3.97inch 16BIT Module MRB3973 User Manual

Product Description

The product is a 3.97-inch TFT LCD module ,it has the 800x480 resolution and supports 16BIT RGB 65K color display, the internal driver IC is NT35510 and use 16-bit parallel port communication. The module includes LCD display, resistive touch screen, SD card slot and PCB backplane. It can be Insert directly the STM32 series development board and supports SD card expansion.

Product Features

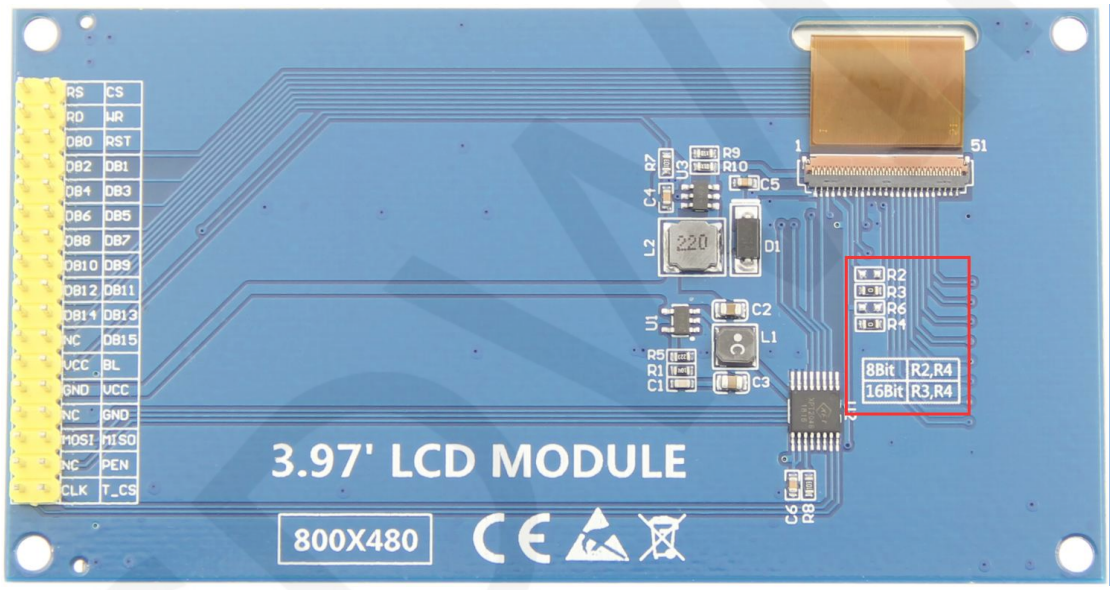
- 3.97-inch color screen, support 16BIT RGB 65K color display, display rich colors
- 800x480 resolution for clear display
- 16-bit parallel bus transmission for fast transfer speed
- Supports ALIENTEK STM32 Mini, Elite, WarShip, Explorer, and Apollo development boards direct plug-in use
- Support for touch function
- Support SD card function expansion
- 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	RGB 65K color
SKU	MRB3973
Screen Size	3.97(inch)
Type	TFT
Driver IC	NT35510
Resolution	800*480 (Pixel)

Module Interface	16Bit parallel interface
Active Area	51.84x86.40(mm)
Module PCB Size	59.18x111.51(mm)
Operating Temperature	-10℃~60℃
Storage Temperature	-20℃~70℃
Operating Voltage	3.3V / 5V
Power Consumption	TBD
Product Weight	TBD

Interface Description



Picture1. Module Pin silk screen picture

Note:

1. **The module hardware supports 8-bit and 16-bit parallel port data bus mode switching (as shown by the red box in Picture 1 above), as follows:**

   A. **Solder R3 and R4 with 0Ω resistor or short circuit directly, and disconnect R2 and R6: select 16-bit data bus mode (default), use DB0~DB15 data pin**

   B. **Solder R2 and R4 with 0Ω resistor or short circuit directly, and disconnect R3 and R6: select 8-bit data bus mode, use DB0~D7 data pin**

