3.5inch 8&16BIT Module MRB3511 User Manual

Product Description

The product is a 3.5-inch TFT LCD display module with capacitive touch screen. It has 480x320 resolution, supports 16BIT RGB 65K color display, and the internal driver IC is ILI9488, which uses 8-bit or 16-bit parallel port communication. The module includes LCD display, capacitive touch screen and PCB backplane. It can be plugged into the TFT LCD slot of the STM32 series development board or used on the C51 platform.

Product Features

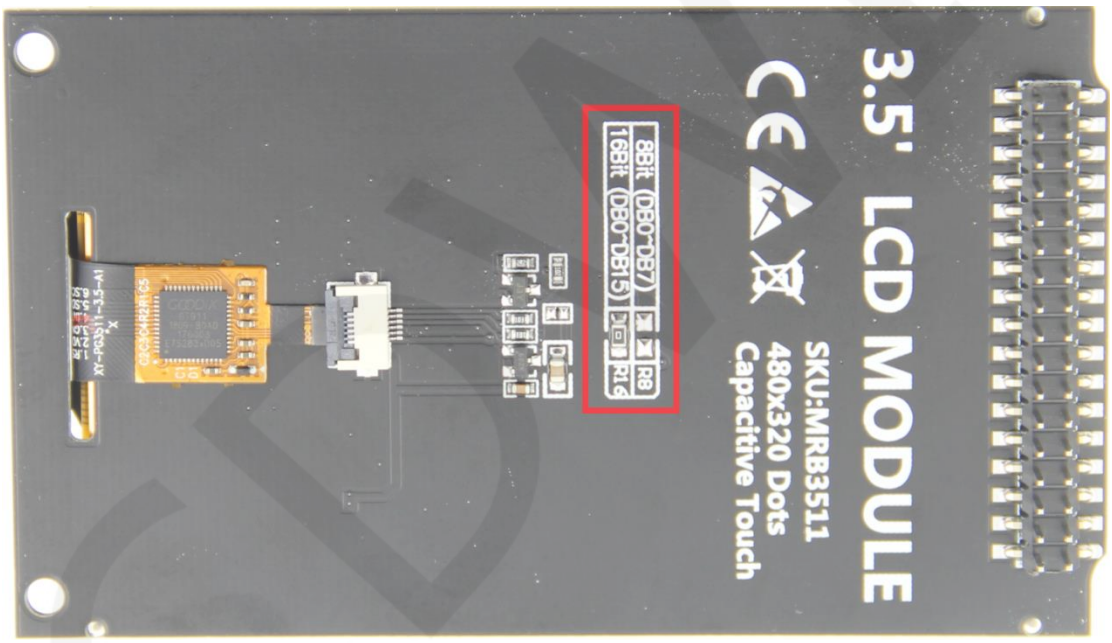
- 3.5-inch color screen, support 16BIT RGB 65K color display, display rich colors
- 320x480 resolution for clear display
- Support 8-bit or 16-bit parallel data bus mode switching, fast transfer speed
- Supports ALIENTEK STM32 Mini, Elite, WarShip, Explorer, and Apollo development boards direct plug-in use
- Support for capacitive touch function
- Provides a rich sample program for STM32 and C51 platforms
- Military-grade process standards, long-term stable work
- Provide underlying driver technical support

Product Parameters

Name	Description
Display Color	16BIT RGB 65K color
SKU	MRB3511
Screen Size	3.5(inch)
Screen Type	TFT
Driver IC	ILI9488
Resolution	480*320 (Pixel)
Module Interface	8Bit or 16Bit parallel interface

Active Area	48.96x73.44 (mm)
Touch Screen Type	Capacitive touch screen
Touch IC	GT911
Module PCB Size	56.41x97.60 (mm)
Operating Temperature	-10℃~60℃
Storage Temperature	-20℃~70℃
Operating Voltage	3.3V / 5V
Power Consumption	TBD
Product Weight(Including packaging)	57g

Interface Description



Picture1. Module rear view

Picture2. Module Pin silk screen picture

NOTE:

1. The module hardware supports 8-bit and 16-bit parallel port data bus mode switching (shown in red box in Picture 1), as follows:
   A. Solder R16 with 0Ω resistor or short circuit directly, and disconnect R8: Select 16-bit parallel port data bus mode, use DB0~DB15 data pins(default)
   B. Solder R8 with 0Ω resistor or short circuit directly, and disconnect R16: Select 8-bit parallel port data bus mode, use DB0~DB7 data pins

Important Note:

1. The following pin numbers 1~34 are the pin number of Module pin with PCB backplane of our company. If you purchase a bare screen, please refer to the pin definition of the bare screen specification, refer to the wiring according to the signal type instead of directly Wire according to the following module pin numbers. For example: CS is 1 pin on our module. It may be x pin on different size bare screen.
2. About VCC supply voltage: If you buy a module with PCB backplane, VCC/VDD power supply can be connected to 5V or 3.3V (module has integrated ultra low dropout 5V to 3V circuit), but it is recommended to connect 3.3V, because

connecting 5V will lead to circuit Increased heat generation, affecting module life; if you buy a bare screen LCD, remember to only connect 3.3V.

