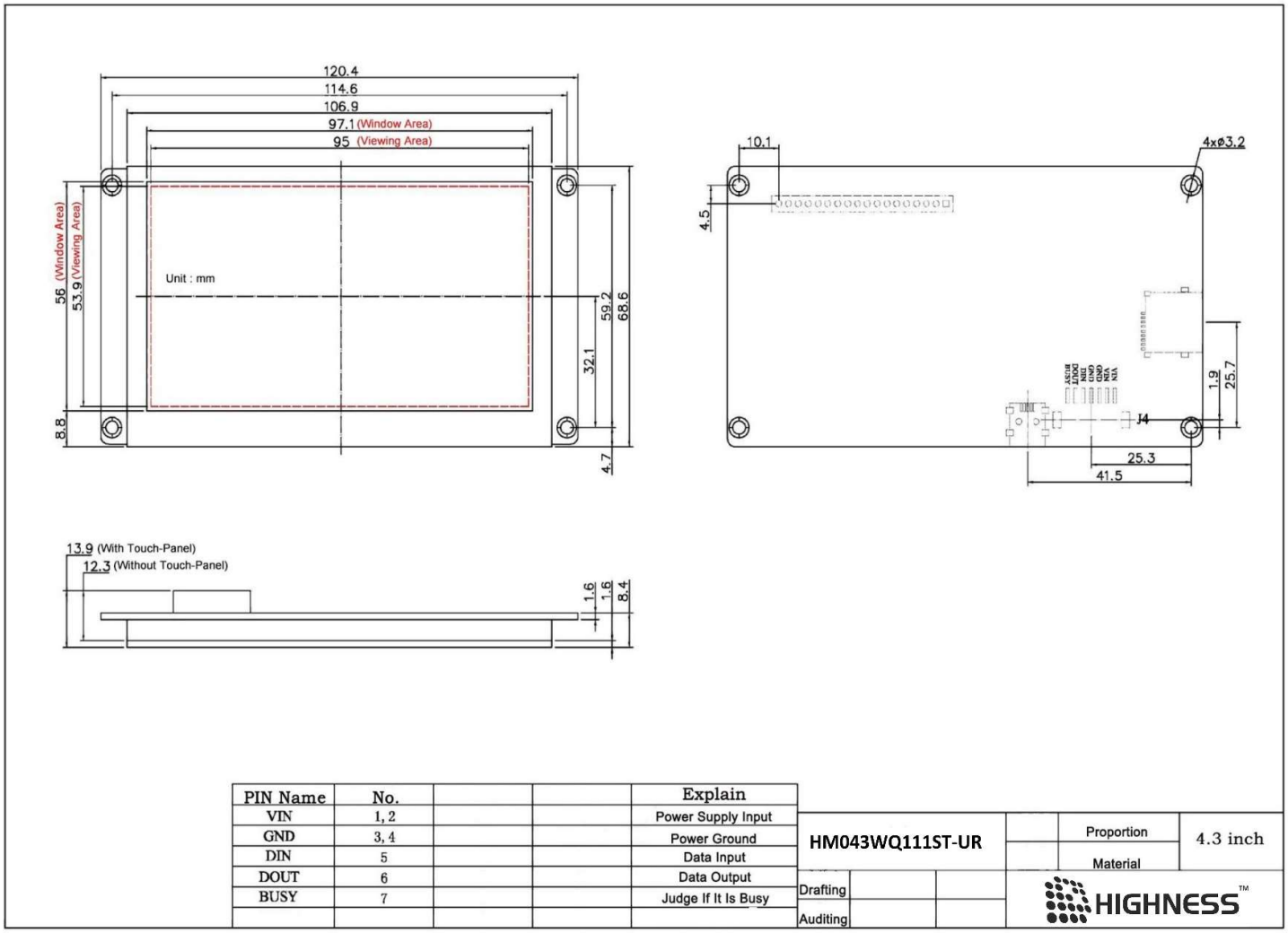


Figure 4-1 HM043WQ111ST-UR dimension

5 Command Set Table

This chapter describes the Command Set:

- Text Command
- Image Command
- Curve Command
- Touch control Command
- Work module Command

Etc.