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 purchase a module with PCB backplane, VCC/VDD can be connected to 5V or 3.3V (module has integrated ultra low dropout 5V to 3V circuit), if you buy a bare screen LCD, remember only Can connect to 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
15	DB9	
16	DB10	
17	DB11	
18	DB12	
19	DB13	
20	DB14	
21	DB15	
22	NC	Undefined, reserved
23	BL	LCD backlight control pin(High level light)
24	VCC	Module power positive pin (module has integrated voltage regulator IC, so the power supply can be connected to 5V or 3.3V)
25	VCC	
26	GND	Module power ground pin
27	GND	
28	NC	Undefined, reserved
29	MISO	Touch screen SPI bus data input pin
30	MOSI	Touch screen SPI bus data output pin
31	PEN	Touch screen interrupt detection pin (Low level when a touch occurs)
32	NC	Undefined, reserved
33	T_CS	Touch screen IC chip select control pin(Low level enable)
34	CLK	Touch screen SPI bus clock control pin

Hardware Configuration

The LCD module hardware circuit comprises four parts: an LCD display control circuit, a resistive touch screen sampling circuit, an SD card interface circuit, and a backlight control circuit.

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

The resistive touch screen sampling circuit is used for detecting a touch event, performing

AD conversion on the touch data, and transmitting touch coordinate values.

SD card control circuit is used for SD card function expansion, controlling SD card identification, reading and writing.

The backlight control circuit is used to control the backlight to be on and off.

working principle

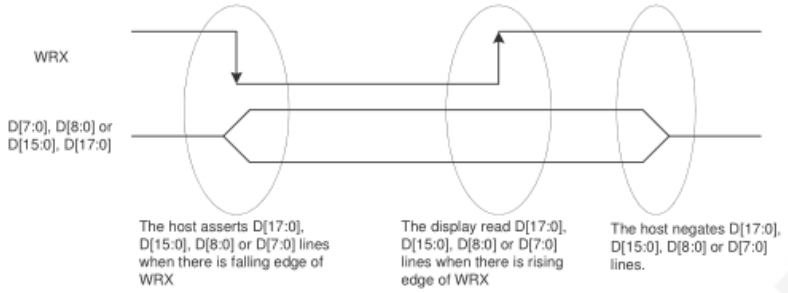
1. Introduction to NT35510 Controller

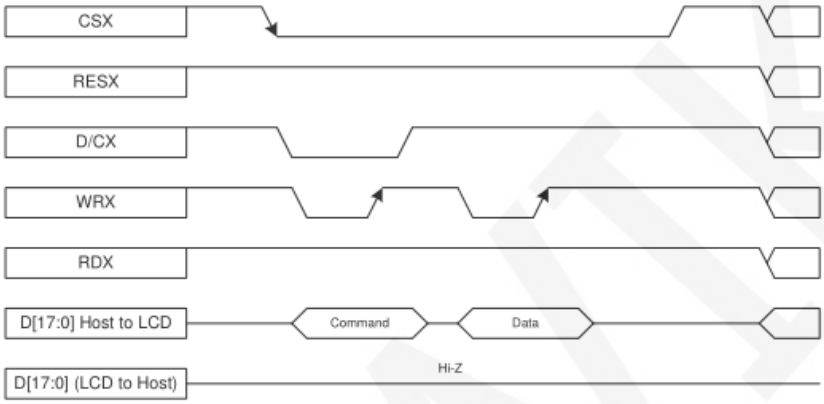
The NT35510 controller is a driver IC for TFT LCDs that supports multiple resolutions: 480*864, 480*854, 480*800, 480*720, 480*640, and 480*1024 (expanded memory required). It has a memory of 1244160 bytes and can support MDDI interface, MIPI interface, 16-bit/18-bit/24-bit RGB interface, 8-bit/16-bit/18-bit/24-bit parallel port, SPI and I2C interface. It supports 8, 65K, 262K and 16.7M RGB color display, display color is very rich, while supporting rotating display and scroll display and video playback, display a variety of ways.

This module uses a 16-bit parallel port to transmit data and 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 the rows and columns, and the incrementing and decreasing direction is determined by the scanning mode. The NT35510 display method is performed by setting the address and then setting the color value.