3. **About the backlight voltage: The module with the PCB backplane has integrated triode backlight control circuit, which only needs to input the high level of the BL pin or the PWM wave to illuminate the backlight. If you are buying a bare screen, the LEDAx is connected to 3.0V-3.3V and the LEDKx is grounded.**

Number	Module Pin	Pin Description
1	CS	LCD reset control pin(low level enable)
2	RS	LCD register / data selection control pin (high level: register, low level: data)
3	WR	LCD write control pin
4	RD	LCD read control pin
5	RST	LCD reset control pin(low level reset)
6	DB0	LCD data bus low 8-bit pin
7	DB1	
8	DB2	
9	DB3	
10	DB4	
11	DB5	
12	DB6	
13	DB7	
14	DB8	LCD data bus high 8-bit pin(When using the 8-bit parallel port data bus mode, the upper 8-bit pin is not used.)
15	DB9	
16	DB10	
17	DB11	
18	DB12	
19	DB13	
20	DB14	
21	DB15	

22	GND	Module power ground pin
23	BL	LCD backlight control pin(High level light)
24	VDD	Module power positive pin (module has integrated voltage regulator IC, so the power supply can be connected to 5V or 3.3V)
25	VDD	
26	GND	Module power ground pin
27	GND	
28	NC	LCD backlight power positive pin (default shared onboard backlight power supply, this pin can not be connected)
29	NC	Not defined, no need to use
30	SDA	Capacitive touch screen IIC bus data pin
31	INT	Capacitive touch screen interrupt detection pin (low level when a touch occurs)
32	NC	Not defined, no need to use
33	CRST	Capacitive touch screen IC reset control pin (low level reset)
34	SCL	Capacitive touch screen IIC bus clock pin

Hardware Configuration

The LCD module hardware circuit comprises five parts: an LCD display control circuit, a power control circuit, a data bus mode selection control circuit, a touch screen control circuit and a backlight control circuit.

LCD display control circuit for controlling the pins of the LCD, including control pins and data transfer pins.

Power control circuit for stabilizing the supply voltage and selecting the external supply voltage

Data bus mode selection control circuit for selecting 8-bit or 16-bit data bus mode.

Capacitive touch screen control circuit is used to control touch screen interrupt acquisition, data sampling, AD conversion, data transmission, etc.

A backlight control circuit is used to control the brightness of the backlight.

working principle

1. Introduction to ITI9488 Controller

The ITI488 controller supports a maximum resolution of 320*480 and has a 345600-byte GRAM. It also supports 8-bit, 9-bit, 16-bit, 18-bit and 24-bit parallel port data buses. It also supports 3-wire and 4-wire SPI serial ports. Since the supported resolution is relatively large and the amount of data transmitted is large, the parallel port transmission is adopted, and the transmission speed is fast. ITI9488 also supports 65K, 262K and 16.7M RGB color display, display color is very rich, while supporting rotating display and scroll display and video playback, display in a variety of ways.

The ITI9488 controller uses 16bit (RGB565) to control a pixel display, so it can display up to 65K colors per pixel. The pixel address setting is performed in the order of rows and columns, and the incrementing and decreasing direction is determined by the scanning mode. The ITI9488 display method is performed by setting the address and then setting the color value.

2. Introduction to parallel port communication

The parallel port communication write mode timing is as shown below:

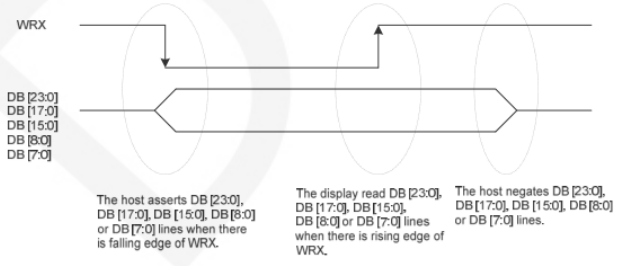
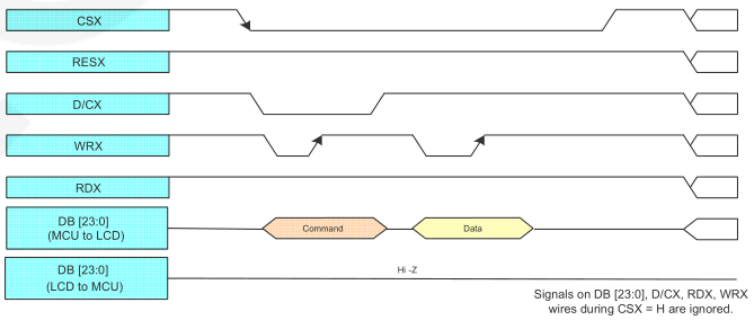


Figure 1: DBI Type B Write Cycle

Note: WRX is an unsynchronized signal that can be terminated when not being used.

When the D/CX signal is driven to low level, the input data on the interface is interpreted as command information.
The D/CX signal can also be pulled to high level when the data is RAM data or command parameter.