Categories	Command	Command Parameter	Illustration
Handshake	0x00	No	Check the configuration and version
Parameter Configuration	0x40	Fcolor+Bcolor	Palette setting
	0x41	D_X(0x00-0x7F)+D_Y(0x00-0x7F)	Character space setting
	0x42	X+Y	Move the appointed color to background color palette
	0x43	X+Y	Move the appointed color to foreground color palette
	0x44	Mode+X+Y+Wide(0x01-0x1F)+Height(0x01-0x1F)	Cursor display mode setting
Text Display	0x53	X+Y+String	8x8 lattice ASCII character
	0x54		16x16 lattice GBK
	0x55		32x32 GB2312
	0x6E		12X12 GBK
	0x6F		24x24 GB2312
	0x98		Display and lattice, any encoded string
Points Setting	0x50	$(x,y)_0+(x,y)_1+\ldots\ldots+(x,y)_n$	More points setting in the background color.(delete point)
	0x51		More points in the foreground color.
	0x74	$X+Y_s+Y_e+Bcolor+(y,Fcolor)_1+\ldots..+(y,Fcolor)_n$	Dynamic curve display
	0x72	$Address(H:M:L)+Data\_word_0+\ldots\ldots Data\_word_n$	Operation to the buffer of video card.
Line& Polygon	0x56	$(x,y)_0+(x,y)_1+\ldots\ldots+(x,y)_n$	Polygon display: Line the points with foreground colored segment.
	0x5D		Polygon delete: Line the points with background colored segment
	0x75	$X+Y+Height\_max+Height_0+Height_1+\ldots\ldots+Height_n$	Spectrum display: display a continuous vertical line with the same end in a fast rhythm.
	0x76	$X+Y\_dis(0x00\text{-}0xFF)+Y_0+Y_1+\ldots\ldots+Y_n$	Line chat display(Xi=X+i*X_dis,Yi=Yi)
Acrs	0x57	$(Type,x,y,r)_0+(Type,x,y,r)_1+\ldots\ldots+(Type,x,y,r)n$	Arcs display
Rectangles	0x59	$(x_s,y_z,x_e,y_e)_0+(x_s,Y_z,x_e,y_e)_1+\ldots\ldots\ldots+(X_s,Y_z,X_e,Y_e)_n$	Show rectangles: display rectangles by foreground color)
	0x69		Delete rectangles: display rectangles by background color
Areas Operation	0x64	X+Y+Color	
	0x52	No	Clear the Screen
	0x5A	$(x_s,y_z,xe,y_e)+(x_s,y_z,x_e,y_e)+\ldots\ldots+(x_s,y_z,x_e,y_e)_n$	Areas Deleting
	0x5B		Fill in more than one appointed areas.
	0x5C		Areas color changing
	0x60	$(x_s,y_z,x_e,y_e,n)_0+(x_s,y_z,x_e,y_e,n)_1+\ldots\ldots+(x_s,y_z,x_e,y_e,n)_n$	Appointed areas ring-shifting to the left
	0x61		Appointed areas ring-shifting to the right
	0x62		Appointed areas ring-shifting to the left
	0x63		Appointed areas shifting to the right
Pictures& Icons	0x70	Picture_ID	Display a full screen image
	0x7B	Picture_ID	Display a full screen image and calculate the cumulative sum.
	0x71	$Picture\_ID+X_s+Y_s+X_e+Y_e+X+Y$	Display part of a picture in the memory(background display)
	0x9C	$Picture\_ID+X_s+Y_s+Y_e+X_e+Y$	Display a part from an image which stored in the module(background not shown),automatically restore the current image background .
	0x9D	$Picture\_ID+X_s+Y_s+X_e+Y_e+X+Y$	Display part of a picture in the memory(background does not display)
	0xE2	Picture ID	Picture saving
	0x99	$(x,y,Icon\_ID)_0+(x,y,Icon\_ID)_1+\ldots\ldots+(x,y,Icon\_ID)_n$	User-defined icons display
Animation	0x9A	0Xff/Pack_ID	Turn off/on the automatic implementation of Command Set
Temporary	0xC0	$Address(H:L)+Data\_word_0+\ldots\ldots\ldots+Data\_word_n$	Writing data to the temporary buffer