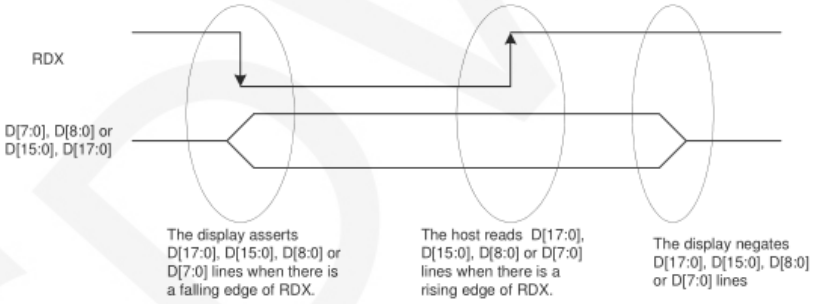
2. Introduction to parallel port communication

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

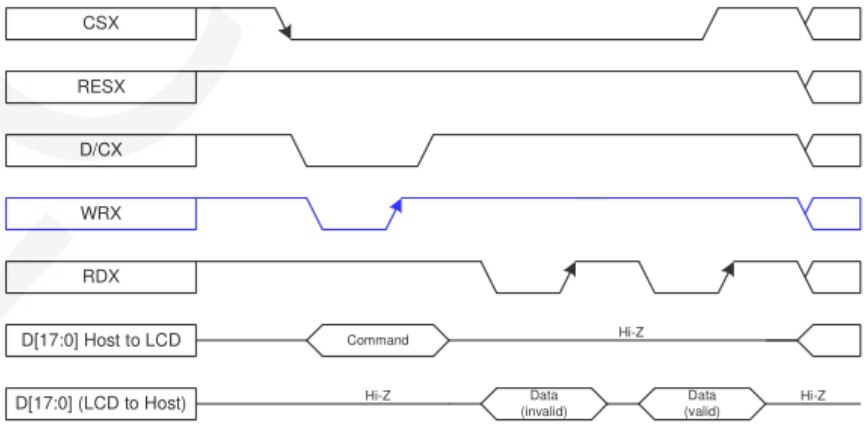




The timing of the parallel port communication read mode is shown in the figure below:



Note: RDX is an unsynchronized signal (It can be stopped).



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. C51 instructions

Wiring instructions:

See the interface description for pin assignments.

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	**NC**	No need to connect
23	**BL**	P32
24	**VCC**	3.3V/5V
25	**VCC**	3.3V/5V
26	**GND**	GND
27	**GND**	GND
28	**NC**	No need to connect
29	**MISO**	P35
30	**MOSI**	P34
31	**PEN**	P40
32	**NC**	No need to connect

33	T_CS	P37
34	CLK	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	NC	No need to connect
23	BL	3.3V
24	VCC	3.3V/5V
25	VCC	3.3V/5V
26	GND	GND

27	GND	GND
28	NC	No need to connect
29	MISO	No need to connect
30	MOSI	No need to connect
31	PEN	No need to connect
32	NC	No need to connect
33	T_CS	No need to connect
34	CLK	No need to connect

Note:

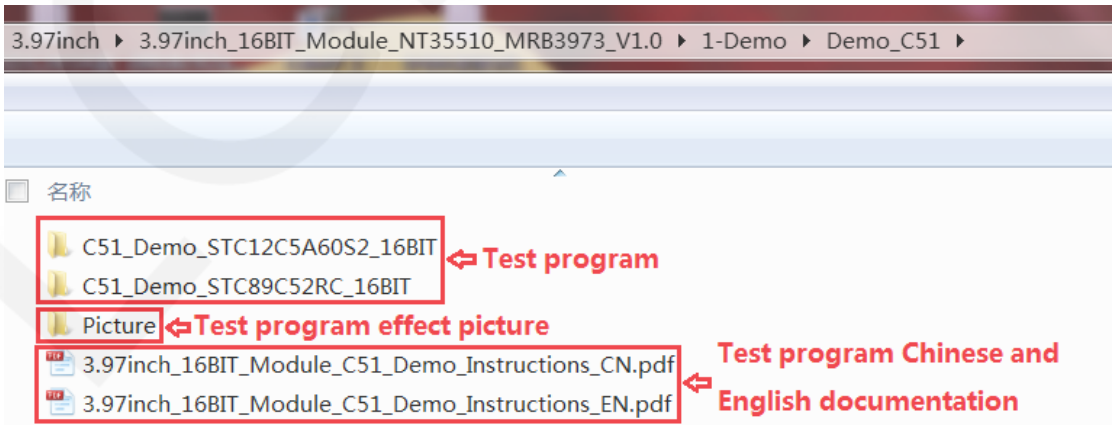
1. **Since the STC89C52RC microcontroller does not have a push-pull output function, the backlight control pin needs to be connected to a 3.3V power supply to be properly lit.**
2. **Since the STC89C52RC microcontroller's Flash capacity is too small (less than 25KB), the program with touch function cannot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

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;

2. STM32 instructions

Wiring instructions:

See the interface description for pin assignments.