The timing of the parallel port communication read mode is shown in the figure below:

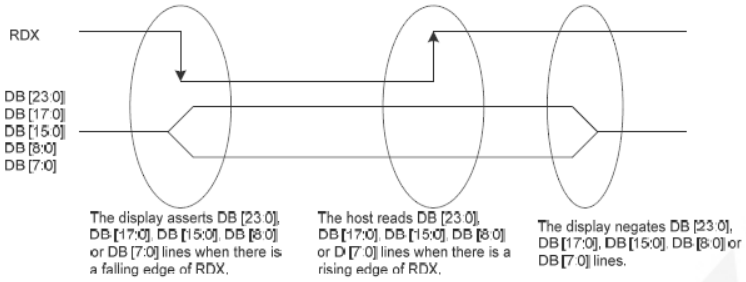
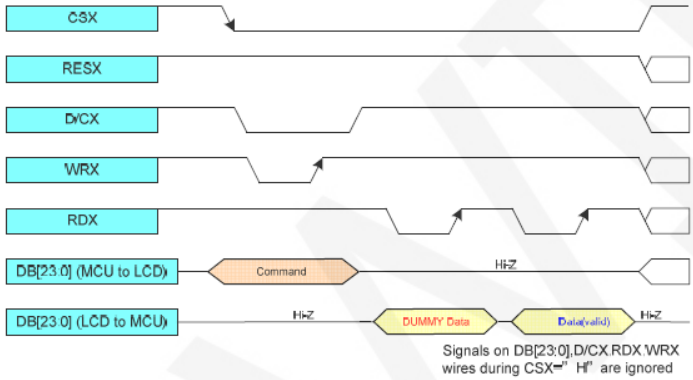


Figure 3: DBI Type B Read Cycle

Note: RDX is an unsynchronized signal that can be terminated when not being used.

When the D/CX signal is driven to the low level, the input data on the interface is interpreted as internal status or parameter data. The D/CX signal can also be pulled to a high level when the data on the interface is RAM data or a command parameter data.



CSX is a chip select signal for enabling and disabling parallel port communication, active low

RESX is an external reset signal, active low

D/CX is the data or command selection signal, 1-write data or command parameters, 0-write command

WRX is a write data control signal

RDX is a read data control signal

D[X:0] is a parallel port data bit, which has four types: 8-bit, 9-bit, 16-bit, and 18-bit.

When performing a write operation, on the basis of the reset, first set the data or command selection signal, then pull the chip select signal low, then input the content to be written from the host, and then pull the write data control signal low. When pulled high, data is written to the LCD control IC on the rising edge of the write control signal. Finally, the chip select signal is pulled high and a data write operation is completed.

When entering the read operation, on the basis of the reset, first pull the chip select signal low, then pull the data or command select signal high, then pull the read data control signal low, and then read the data from the LCD control IC. And then The read data control signal is pulled high, and the data is read out on the rising edge of the read data control signal. Finally, the chip select signal is pulled high, and a data read operation is completed.

Instructions for use

1. STM32 instructions

Wiring instructions:

See the interface description for pin assignments.

Note:

1. **This module can be directly inserted into the TFTLCD slot of the punctual atom development board, no manual wiring is required.**
2. **The following internal plug-in pins of the corresponding MCU refer to the MCU pins directly connected to the TFTLCD slot inside the development board, only for reference.**

MiniSTM32 development board TFTLCD socket in-line instructions			
Number	**Module Pin**	**Corresponding TFTLCD socket pin**	**Corresponding to STM32F103RCT6 microcontroller internal connection pin**
1	**CS**	CS	PC9
2	**RS**	RS	PC8
3	**WR**	WR	PC7
4	**RD**	RD	PC6
5	**RST**	RST	PC4
6	**DB0**	D0	PB0
7	**DB1**	D1	PB1
8	**DB2**	D2	PB2
9	**DB3**	D3	PB3
10	**DB4**	D4	PB4

11	**DB5**	D5	PB5
12	**DB6**	D6	PB6
13	**DB7**	D7	PB7
14	**DB8**	D8	PB8
15	**DB9**	D9	PB9
16	**DB10**	D10	PB10
17	**DB11**	D11	PB11
18	**DB12**	D12	PB12
19	**DB13**	D13	PB13
20	**DB14**	D14	PB14
21	**DB15**	D15	PB15
22	**GND**	GND	GND
23	**BL**	BL	PC10
24	**VDD**	3.3	3.3V
25	**VDD**	3.3	3.3V
26	**GND**	GND	GND
27	**GND**	GND	GND
28	**NC**	Not used	5V
29	**NC**	Not used	PC2
30	**SDA**	MOSI	PC3
31	**INT**	PEN	PC1
32	**NC**	Not used	NC
33	**CRST**	TCS	PC13
34	**SCL**	CLK	PC0

Elite STM32 development board TFTLCD socket in-line instructions			
Number	Module Pin	**Corresponding TFTLCD socket pin**	**Corresponding to STM32F103ZET6 microcontroller internal connection pin**
1	**CS**	CS	PG12
2	**RS**	RS	PG0
3	**WR**	WR	PD5