Buffer Operation	0xC1	0x01+Address+Pixel_Number(H:L)	Display the pre-set date lines in the temporary buffer
		0x02+Address+Line_Number(H:L)	Display the pre-set date lines in the temporary buffer
		0x03+Address+X+Y+Line_Number+D_x+Dis_x+K_y+Color	Dynamic curve scaling: connecting the data points in the temporary buffer zone
		0x04+Addr1+X+Y+Line_Number+0x01+Dis_x+Color1+Addr0+Color0	Oscillometer: connecting the data points in temporary buffer in a flicker-free high-speed
		0x05+Address+X+Y+Line_Number+D_x+Dis_x+M_y+D_y+Color	Using the data in the temporary buffer to display line charts.
		0x06+Address+X+Y+Line_Number+D_x+Dis_x+M_y+Color+Ymin+Ymax	Using the data in the temporary buffer zoom to display a window-constrained bi-directional line chart
		0x10+Address+Frame_Number	Using the command in the temporary buffer to perform a synchronize display
	0xC2	<Address>+<Data_length>	Read back data from the temporary buffer.
Database Operation	0xF2	0xF2+0xF2+0x5A+0xA5+Lib_ID	Font modification
	0x90	0x55+0xAA+0x5A+0xA5+Address(H:MH:ML:L)+Data	Write data to the user's database(32MB)
	0x91	Address+Read_Length(H:L)	Read data from the database(32MB)
Keyboard Operation	0x71	K_code	Key code uploading
	0xE5	0x55+0xAA+0x5A+0x+A5+K_Code$_0$+……+K_Code$_n$	Key code port modification
Touchpad Operation	0x72	Touch_X+Touch_Y	Uploading the last data after the touch-screen is released,(which can turn off by 0xE0 Command)
	0x73		Uploading data when pressing the touch panel(uploading once only by setting the command of 0xE0)
	0xE4	0x55+0xAA+0x5A+0xA5	Touch panel adjusting
	0x78	Touch Code	Uploading the defaulted key code when switching the touch interface.
	0x79		
Buzzer Operation	0x79	BZ_time	Buzzing once only (10xBz_time mS)
Backlight Control	0x5E	Non or 0x55+0x AA+0x5A+0xA5+V_ON+V_OFF+ON_TIME	Turn off the backlight or control the backlight mode by touching or keying
	0x5F	Non or PWM_T(0x00-0x3F)	Turn the backlight on or adjusting the brightness by PWM.
Clock Operation	0x 9B	0x5A,0x5B(read)/0x00(off)/0xFF+M+TM+Color+X+Y(ON)	Clock on/off; read the clock
	0x E7	0x55+0xAA+0x5A+0xA5+YY:MM:DD:HH:MM:SS	Clock adjusting
Parameter Configuration	0x E0	0x55+0 x AA+0x5A+0x A5+Panel_Set+Bode_Set+Para1	Configuring the user's serial port speed and the touch-screen data uploading.
Algorithm	0xB0	Download:0x01+PY_Code answer:0x01+HZ_num+String	
		Download: 0x02+A+B+C+D answer:0x02+E+F	Calculating(AxB+C)/D,E is 4 bytes reminder
		Download:0x03+Data_Pack0 answer:0x03+Data_Pack1	Array listing of unsigned integers(2 bytes)
		Download:0x04+PY_Code answer:0x04+HZ_num+String	PINYIN input based on GBK
Volume Operation	0x30	Start_Set+Play_number+Play_time	Play the music in the appointed zoom
	0x32	Volume_L+Volume_R+0x00	Volume adjusting
	0x33	0x55+0 x AA+0x5A	Stop playing