Note:

2. **If using the IO simulation test program, you need to connect the module to the development board (except MiniSTM32) with DuPont line;**
3. **If you use the FSMC test program, you can plug the module directly into the TFTLCD slot of the development board;**
4. **The STM32F103RCT6 microcontroller does not have the FSMC function, but the module can also be plugged directly into the MiniSTM32 development board;**

STM32F103RCT6 microcontroller test program directly insert instructions		
Number	**Module Pin**	**Corresponding to MiniSTM32 development board directly insert pin**
1	**CS**	PC9
2	**RS**	PC8
3	**WR**	PC7
4	**RD**	PC6
5	**RST**	PC4
6	**DB0**	PB0
7	**DB1**	PB1
8	**DB2**	PB2
9	**DB3**	PB3
10	**DB4**	PB4

11	DB5	PB5
12	DB6	PB6
13	DB7	PB7
14	DB8	PB8
15	DB9	PB9
16	DB10	PB10
17	DB11	PB11
18	DB12	PB12
19	DB13	PB13
20	DB14	PB14
21	DB15	PB15
22	NC	No need to connect
23	BL	PC10
24	VCC	3.3V/5V
25	VCC	3.3V/5V
26	GND	GND
27	GND	GND
28	NC	No need to connect
29	MISO	PC2
30	MOSI	PC3
31	PEN	PC1
32	NC	No need to connect
33	T_CS	PC13
34	CLK	PC0

STM32F103ZET6 microcontroller test program wiring instructions

Number	Module Pin	Corresponding to Elite STM32 development board wiring pin	
		IO Simulation	FSMC
1	CS	PC9	PG12
2	RS	PC8	PG0

3	**WR**	PC7	PD5
4	**RD**	PC6	PD4
5	**RST**	PC4	reset pin
6	**DB0**	PF0	PD14
7	**DB1**	PF1	PD15
8	**DB2**	PF2	PD0
9	**DB3**	PF3	PD1
10	**DB4**	PF4	PE7
11	**DB5**	PF5	PE8
12	**DB6**	PF6	PE9
13	**DB7**	PF7	PE10
14	**DB8**	PF8	PE11
15	**DB9**	PF9	PE12
16	**DB10**	PF10	PE13
17	**DB11**	PF11	PE14
18	**DB12**	PF12	PE15
19	**DB13**	PF13	PD8
20	**DB14**	PF14	PD9
21	**DB15**	PF15	PD10
22	**NC**	No need to connect	No need to connect
23	**BL**	PC10	PB0
24	**VCC**	3.3V/5V	3.3V/5V
25	**VCC**	3.3V/5V	3.3V/5V
26	**GND**	GND	GND
27	**GND**	GND	GND
28	**NC**	No need to connect	No need to connect
29	**MISO**	PC2	PB2
30	**MOSI**	PC3	PF9
31	**PEN**	PC1	PF10
32	**NC**	No need to connect	No need to connect
33	**T_CS**	PC13	PF11
34	**CLK**	PC0	PB1

STM32F103ZET6 microcontroller test program wiring instructions				
Number	Module Pin	Corresponding to WarShip STM32 development board wiring pin		
		IO Simulation	FSMC	
			V2	V3
1	CS	PC9	PG12	PG12
2	RS	PC8	PG0	PG0
3	WR	PC7	PD5	PD5
4	RD	PC6	PD4	PD4
5	RST	PC4	reset pin	reset pin
6	DB0	PF0	PD14	PD14
7	DB1	PF1	PD15	PD15
8	DB2	PF2	PD0	PD0
9	DB3	PF3	PD1	PD1
10	DB4	PF4	PE7	PE7
11	DB5	PF5	PE8	PE8
12	DB6	PF6	PE9	PE9
13	DB7	PF7	PE10	PE10
14	DB8	PF8	PE11	PE11
15	DB9	PF9	PE12	PE12
16	DB10	PF10	PE13	PE13
17	DB11	PF11	PE14	PE14
18	DB12	PF12	PE15	PE15
19	DB13	PF13	PD8	PD8
20	DB14	PF14	PD9	PD9
21	DB15	PF15	PD10	PD10
22	NC	No need to connect	No need to connect	No need to connect
23	BL	PC10	PB0	PB0
24	VCC	3.3V/5V	3.3V/5V	3.3V/5V
25	VCC	3.3V/5V	3.3V/5V	3.3V/5V