4	**RD**	RD	PD4
5	**RST**	RST	reset pin
6	**DB0**	D0	PD14
7	**DB1**	D1	PD15
8	**DB2**	D2	PD0
9	**DB3**	D3	PD1
10	**DB4**	D4	PE7
11	**DB5**	D5	PE8
12	**DB6**	D6	PE9
13	**DB7**	D7	PE10
14	**DB8**	D8	PE11
15	**DB9**	D9	PE12
16	**DB10**	D10	PE13
17	**DB11**	D11	PE14
18	**DB12**	D12	PE15
19	**DB13**	D13	PD8
20	**DB14**	D14	PD9
21	**DB15**	D15	PD10
22	**GND**	GND	GND
23	**BL**	BL	PB0
24	**VDD**	VDD	3.3V
25	**VDD**	VDD	3.3V
26	**GND**	GND	GND
27	**GND**	GND	GND
28	**NC**	Not used	5V
29	**NC**	Not used	PB2
30	**SDA**	MOSI	PF9
31	**INT**	PEN	PF10
32	**NC**	Not used	NC
33	**CRST**	TCS	PF11
34	**SCL**	CLK	PB1

WarShip STM32 development board TFTLCD socket in-line instructions				
Number	Module Pin	Corresponding TFTLCD socket pin	Corresponding to STM32F103ZET6 microcontroller internal connection pin	
			V2	V3
1	CS	CS	PG12	
2	RS	RS	PG0	
3	WR	WR	PD5	
4	RD	RD	PD4	
5	RST	RST	reset pin	
6	DB0	D0	PD14	
7	DB1	D1	PD15	
8	DB2	D2	PD0	
9	DB3	D3	PD1	
10	DB4	D4	PE7	
11	DB5	D5	PE8	
12	DB6	D6	PE9	
13	DB7	D7	PE10	
14	DB8	D8	PE11	
15	DB9	D9	PE12	
16	DB10	D10	PE13	
17	DB11	D11	PE14	
18	DB12	D12	PE15	
19	DB13	D13	PD8	
20	DB14	D14	PD9	
21	DB15	D15	PD10	
22	GND	GND	GND	
23	BL	BL	PB0	
24	VDD	VDD	3.3V	
25	VDD	VDD	3.3V	
26	GND	GND	GND	
27	GND	GND	GND	
28	NC	Not used	5V	

29	**NC**	Not used	PF8	PB2
30	**SDA**	MOSI	PF9	
31	**INT**	PEN	PF10	
32	**NC**	Not used	NC	
33	**CRST**	TCS	PB2	PF11
34	**SCL**	CLK	PB1	

Explorer STM32F4 development board TFTLCD socket in-line instructions			
Number	**Module Pin**	**Corresponding TFTLCD socket pin**	**Corresponding to STM32F407ZGT6 microcontroller internal connection pin**
1	**CS**	CS	PG12
2	**RS**	RS	PF12
3	**WR**	WR	PD5
4	**RD**	RD	PD4
5	**RST**	RST	reset pin
6	**DB0**	D0	PD14
7	**DB1**	D1	PD15
8	**DB2**	D2	PD0
9	**DB3**	D3	PD1
10	**DB4**	D4	PE7
11	**DB5**	D5	PE8
12	**DB6**	D6	PE9
13	**DB7**	D7	PE10
14	**DB8**	D8	PE11
15	**DB9**	D9	PE12
16	**DB10**	D10	PE13
17	**DB11**	D11	PE14
18	**DB12**	D12	PE15
19	**DB13**	D13	PD8
20	**DB14**	D14	PD9
21	**DB15**	D15	PD10

22	**GND**	GND	GND
23	**BL**	BL	PB15
24	**VDD**	VDD	3.3V
25	**VDD**	VDD	3.3V
26	**GND**	GND	GND
27	**GND**	GND	GND
28	**NC**	Not used	5V
29	**NC**	Not used	PB2
30	**SDA**	MOSI	PF11
31	**INT**	PEN	PB1
32	**NC**	Not used	NC
33	**CRST**	TCS	PC13
34	**SCL**	CLK	PB0

Apollo STM32F4/F7 development board TFTLCD socket in-line instructions			
Number	**Module Pin**	**Corresponding TFTLCD socket pin**	**Corresponding to STM32F429IGT6、STM32F767IGT6、STM32H743IIT6 microcontroller internal connection pin**
1	**CS**	CS	PD7
2	**RS**	RS	PD13
3	**WR**	WR	PD5
4	**RD**	RD	PD4
5	**RST**	RST	reset pin
6	**DB0**	D0	PD14
7	**DB1**	D1	PD15
8	**DB2**	D2	PD0
9	**DB3**	D3	PD1
10	**DB4**	D4	PE7
11	**DB5**	D5	PE8
12	**DB6**	D6	PE9
13	**DB7**	D7	PE10