6 APPENDIX

A MCU Sample Program

B MCU Circuit Design

C ESD Guidelines

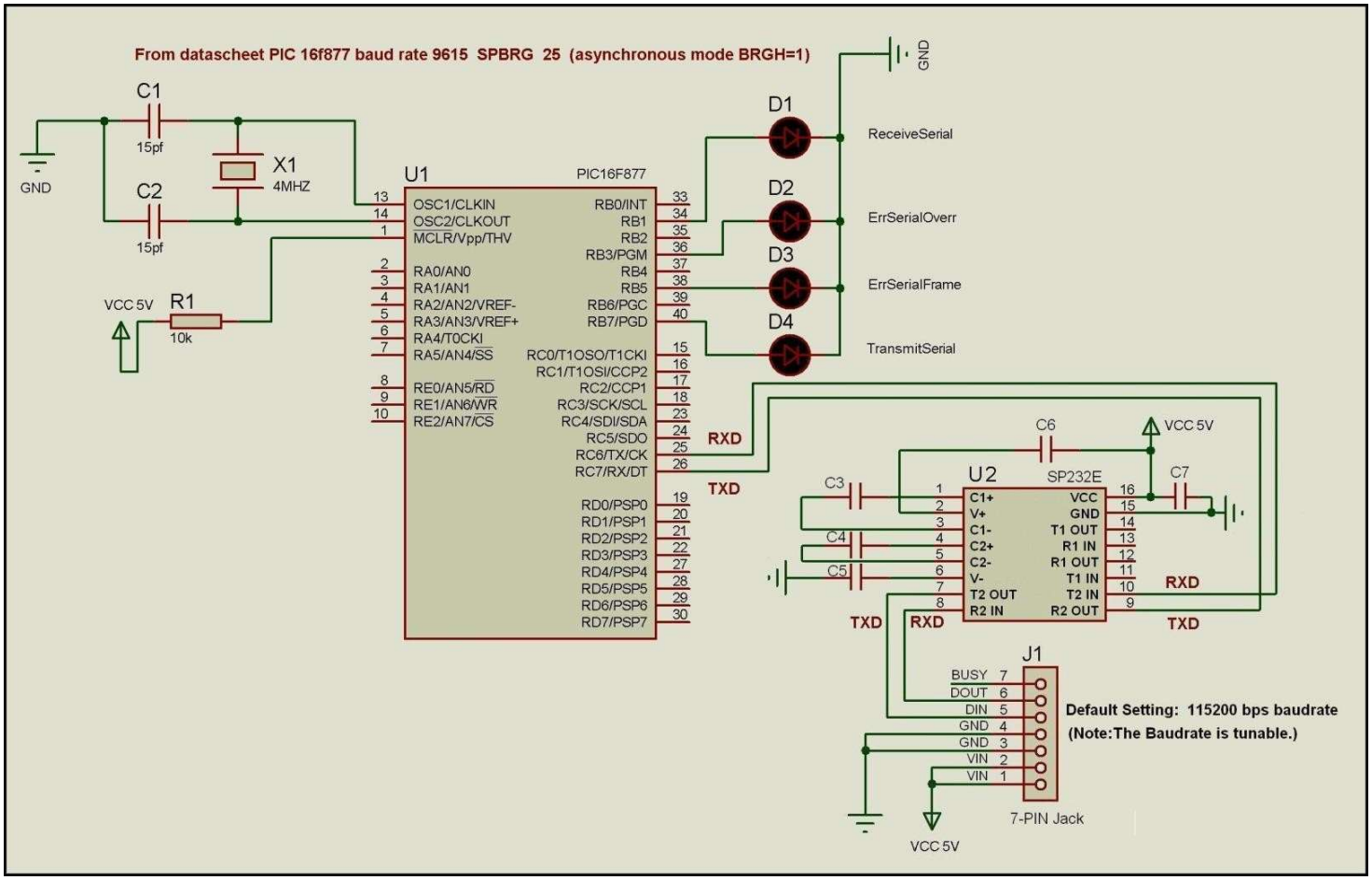
A. MCU Sample Program

C8051 MCU C Language

```
//-----------------------------------------------------------------------------
// Includes
//-----------------------------------------------------------------------------
#include<reg52.h>

//-----------------------------------------------------------------------------
// sbit  Definitions
//-----------------------------------------------------------------------------
sbit  LED=P0^0;

//-----------------------------------------------------------------------------
// Global CONSTANTS
//-----------------------------------------------------------------------------
#define SYSCLK       22118400                  // "SYSCLK frequency in Hz"
#define BAUD_RATE    115200                    // "Baud rate"
#define uchar unsigned char
#define uint unsigned int
uchar pic[3]={0xAA,0x70,0x08};
/-----------------------------------------------------------------------------
// Function PROTOTYPES
//-----------------------------------------------------------------------------
void Uart0_transmit(uchar i);                  // "Send a byte to the terminal"
void send_str(uchar *p,uchar s);               // "Send a string to the terminal"
void delay_ms(uchar n);                        // "Delay"
void SysInit(void);                            // "Initialization of system"
void en(void);                                 // "Frame end"
void pic_str(uchar i);                         // "Picture switching sub-function"

//-----------------------------------------------------------------------------
// Uart0_transmit
//-----------------------------------------------------------------------------
void Uart0_transmit(uchar i)                   // "Send 1 byte to terminal"
{
                              ES=0;
                              TI=0;
                              SBUF=i;                // "Send data to uart0"
                              while (!TI);           // "Wait for the finish of sending a byte"
                              TI=                    // "Clear the interruption mark"
                              ES=1;
}
void send_str(uchar *p,uchar s)                // "Send a string to the terminal"
{
                              uchar m;
