3.95inch Arduino 8&16BIT Module
MAR3953
User Manual

Product Description

The product is a 3.95-inch TFT LCD module with 480x320 resolution, 16BIT RGB 65K color display, internal drive IC ST7796S, 8-bit and 16-bit parallel port communication, and 8-bit parallel port communication. The module includes LCD display, resistive touch screen, SD card slot and PCB backplane. It supports SD card expansion and can be directly plugged into the Arduino MEGA2560 development board. It can also be used on C51 and STM32 platforms.

Product Features

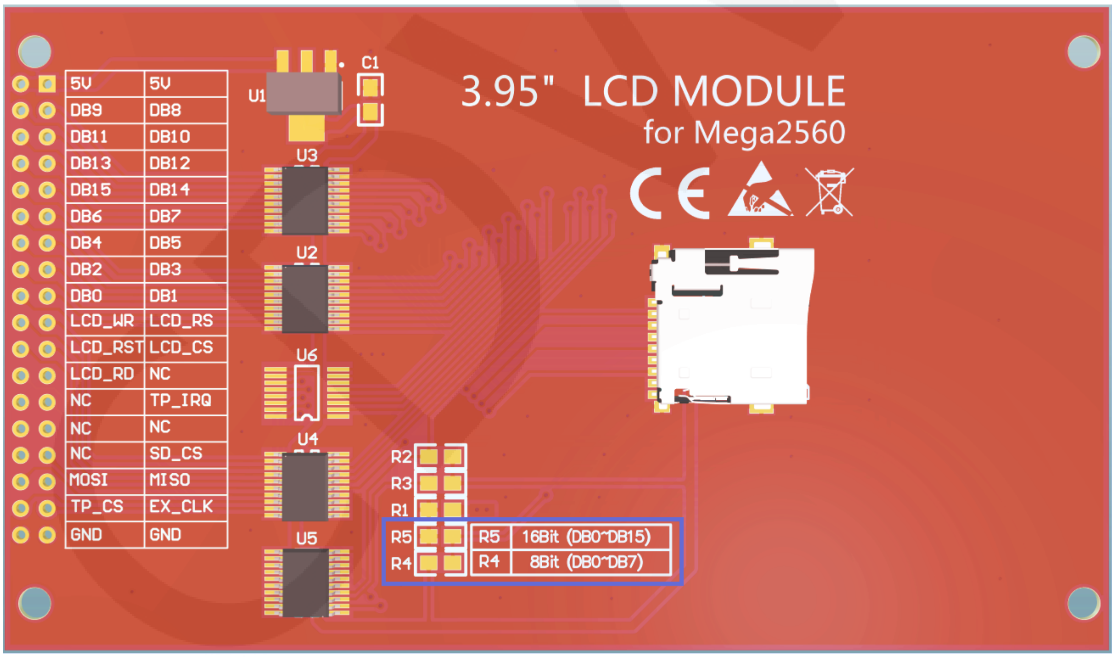
- 3.95-inch color screen, support 16BIT RGB 65K color display, display rich colors
- 480x320 resolution for clear display
- Supports 8-bit and 16-bit parallel bus transmission with fast transfer speed
- On-board 5V/3.3V level-shifting IC compatible with 5V/3.3V operating voltage
- Support Arduino Mage2560 for direct plug-in use
- Support for touch function
- Support SD card function extension
- Provide Arduino libraries and rich sample programs
- Available on C51 and STM32 platforms with a rich sample program
- Military-grade process standards, long-term stable work
- Provide underlying driver technical support

Product Parameters

Name	Description
Display Color	RGB 65K color
SKU	MAR3953
Screen Size	3.95(inch)
Type	TFT

Driver IC	ST7796S
Resolution	480*320 (Pixel)
Module Interface	8Bit or 16Bit parallel interface
Active Area	83.52x55.68(mm)
Module PCB Size	61.54x105.69 (mm)
Back Light	6 chip HighLight white LEDs
Operating Temperature	-10℃~60℃
Storage Temperature	-20℃~70℃
Operating Voltage	3.3V / 5V
Power Consumption	TBD
Product Weight	TBD