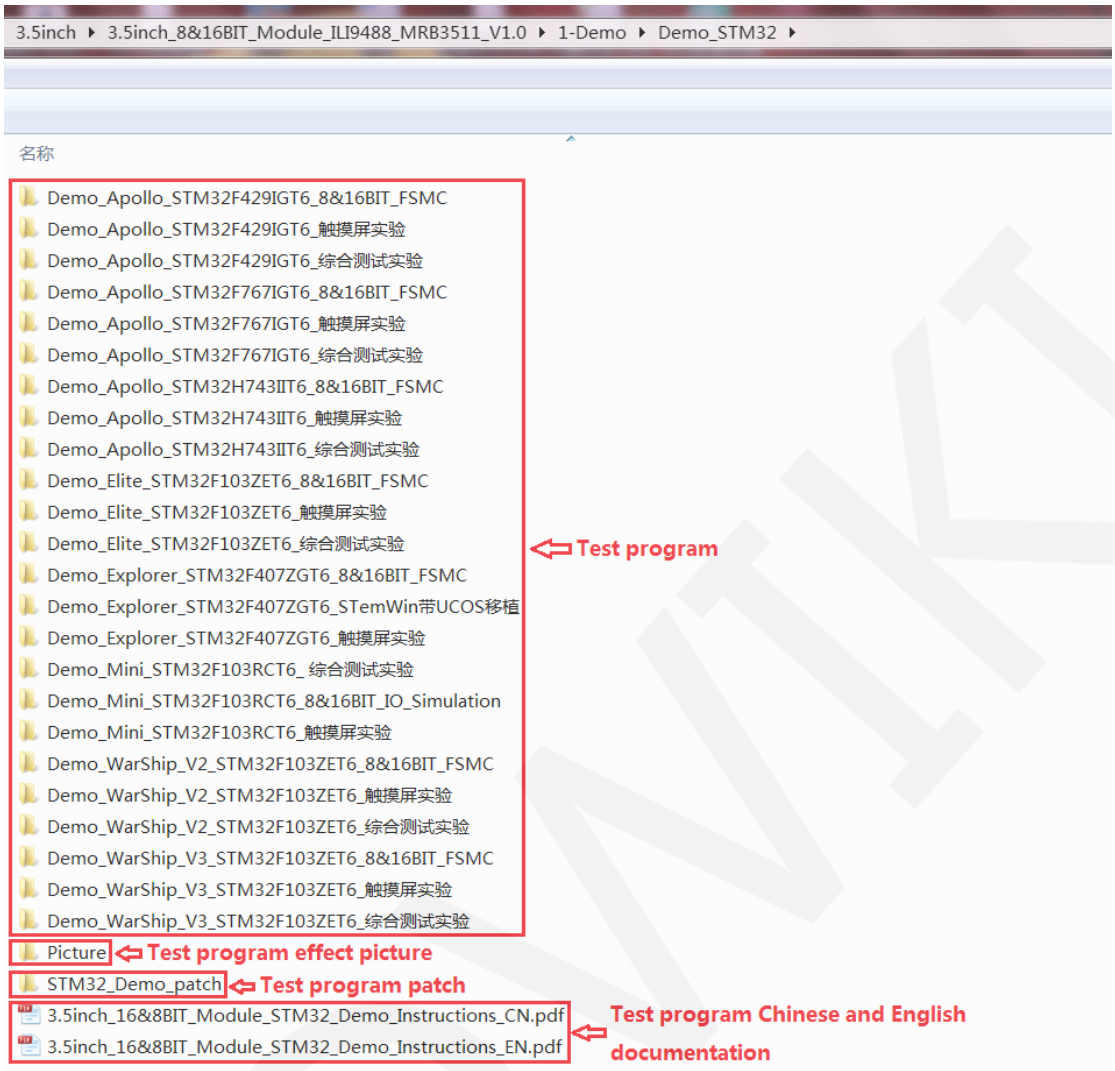
14	**DB8**	D8	PE11
15	**DB9**	D9	PE12
16	**DB10**	D10	PE13
17	**DB11**	D11	PE14
18	**DB12**	D12	PE15
19	**DB13**	D13	PD8
20	**DB14**	D14	PD9
21	**DB15**	D15	PD10
22	**GND**	GND	GND
23	**BL**	BL	PB5
24	**VDD**	VDD	3.3V
25	**VDD**	VDD	3.3V
26	**GND**	GND	GND
27	**GND**	GND	GND
28	**NC**	Not used	5V
29	**NC**	Not used	PG3
30	**SDA**	MOSI	PI3
31	**INT**	PEN	PH7
32	**NC**	Not used	NC
33	**CRST**	TCS	PI8
34	**SCL**	CLK	PH6

Operating Steps：

A. Connect the LCD module(As shown in Picture 1) and the STM32 MCU according to the above wiring instructions, and power on;

B. Select the C51 test program to be tested, as shown below:

(Please refer to the test program documentation for the test program description. If you need to use patch, please refer to the patch documentation in the STM32_Demo_patch directory.)



C. Open the selected test program project, compile and download;

detailed description of the STM32 test program compilation and download can be found in the following document:

http://www.lcdwiki.com/res/PublicFile/STM32_Keil_Use_Illustration_EN.pdf

D. If the LCD module displays characters and graphics normally, the program runs successfully;

2. C51 instructions

Wiring instructions:

See the interface description for pin assignments.

Note:

1. **Since the input and output levels of the GPIO of the STC12C5A60S2**

microcontroller are 5V, the capacitive touch IC cannot work normally (only 1.8~3.3V can be accepted). If you want to use the capacitive touch function, you need to connect to the level conversion module;

2. Since the STC89C52RC microcontroller does not have a push-pull output function, the backlight control pin needs to be connected to a 3.3V power supply to be properly lit.
3. Since the STC89C52RC microcontroller's Flash capacity is too small (less than 25KB), the program with touch function cannot be downloaded, so the touch screen does not need wiring.