                              for(m=0;m<s;m++)
                              {
                                 Uart0_transmit(*p);
                                p++;
                              }
}
//-----------------------------------------------------------------------------
// delay                                       // "Delay sub-function"
//-----------------------------------------------------------------------------
```

```
void delay_ms(uchar n)
{
                    uint i,j;
                    for(i=1000;i>0;i--) {
                    for(j=25*n;j>0;j--) {;}
                    }
}

//-----------------------------------------------------------------------------
// SysInit                                  // "Initialization of system"
//-----------------------------------------------------------------------------
void SysInit(void)
{
                          PCON |=0x80;
                          SCON=0x50;
                          TMOD=0x21;
                          TH1=255;
                          TL1=255;
                          TR1=1;
        ES=0;
        TH0=0xDC;
        TL0=0x00;
        TR0=1;
        ET0=1;
}


//-----------------------------------------------------------------------------
// pic_str                                  // "Picture switching sub-function"
//-----------------------------------------------------------------------------
void pic_str(uchar i)
{
            pic[2]=i;
            send_str(pic,3);                // "Send the command of picture switching"
            en();
}

//-----------------------------------------------------------------------------
// main() Routine
//-----------------------------------------------------------------------------
void main (void)                            // "main function"
{
  EA=0;                                     // "Close Interruption"
  SysInit();
  EA=1;                                     // "Open Interruption"
  delay_ms(40);

    while (1)
          {
       pic_str();                           // "Picture switching"
          }
           Return 0;
}
//-----------------------------------------------------------------------------
// End Of File
//-----------------------------------------------------------------------------
```

B. MCU Circuit Design



C. ESD Guidelines

What does ESD mean?