Interface Description



Picture1. Module Pin silkscreen picture

Note:

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

A. Solder R5 with 0Ω resistor or short circuit directly, and disconnect R4: select 16-bit data bus mode (default), use DB0~DB15 data pin

B. Solder R4 with 0Ω resistor or short circuit directly, and disconnect R5: select 8-bit data bus mode, use DB0~DB7 data pin

Important Note:

1. The following pin numbers 1~30 refer to the module pin number of our company with PCB backplane. 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: LCD_CS is 20 feet on our module, which may be x feet on different sizes of bare screen.
2. About VCC supply voltage: If you purchase a module with PCB backplane, VCC/VDD power supply needs to be connected to 5V (module has integrated ultra low dropout 5V to 3.3V circuit), if you buy a bare screen LCD screen, remember to only connect 3.3V.
3. About backlight voltage: Modules with PCB backplane are connected to 3.3V, no need to manually access. If you are buying a bare screen, the LEDA is connected to 3.0V-3.3V, and the LEDKx can be grounded.

Number	Module Pin	Pin Description
1	5V	Power pin
2	DB0	Data bus low 8-bit pin
3	DB1	
4	DB2	
5	DB3	
6	DB4	
7	DB5	
8	DB6	
9	DB7	
10	DB8	Data bus high 8-bit pin
11	DB9	

12	**DB10**	
13	**DB11**	
14	**DB12**	
15	**DB13**	
16	**DB14**	
17	**DB15**	
18	**LCD_RS**	LCD register / data selection pin
19	**LCD_WR**	LCD write control pin
20	**LCD_CS**	LCD chip select control pin
21	**LCD_RST**	LCD reset control pin
22	**LCD_RD**	LCD read control pin
23	**NC**	Undefined, reserved
24	**TP_IRQ**	Touch screen interrupt control pin
25	**SD_CS**	Extended reference: SD card select pin
26	**MISO**	SPI bus input pin
27	**MOSI**	SPI bus output pin
28	**EX_CLK**	SPI bus clock pin
29	**TP_CS**	Touch screen chip select pin
30	**GND**	Power ground pin

Hardware Configuration

The LCD module hardware circuit comprises five parts: an LCD display control circuit, a level shift circuit, an SD card control circuit, a touch screen control circuit, and an 8-bit and 16-bit data bus mode switching circuit.

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

Level shifting circuit for 5V/3.3V conversion, making the module compatible with 3.3V/5V power supply.

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

The touch screen control circuit is used to control touch screen interrupt acquisition, data sampling, AD conversion, data transmission, and the like.

The 8-bit and 16-bit data bus mode switching circuits are used to switch the data bus type (8-bit mode and 16-bit mode). For details, see the red box in Picture 1 above or refer to the module circuit schematic.

working principle

1. Introduction to ST7796S Controller

The ST7796S is a single-chip controller for 262 K color TFT-LCDs. It supports a maximum resolution of 320*480 and has a GRAM of 345600 bytes. It also supports 8-bit, 9-bit, 16-bit, and 18-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. ST7796S also supports 65K, 262K, 16M 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 ST7796S 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 ST7796S display method is performed by setting the address and then setting the color value.