STC12C5A60S2 microcontroller test program wiring instructions		
Number	**Module Pin**	**Corresponding to STC12 development board wiring pin**
1	**CS**	P13
2	**RS**	P12
3	**WR**	P11
4	**RD**	P10
5	**RST**	P33
6	**DB0**	P00
7	**DB1**	P01
8	**DB2**	P02
9	**DB3**	P03
10	**DB4**	P04
11	**DB5**	P05
12	**DB6**	P06
13	**DB7**	P07
14	**DB8**	P20
15	**DB9**	P21
16	**DB10**	P22
17	**DB11**	P23
18	**DB12**	P24

19	**DB13**	P25
20	**DB14**	P26
21	**DB15**	P27
22	**GND**	GND
23	**BL**	P32
24	**VDD**	3.3V/5V
25	**VDD**	3.3V/5V
26	**GND**	GND
27	**GND**	GND
28	**NC**	No need to connect
29	**NC**	No need to connect
30	**SDA**	P34
31	**INT**	P40
32	**NC**	No need to connect
33	**CRST**	P37
34	**SCL**	P36

STC89C52RC microcontroller test program wiring instructions		
Number	**Module Pin**	**Corresponding to STC89 development board wiring pin**
1	**CS**	P13
2	**RS**	P12
3	**WR**	P11
4	**RD**	P10
5	**RST**	P14
6	**DB0**	P30
7	**DB1**	P31
8	**DB2**	P32
9	**DB3**	P33
10	**DB4**	P34
11	**DB5**	P35
12	**DB6**	P36

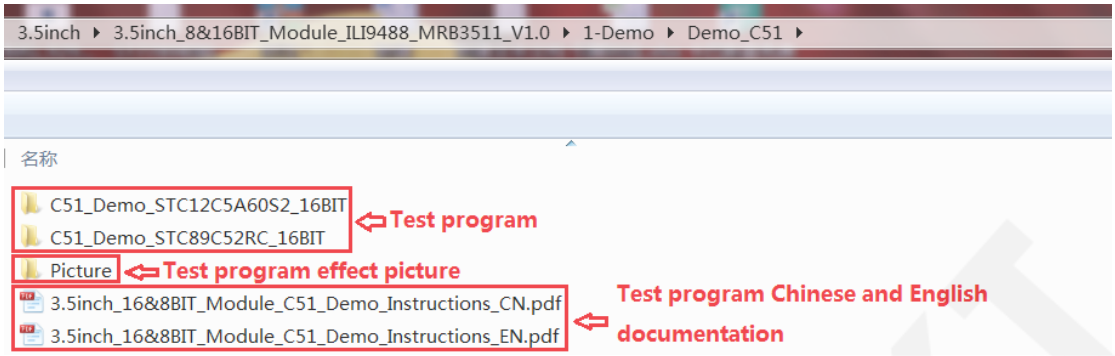
13	DB7	P37
14	DB8	P20
15	DB9	P21
16	DB10	P22
17	DB11	P23
18	DB12	P24
19	DB13	P25
20	DB14	P26
21	DB15	P27
22	GND	GND
23	BL	3.3V
24	VDD	3.3V/5V
25	VDD	3.3V/5V
26	GND	GND
27	GND	GND
28	NC	No need to connect
29	NC	No need to connect
30	SDA	No need to connect
31	INT	No need to connect
32	NC	No need to connect
33	CRST	No need to connect
34	SCL	No need to connect

Operating Steps：

A. Connect the LCD module (As shown in Picture 1)and the C51 MCU according to the above wiring instructions, and power on;

B. Select the C51 test program to be tested, as shown below:

(Test program description please refer to the test program description document in the test package)



C. Open the selected test program project, compile and download;

detailed description of the C51 test program compilation and download can be found in the following document:

http://www.lcdwiki.com/res/PublicFile/C51_Keil%26stc-isp_Use_Illustration_EN.pdf

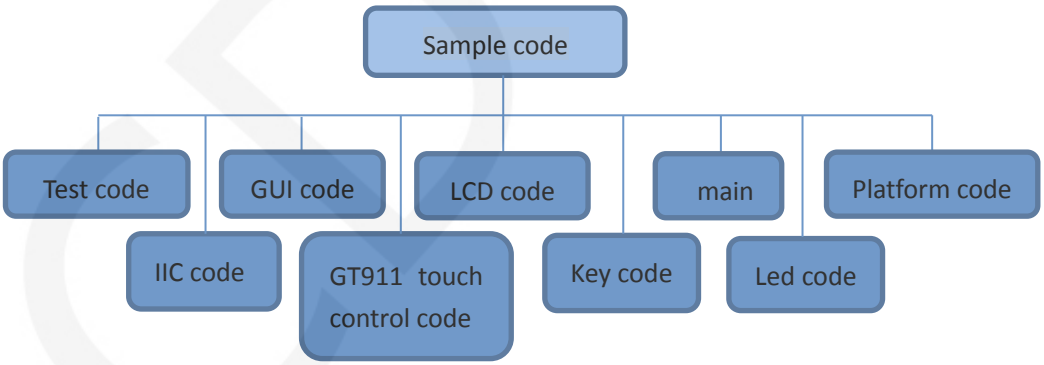
D. If the LCD module displays characters and graphics normally, the program runs successfully;

Software Description

1. Code Architecture

A. C51 and STM32 code architecture description

The code architecture is shown below:



The Demo API code for the main program runtime is included in the test code;

LCD initialization and related bin parallel port write data operations are included in the LCD code;

Drawing points, lines, graphics, and Chinese and English character display related operations are included in the GUI code;

The main function implements the application to run;

Platform code varies by platform;

The IIC code is used by the capacitive touch IC GT911, including IIC initialization, data writing and reading, etc;

GT911 touch control code includes touch detection, touch command transmission and touch data reading, etc.