26	**GND**	GND	GND	GND
27	**GND**	GND	GND	GND
28	**NC**	No need to connect	No need to connect	No need to connect
29	**MISO**	PC2	PF8	PB2
30	**MOSI**	PC3	PF9	PF9
31	**PEN**	PC1	PF10	PF10
32	**NC**	No need to connect	No need to connect	No need to connect
33	**T_CS**	PC13	PB2	PF11
34	**CLK**	PC0	PB1	PB1

STM32F407ZGT6 microcontroller test program wiring instructions			
Number	Module Pin	**Corresponding to Explorer STM32F4 development board wiring pin**	
		IO Simulation	**FSMC**
1	**CS**	PC9	PG12
2	**RS**	PC8	PF12
3	**WR**	PC7	PD5
4	**RD**	PC6	PD4
5	**RST**	PC4	reset pin
6	**DB0**	PG0	PD14
7	**DB1**	PG1	PD15
8	**DB2**	PG2	PD0
9	**DB3**	PG3	PD1
10	**DB4**	PG4	PE7
11	**DB5**	PG5	PE8
12	**DB6**	PG6	PE9
13	**DB7**	PG7	PE10
14	**DB8**	PG8	PE11
15	**DB9**	PG9	PE12
16	**DB10**	PG10	PE13
17	**DB11**	PG11	PE14

18	**DB12**	PG12	PE15
19	**DB13**	PG13	PD8
20	**DB14**	PG14	PD9
21	**DB15**	PG15	PD10
22	**NC**	No need to connect	No need to connect
23	**BL**	PC10	PB15
24	**VCC**	3.3V/5V	3.3V/5V
25	**VCC**	3.3V/5V	3.3V/5V
26	**GND**	GND	GND
27	**GND**	GND	GND
28	**NC**	No need to connect	No need to connect
29	**MISO**	PF2	PB2
30	**MOSI**	PF3	PF11
31	**PEN**	PF1	PB1
32	**NC**	No need to connect	No need to connect
33	**T_CS**	PF13	PC13
34	**CLK**	PF0	PB0

STM32F429IGT6、STM32F767IGT6、STM32H743IIT6 microcontroller test program wiring instructions			
Number	**Module Pin**	**Corresponding to Apollo STM32F4/F7 development board wiring pin**	
		IO Simulation	**FSMC**
1	**CS**	PC9	PD7
2	**RS**	PC8	PD13
3	**WR**	PC7	PD5
4	**RD**	PC6	PD4
5	**RST**	PC4	reset pin
6	**DB0**	PE0	PD14
7	**DB1**	PE1	PD15
8	**DB2**	PE2	PD0
9	**DB3**	PE3	PD1