Introduction to parallel port communication

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

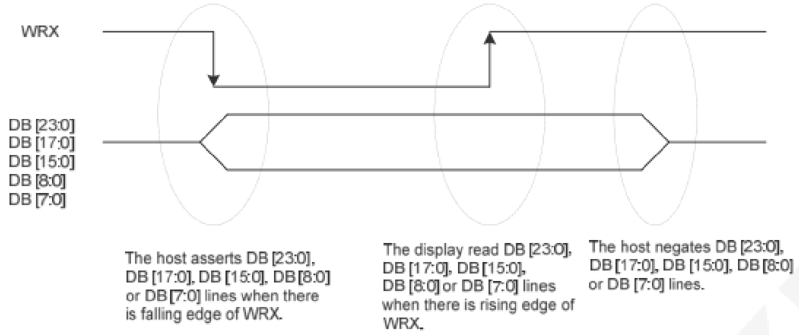
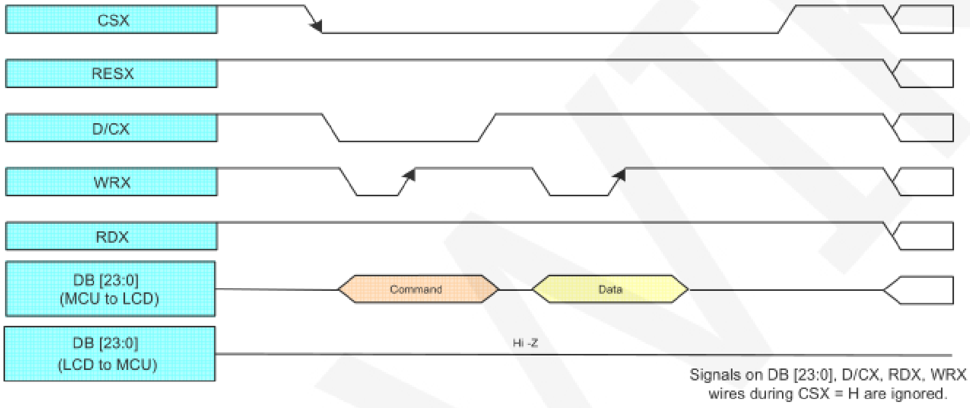

WRX
DB [23:0]
DB [17:0]
DB [15:0]
DB [8:0]
DB [7:0]
The host asserts DB [23:0], DB [17:0], DB [15:0], DB [8:0] or DB [7:0] lines when there is falling edge of WRX.
The display read DB [23:0], DB [17:0], DB [15:0], DB [8:0] or DB [7:0] lines when there is rising edge of WRX.
The host negates DB [23:0], DB [17:0], DB [15:0], DB [8:0] or DB [7:0] lines.

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.



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

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.