The key processing related code is included in the key code (the C51 platform does not have a button processing code);

The code related to the led configuration operation is included in the led code(the C51 platform does not have a led processing code);

2. GPIO definition description

A. STM32 test program GPIO definition description

The GPIO definition of the LCD screen of the STM32 test program is placed in the lcd.h file, which is defined in two ways:

1) STM32F103RCT6 microcontroller test program uses IO analog mode (it does not support FSMC bus)
2) Other STM32 MCU test programs use FSMC bus mode

STM32F103RCT6 MCU IO analog test program LCD screen GPIO definition as shown below:

```
//////////////////////////////////////////////////////////////////////////////
//-----------------LCD端口定义----------------
#define GPIO_TYPE  GPIOC  //GPIO组类型
#define LED      10         //背光控制引脚      PC10
#define LCD_CS   9        //片选引脚          PC9
#define LCD_RS   8        //寄存器/数据选择引脚 PC8
#define LCD_RST  4        //复位引脚          PC4
#define LCD_WR   7        //写引脚            PC7
#define LCD_RD   6        //读引脚            PC6

//PB0~15,作为数据线
//注意：如果使用8位模式数据总线，则液晶屏的数据高8位是接到MCU的高8位总
//举例：如果接8位模式则本示例接线为液晶屏DB10-DB17对应接至单片机GPIOB_
//举例：如果是16位模式：DB0-DB7分别接GPIOB_Pin0-GPIOB_Pin7,DB10-DB17对
#define DATAOUT(x) GPIOB->ODR=x; //数据输出
#define DATAIN     GPIOB->IDR;   //数据输入
```

FSMC test program lcd screen GPIO is defined as shown below (take STM32F103ZET6 microcontroller FSMC test program as an example):

```
//////////////////////////////////////////////////////////////////
//-----------------LCD端口定义----------------
#define LED      0      //背光控制引脚        PB0

//QDtech全系列模块采用了三极管控制背光亮灭，用户也可以接PWM调节背光亮度
#define LCD_LED PBout(LED) //LCD背光

//LCD地址结构体
typedef struct
{
#if LCD_USE8BIT_MODEL
  vu8 LCD_REG;
  vu8 LCD_RAM;
#else
  vu16 LCD_REG;
  vu16 LCD_RAM;
#endif
} LCD_TypeDef;

//使用NOR/SRAM的 Bank1.sector4,地址位HADDR[27,26]=11 A10作为数据命令区分线
#if LCD_USE8BIT_MODEL
//使用8位模式时，STM32内部地址不需要右移一位
#define LCD_BASE        ((u32)(0x6C000000 | 0x000003FF))
#else
//使用16位模式时，注意设置时STM32内部地址需要右移一位对齐!
#define LCD_BASE        ((u32)(0x6C000000 | 0x000007FE))
#endif
#define LCD             ((LCD_TypeDef *) LCD_BASE)
```

The STM32 touch screen related GPIO definition consists of two parts: IIC's GPIO definition and screen interrupt and reset GPIO definition.

The IIC GPIO definition is placed in the gtiic.h file as shown below (take the STM32F103ZET6 microcontroller FSMC test program as an example):

```
//IO方向设置
#define GT_SDA_IN()  {GPIOF->CRH&=0XFFFFFF0F;GPIOF->CRH|=8<<4*1;}
#define GT_SDA_OUT() {GPIOF->CRH&=0XFFFFFF0F;GPIOF->CRH|=3<<4*1;}

//IO操作函数
#define GT_IIC_SCL    PBout(1)       //SCL
#define GT_IIC_SDA    PFout(9)       //SDA
#define GT_READ_SDA   PFin(9)        //输入SDA
```

The interrupt of the screen and the reset GPIO definition are placed in GT911.h, as shown in the following figure (take the STM32F103ZET6 microcontroller FSMC test program as an example):

```
#define RST_OUT() { GPIOF->CRH&=0XFFFF0FFF;GPIOF->CRH|=0X00003000;}   //set RSSET pin to output
#define INT_OUT() { GPIOF->CRH&=0XFFFFF0FF;GPIOF->CRH|=0X00000300;}   //set RSSET pin to output
#define INT_IN()  { GPIOF->CRH&=0XFFFFFFFF;GPIOC->CRH|=0X00000400;}   //set RSSET pin to output
//#define INT_OUT() { GPIOB->CRH&=0XFFFFFFF0;GPIOB->CRH|=0X00000003;}   //set INT pin to output
//#define INT_IN()  { GPIOB->CRH&=0XFFFFFFF0;GPIOB->CRH|=0X00000004;}   //set INT pin to input

//RST--PF11
//INT--PF10
#define RST_CTRL      PFout(11) //GT911 RESET pin out high or low
#define INT_CTRL      PFout(10)   //GT911 INT pin out high or low
#define INT_GET       PFin(10)  //Get GT911 INT pin status
```