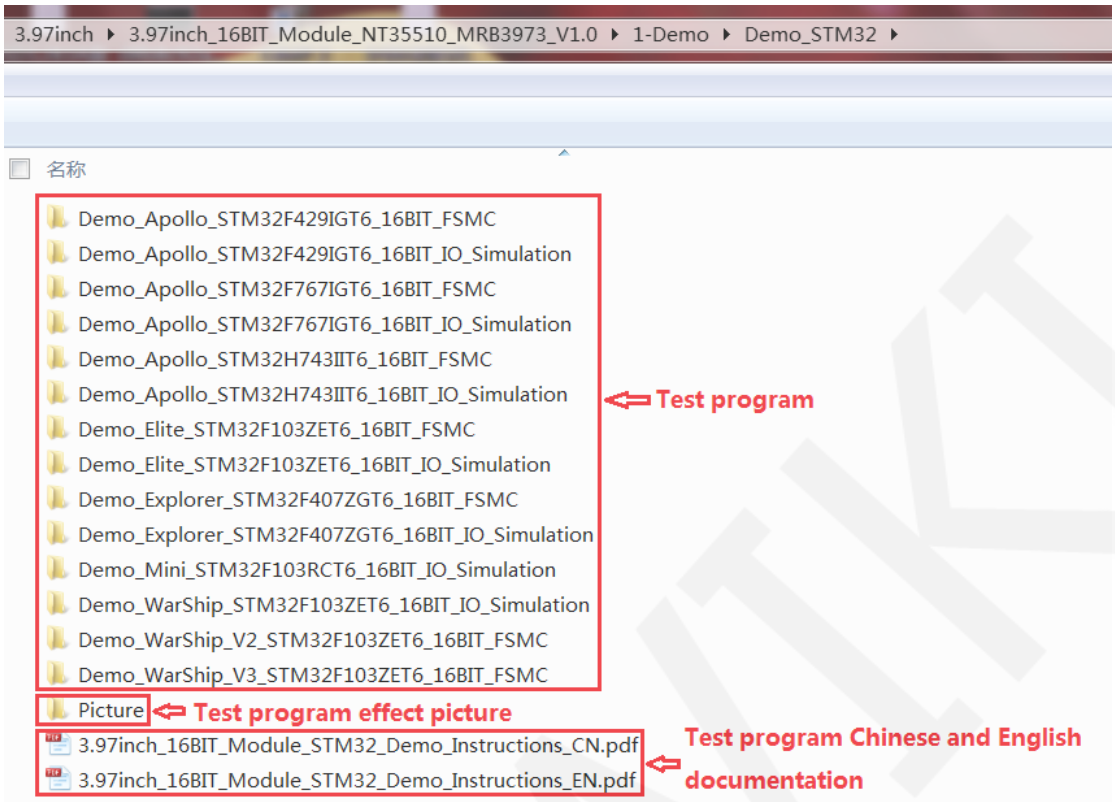
10	**DB4**	PE4	PE7
11	**DB5**	PE5	PE8
12	**DB6**	PE6	PE9
13	**DB7**	PE7	PE10
14	**DB8**	PE8	PE11
15	**DB9**	PE9	PE12
16	**DB10**	PE10	PE13
17	**DB11**	PE11	PE14
18	**DB12**	PE12	PE15
19	**DB13**	PE13	PD8
20	**DB14**	PE14	PD9
21	**DB15**	PE15	PD10
22	**NC**	No need to connect	No need to connect
23	**BL**	PC10	PB5
24	**VCC**	3.3V/5V	3.3V/5V
25	**VCC**	3.3V/5V	3.3V/5V
26	**GND**	GND	GND
27	**GND**	GND	GND
28	**NC**	No need to connect	No need to connect
29	**MISO**	PH11	PG3
30	**MOSI**	PH12	PI3
31	**PEN**	PH10	PH7
32	**NC**	No need to connect	No need to connect
33	**T_CS**	PH13	PI8
34	**CLK**	PH9	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:

(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 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

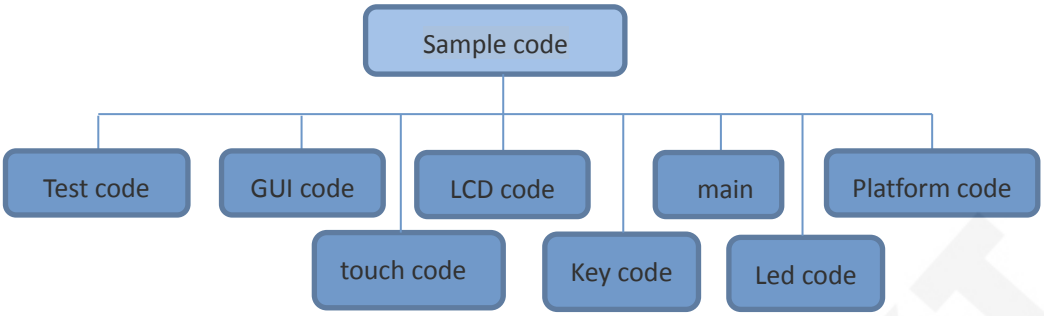
D. If the LCD module displays characters and graphics normally, the program runs successfully;

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;

Touch screen related operations are included in the touch code;

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. 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 touch screen related GPIO definition is placed in the touch.h file, as shown below (take the STC12C5A60S2 microcontroller test program as an example):

```
//IO连接
sfr    P4     = 0xC0;
sbit DCLK     =     P3^6;
sbit TCS          =     P3^7;
sbit DIN        =     P3^4;
sbit DOUT       =     P3^5;
sbit Penirq     =     P4^0;    //检测触摸屏响应信号
```

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.