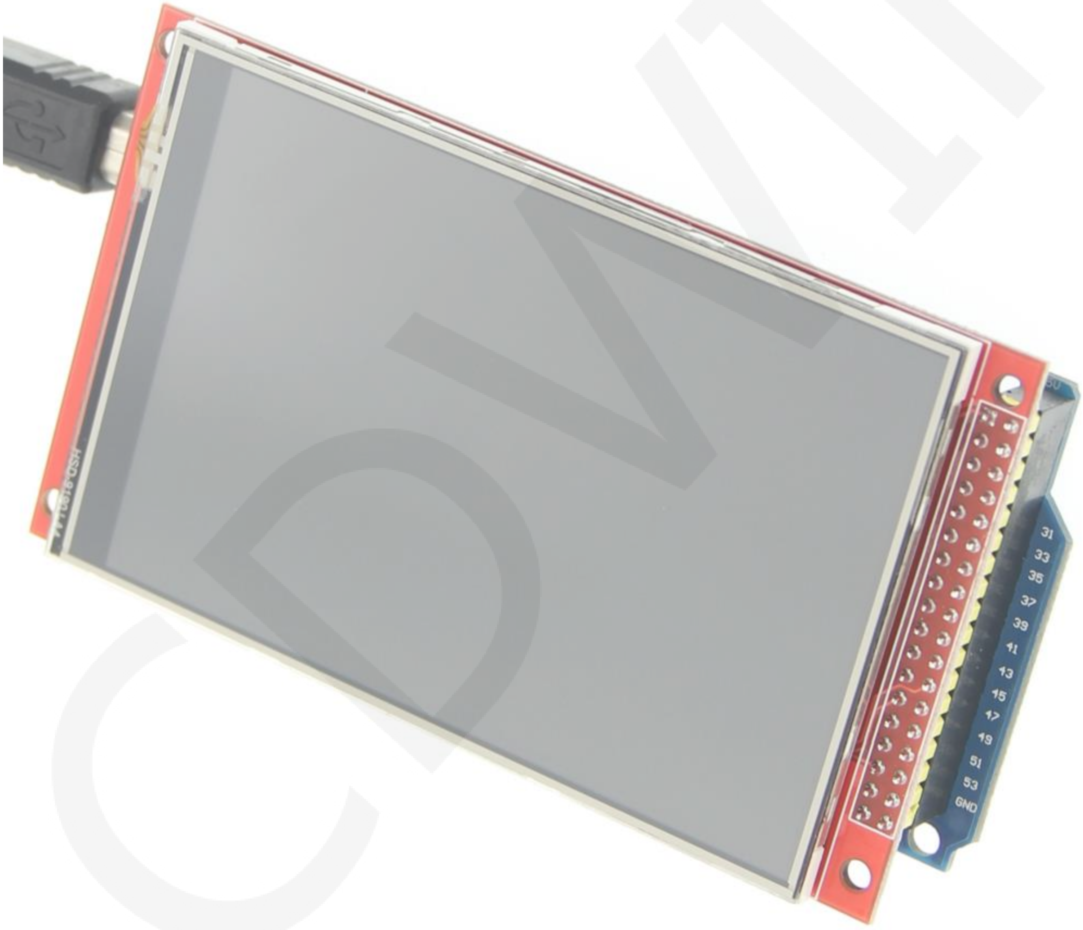
Instructions for use

1. Arduino instructions

Wiring instructions:

See the interface description for pin assignments.

This module can be directly inserted into the Arduino UNO and Mega2560, no need to manually wire, as shown below:



Mega2560 directly inserted picture

Direct insertion instructions for Arduino MEGA2560 microcontroller test program pins

Number	Module Pin	Corresponding to MEGA2560 development board direct plug pins	
		8-bit mode	16-bit mode
1	5V	5V	
2	DB0	37	
3	DB1	36	
4	DB2	35	
5	DB3	34	
6	DB4	33	
7	DB5	32	
8	DB6	31	
9	DB7	30	
10	DB8	not used	22
11	DB9		23
12	DB10		24
13	DB11		25
14	DB12		26
15	DB13		27
16	DB14		28
17	DB15		29
18	LCD_RS	38	
19	LCD_WR	39	
20	LCD_CS	40	
21	LCD_RST	41	
22	LCD_RD	43	
23	NC	not used	
24	TP_IRQ	44	
25	SD_CS	48	
26	MISO	50	
27	MOSI	51	
28	TP_CS	53	

29	**EX_CLK**	52
30	**GND**	GND

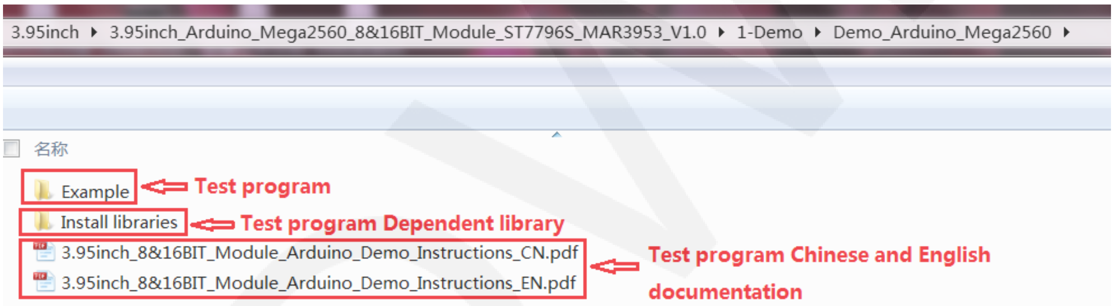
Operating Steps:

A. Insert the LCD module directly into the Arduino MCU according to the above wiring instructions, and power on;

B. Copy the dependent libraries in the Install libraries directory of the test package to the libraries folder of the Arduino project directory (if you do not need to depend on the libraries, you do not need to copy them);

C. Open the directory where the Arduino test program is located and select the example you want to test, as shown below:

(Please refer to the test program description document in the test package for the test program description)