B. C51 test program GPIO definition description

C51 test program lcd screen GPIO definition is placed in the lcd.h file, as shown below(Taking the STC12C5A60S2 microcontroller test program as an example):

```
//IO连接
#define  LCD_DataPortH P2      //高8位数据口,8位模式下只使用高8位
#define  LCD_DataPortL P0      //低8位数据口 ,8位模式下低8位可以不接线
sbit LCD_RS = P1^2;       //数据/命令切换
sbit LCD_WR = P1^1;      //写控制
sbit LCD_RD = P1^0;         //读控制
sbit LCD_CS = P1^3;    //片选
sbit LCD_RESET = P3^3;          //复位
sbit LCD_BL=P3^2;    //背光控制，如果不需要控制，接3.3V
```

Parallel pin definition needs to select the whole set of GPIO port groups, such as P0, P2, etc., so that when transferring data, the operation is convenient.Other pins can be defined as any free GPIO.

The C51 touch screen related GPIO definition consists of two parts: IIC's GPIO definition and screen interrupt and reset GPIO definition.

The IIC GPIO definition is placed in the gtiic.h file as shown below (take the STC12C5A60S2 microcontroller test program as an example):

```
//引脚定义
sbit GT_IIC_SCL = P3^6;      //SCL
sbit GT_IIC_SDA = P3^4;      //SDA
```

The interrupt of the screen and the reset GPIO definition are placed in GT911.h, as shown in the following figure (take the STC12C5A60S2 microcontroller test program as an example):

```
//引脚定义
sfr    P4     = 0xC0;
sbit RST_CTRL = P3^7; //GT911 RESET pin out high or low
sbit INT_CTRL = P4^0; //GT911 INT pin out high or low
```

The GPIO definition of the touch screen can be modified and can be defined as any other free GPIO.

If the microcontroller does not have a P4 GPIO group, penirq can be defined as other GPIOs.

3. Parallel port communication code implementation

A. STM32 test program parallel port communication code implementation

The STM32 test program parallel port communication code is placed in the LCD.c file, which is implemented in two ways:

1) STM32F103RCT6 microcontroller test program uses IO analog mode (it does not support FSMC bus)
2) Other STM32 MCU test programs use FSMC bus mode

The IO simulation test program is implemented as shown below:

```
void LCD_write(u16 VAL)
{
  LCD_CS_CLR;
  DATAOUT(VAL);
  LCD_WR_CLR;
  LCD_WR_SET;
  LCD_CS_SET;
}

u16 LCD_read(void)
{
  u16 data;
  LCD_CS_CLR;
  LCD_RD_CLR;
  delay_us(1);//延时1us
  data = DATAIN;
  LCD_RD_SET;
  LCD_CS_SET;
  return data;
}
```

The FSMC test program is implemented as shown below:

```
u16 LCD_read(void)
{
  vu16 data;  //防止被优化
  data=LCD->LCD_RAM;
  return data;
}

/*****************************************************************************
 * @name       :void LCD_WR_REG(u16 data)
 * @date       :2018-08-09
 * @function   :Write an 16-bit command to the LCD screen
 * @parameters :data:Command value to be written
 * @retvalue   :None
*****************************************************************************/
void LCD_WR_REG(u16 data)
{
  LCD->LCD_REG=data; //写入要写的寄存器序号
}

/*****************************************************************************
 * @name       :void LCD_WR_DATA(u16 data)
 * @date       :2018-08-09
 * @function   :Write an 16-bit data to the LCD screen
 * @parameters :data:data value to be written
 * @retvalue   :None
*****************************************************************************/
void LCD_WR_DATA(u16 data)
{
  LCD->LCD_RAM=data; //写入要写的数据
}
```

Both 8- and 16-bit command writes and 8- and 16-bit data writes and reads are implemented.

B. C51 test program parallel port communication code implementation

The relevant code is implemented in the LCD.c file as shown below:

```
void LCD_write(u8 HVAL,u8 LVAL)
{
  LCD_CS = 0;
  LCD_WR = 0;
  LCD_DataPortH = HVAL;
  LCD_DataPortL = LVAL;
  LCD_WR = 1;
  LCD_CS = 1;
}

u16 LCD_read(void)
{
  u16 d;
  LCD_CS = 0;
  LCD_RD = 0;
  delay_us(1); //delay 1 us
  d = LCD_DataPortH;
  d = (d<<8)|LCD_DataPortL;
  LCD_RD = 1;
  LCD_CS = 1;
  return d;
}
```

Implemented 8-bit and 16-bit commands and 8-bit and 16-bit data write and read.

Common software

This set of test examples requires the display of Chinese and English, symbols and pictures, so the modulo software is used. There are two types of modulo software: Image2Lcd and PCtoLCD2002. Here is only the setting of the modulo software for the test program.

The **PCtoLCD2002** modulo software settings are as follows:

Dot matrix format select Dark code

the modulo mode select the progressive mode

Take the model to choose the direction (high position first)

Output number system selects hexadecimal number

Custom format selection C51 format

The specific setting method is as follows:

http://www.lcdwiki.com/Chinese_and_English_display_modulo_settings

Image2Lcd modulo software settings are shown below:

The Image2Lcd software needs to be set to horizontal, left to right, top to bottom, and low position to the front scan mode.