B. STM32 test program GPIO definition description

STM32 FSMC test program lcd screen GPIO is defined by FSMC bus. The related definition method can refer to FSMC bus description data. Its GPIO definition is placed in lcd.h file 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
//如果使用官方库函数定义下列底层，速度将会下降到14帧每秒，建议采用我司推荐方法
//以下IO定义直接操作寄存器，快速IO操作，刷屏速率可以达到28帧每秒！

//LCD地址结构体
typedef struct
{
  vu16 LCD_REG;
  vu16 LCD_RAM;
} 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
//注意设置时STM32内部会右移一位对其!
#define LCD_BASE        ((u32)(0x6C000000 | 0x000007FE))
#define LCD             ((LCD_TypeDef *) LCD_BASE)
```

STM32 IO simulation test program lcd screen GPIO definition is placed in the lcd.h

file, as shown below (take STM32F103RCT6 microcontroller IO simulation test program as an example):

```
//////////////////////////////////////////////////////////////////
//-----------------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
```

Data parallel port pin definition needs to select a complete set of GPIO port groups, such as PB, when transferring data, it is convenient to operate.Other pins can be defined as any free GPIO.

The GPIO definition related to the STM32 touch screen is placed in the touch.h file as shown below (take the STM32F103RCT6 microcontroller IO simulation test program as an example):

```
//与触摸屏芯片连接引脚
//与触摸屏芯片连接引脚
#define PEN  PCin(1)    //PC1  INT
#define DOUT PCin(2)     //PC2  MISO     PC2--PB14
#define TDIN PCout(3)    //PC3  MOSI     PC3--PB15
#define TCLK PCout(0)    //PC0  SCLK   PC0--PB13
#define TCS  PCout(13)   //PC13 CS
```

If you use the IO simulation test program, you can modify the values in the parentheses. All pin definitions can be modified and can be defined as any other free GPIO.

If the FSMC test program is used, the touch screen GPIO cannot be modified because the GPIO pins on the development board are fixed by the in-line connection.

3. Parallel port communication code implementation

A. 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.

B. STM32 test program parallel port communication code implementation

The STM32 test program parallel port communication code is implemented in the LCD.c file.

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
}
```

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 IO analog test program implements 8- and 16-bit commands and 8- and 16-bit data write and read.

The FSMC test program implements 16-bit commands and 16-bit data write and

read.

4. touch screen calibration instructions

A. C51 test program touch screen calibration instructions

The C51 touch screen calibration needs to execute the Touch_Adjust test item (only available in the STC12C5A60S2 test program), as shown below:

```
//循环进行各项测试
while(1)
{
  main_test();     //测试主界面
  Test_Color();      //简单刷屏填充测试
  Test_FillRec();    //GUI矩形绘图测试
  Test_Circle();     //GUI画圆测试
  Test_Triangle();    //GUI三角形填充测试
  English_Font_test();//英文字体示例测试
  Chinese_Font_test();//中文字体示例测试
  Pic_test();      //图片显示示例测试
  Rotate_Test();
//不使用触摸或者模块本身不带触摸，请屏蔽下面触摸屏测试
  Touch_Test();    //触摸屏手写测试
//需要触摸校准时，请将触摸手写测试屏蔽，将下面触摸校准测试项打开
//  Touch_Adjust();  //触摸校准
}
```