3.95inch ▸ 3.95inch_Arduino_Mega2560_8&16BIT_Module_ST7796S_MAR3953_V1.0 ▸ 1-Demo ▸ Demo_Arduino_Mega2560 ▸
名称
Example
Test program
Install libraries
Test program Dependent library
3.95inch_8&16BIT_Module_Arduino_Demo_Instructions_CN.pdf
3.95inch_8&16BIT_Module_Arduino_Demo_Instructions_EN.pdf
Test program Chinese and English documentation

D. Open the selected sample project, compile and download.

The specific operation methods for the Arduino test program relying on library copy, compile and download are as follows:

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

E. 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.

STC89C52RC microcontroller test program wiring instructions

Number	Module Pin	Corresponding to STC89 development board wiring pin	
		8-bit mode	16-bit mode
1	5V	5V	
2	DB0	P30	
3	DB1	P31	
4	DB2	P32	
5	DB3	P33	
6	DB4	P34	
7	DB5	P35	
8	DB6	P36	
9	DB7	P37	
10	DB8	no need to connect	P20
11	DB9		P21
12	DB10		P22
13	DB11		P23
14	DB12		P24
15	DB13		P25
16	DB14		P26
17	DB15		P27
18	LCD_RS	P12	
19	LCD_WR	P11	
20	LCD_CS	P13	
21	LCD_RST	P14	
22	LCD_RD	P10	
23	NC	no need to connect	
24	TP_IRQ	no need to connect (cannot test touch)	
25	SD_CS	no need to connect	
26	MISO	no need to connect (cannot test touch)	
27	MOSI	no need to connect (cannot test touch)	

28	**TP_CS**	no need to connect (cannot test touch)
29	**EX_CLK**	no need to connect (cannot test touch)
30	**GND**	GND

STC12C5A60S2 microcontroller test program wiring instructions

Number	**Module Pin**	**Corresponding to STC12 development board wiring pin**	
		8-bit mode	**16-bit mode**
1	**5V**	5V	
2	**DB0**	P00	
3	**DB1**	P01	
4	**DB2**	P02	
5	**DB3**	P03	
6	**DB4**	P04	
7	**DB5**	P05	
8	**DB6**	P06	
9	**DB7**	P07	
10	**DB8**	no need to connect	P20
11	**DB9**		P21
12	**DB10**		P22
13	**DB11**		P23
14	**DB12**		P24
15	**DB13**		P25
16	**DB14**		P26
17	**DB15**		P27
18	**LCD_RS**	P12	
19	**LCD_WR**	P11	
20	**LCD_CS**	P13	
21	**LCD_RST**	P33	
22	**LCD_RD**	P10	
23	**NC**	no need to connect	

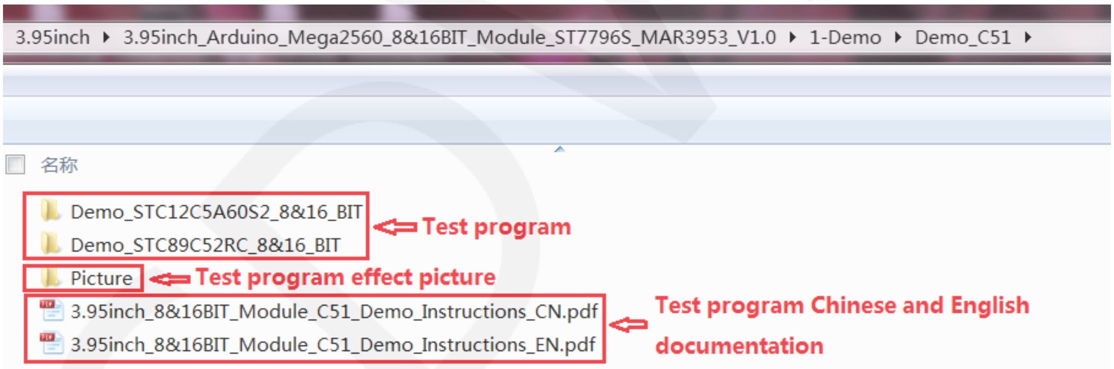
24	**TP_IRQ**	P40
25	**SD_CS**	no need to connect
26	**MISO**	P35
27	**MOSI**	P34
28	**TP_CS**	P37
29	**EX_CLK**	P36
30	**GND**	GND