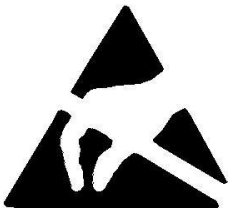
Virtually all present-day modules incorporate highly integrated MOS devices or components. For technological reasons, these electronic components are very sensitive to overvoltages and consequently therefore to electrostatic discharge:

These devices are referred to in German as Elektrostatisch Gefährdeten Bauelemente/ Baugruppen: ºEGBº

The more frequent international name is:

ºESDº (E lectrostatic Sensitive Device)

The following symbol on plates on cabinets, mounting racks or packages draws attention to the use of electrostatic sensitive devices and thus to the contact sensitivity of the assemblies concerned:



ESDs may be destroyed by voltages and energies well below the perception threshold of persons. Voltages of this kind occur as soon as a device or an assembly is touched by a person who is not electrostatically discharged . Devices exposed to such overvoltages cannot immediately be detected as defective in the majority of cases since faulty behavior may occur only after a long period of operation.

Precautions against electrostatic discharge

Most plastics are capable of carrying high charges and it is therefore imperative that they be kept away from sensitive components.

When handling electrostatic sensitive devices, make sure that persons, workplaces and packages are properly grounded.

Handling ESD assemblies

A general rule is that assemblies should be touched only when this cannot be avoided owing to the work that has to be performed on them. Under no circumstances should you handle printed-circuit boards by touching device pins or circuitry.

You should touch devices only if

- you are grounded by permanently wearing an ESD wrist strap or
- you are wearing ESD shoes or ESD shoe-grounding protection straps in conjunction with an ESD floor.

Before you touch an electronic assembly, your body must be discharged. The simplest way of doing this is to touch a conductive, grounded object immediately beforehand ± for example, bare metal parts of a cabinet, water pipe etc.

Assemblies should not be brought into contact with charge-susceptible and highly insulating materials such as plastic films, insulating table tops and items of clothing etc. containing synthetic fibers.

Assemblies should be deposited only on conductive surfaces (tables with an ESD coating, conductive ESD cellular material, ESD bags, ESD shipping containers).

Do not place assemblies near visual display units, monitors or television sets (minimum distance to screen > 10 cm).

Measuring and modifying ESD assemblies

Perform measurements on ESD assemblies only when

- the measuring instrument is grounded ± for example, by means of a protective conductor ± or
- the measuring head has been briefly discharged before measurements are made with a potential-free measuring instrument ± for example, by touching a bare metal control cabinet.

When soldering, use only grounded soldering irons.

Shipping ESD assemblies

Always store and ship assemblies and devices in conductive packing ± for example, metallized plastic boxes and tin cans.

If packing is not conductive, assemblies must be conductively wrapped before they are packed. You can use, for example, conductive foam rubber, ESD bags, domestic aluminum foil or paper (never use plastic bags or foils).

With assemblies containing fitted batteries, make sure that the conductive packing does not come into contact with or short-circuit battery connectors. If necessary, cover the connectors beforehand with insulating tape or insulating material.

Glossary

B

Baud rate

Rate of speed at which data is downloaded. Baud rate is specified in Bit/s.

Boot

A loading process which downloads the operating system in the working memory of the operating unit.

C

Command Set

Hex Code, the MCU can control the TFT Module via the command set.

Configuration file

It can be created by the softwares.

D

Download

Download the image, configuration files and data through mini USB port or TF card.

Download mode

Through mini USB port or TF card.

F

Flash memory

Programmable memory which can be electrically deleted and written to again segment-by-segment.

H

Half Brightness Life

The period of time after which the brightness tube only achieves 50% of the original value.

I

Input field

Enables the user to enter values which are subsequently sent to the **MCU**.

M

MCU

Micro Control Unit, it is widely used in the industrial control.

N

Normal operation

Operating unit operating mode in which messages are displayed and screens can be operated.

Output field

Displays current values from the **MCU** on the operating unit.

Process screen

The display of process values and process progress on the operating unit in the form of screens, which may contain graphics, texts and values.

RS485

Standard interface for serial data transfer at a very high transmission rate.

Screen

A screen displays all the logically related process data on the operating unit, whereby the individual values can be modified.

Touch panel

This is an operating unit without a keyboard. The touch panel (abbreviated to TP) is operated via the contact-sensitive screen elements.