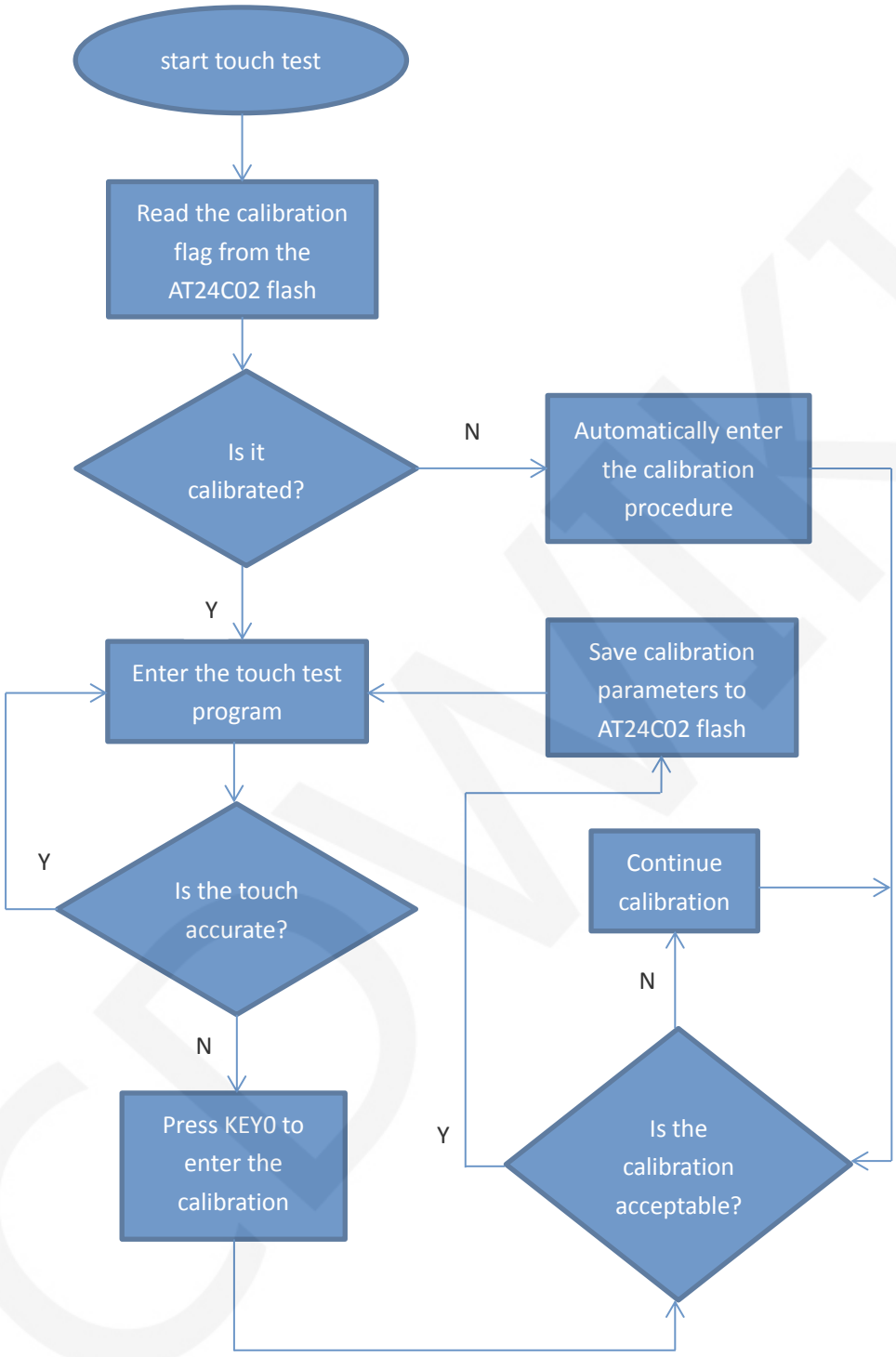
After the touch calibration is passed, you need to save the calibration parameters displayed on the screen in the touch.c file, as shown below:

```
//***因触摸屏批次不同等原因，默认的校准参数值可能会引起触摸
u16 vx=11738,vy=7736;   //比例因子，此值除以1000之后表示多少
u16 chx=3905,chy=246;//默认像素点坐标为0时的AD起始值
//***因触摸屏批次不同等原因，默认的校准参数值可能会引起触摸
```

B. STM32 test program touch screen calibration instructions

The STM32 touch screen calibration program automatically recognizes whether calibration is required or manually enters calibration by pressing a button.

It is included in the touch screen test item. The calibration mark and calibration parameters are saved in the AT24C02 flash. If necessary, read from the flash. The calibration process is as shown below:


start touch test
Read the calibration flag from the AT24C02 flash
Is it calibrated?
N
Automatically enter the calibration procedure
Y
Enter the touch test program
Save calibration parameters to AT24C02 flash
Y
Is the touch accurate?
Continue calibration
N
N
Press KEY0 to enter the calibration
Y
Is the calibration acceptable?

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.