Operating Steps:

A. Connect the LCD module and the C51 MCU according to the above wiring instructions, and power on;

B. Open the directory where the C51 test program is located and select the example to be tested, as shown below:

(Please refer to the test program description document for test program description)



3.95inch ▸ 3.95inch_Arduino_Mega2560_8&16BIT_Module_ST7796S_MAR3953_V1.0 ▸ 1-Demo ▸ Demo_C51 ▸

□ 名称

Demo_STC12C5A60S2_8&16_BIT
Demo_STC89C52RC_8&16_BIT
Picture
3.95inch_8&16BIT_Module_C51_Demo_Instructions_CN.pdf
3.95inch_8&16BIT_Module_C51_Demo_Instructions_EN.pdf

Test program
Test program effect picture
Test program Chinese and English documentation

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

3. STM32 instructions

Wiring instructions:

See the interface description for pin assignments.

STM32F103RCT6 microcontroller test program wiring instructions

Number	Module Pin	Corresponding to MiniSTM32 development board wiring pin	
		8-bit mode	16-bit mode
1	5V	5V	
2	DB0	PB0	
3	DB1	PB1	
4	DB2	PB2	
5	DB3	PB3	
6	DB4	PB4	
7	DB5	PB5	
8	DB6	PB6	
9	DB7	PB7	
10	DB8	no need to connect	PB8
11	DB9		PB9
12	DB10		PB10
13	DB11		PB11
14	DB12		PB12
15	DB13		PB13
16	DB14		PB14
17	DB15		PB15
18	LCD_RS	PC8	
19	LCD_WR	PC7	
20	LCD_CS	PC9	
21	LCD_RST	PC10	
22	LCD_RD	PC6	
23	NC	no need to connect	
24	TP_IRQ	PC1	
25	SD_CS	no need to connect	
26	MISO	PC2	
27	MOSI	PC3	

28	**TP_CS**	PC13
29	**EX_CLK**	PC0
30	**GND**	GND

STM32F103ZET6 microcontroller test program wiring instructions

Number	Module Pin	Corresponding to Elite STM32 development board wiring pin	
		8-bit mode	16-bit mode
1	**5V**	5V	
2	**DB0**	PF0	
3	**DB1**	PF1	
4	**DB2**	PF2	
5	**DB3**	PF3	
6	**DB4**	PF4	
7	**DB5**	PF5	
8	**DB6**	PF6	
9	**DB7**	PF7	
10	**DB8**	no need to connect	PF8
11	**DB9**		PF9
12	**DB10**		PF10
13	**DB11**		PF11
14	**DB12**		PF12
15	**DB13**		PF13
16	**DB14**		PF14
17	**DB15**		PF15
18	**LCD_RS**	PC8	
19	**LCD_WR**	PC7	
20	**LCD_CS**	PC9	
21	**LCD_RST**	PC10	
22	**LCD_RD**	PC6	
23	**NC**	no need to connect	
24	**TP_IRQ**	PC1	

25	**SD_CS**	no need to connect
26	**MISO**	PC2
27	**MOSI**	PC3
28	**TP_CS**	PC13
29	**EX_CLK**	PC0
30	**GND**	GND

STM32F407ZGT6 microcontroller test program wiring instructions

Number	Module Pin	Corresponding to Explorer STM32F4 development board wiring pin	
		8-bit mode	16-bit mode
1	**5V**	5V	
2	**DB0**	PG0	
3	**DB1**	PG1	
4	**DB2**	PG2	
5	**DB3**	PG3	
6	**DB4**	PG4	
7	**DB5**	PG5	
8	**DB6**	PG6	
9	**DB7**	PG7	
10	**DB8**	no need to connect	PG8
11	**DB9**		PG9
12	**DB10**		PG10
13	**DB11**		PG11
14	**DB12**		PG12
15	**DB13**		PG13
16	**DB14**		PG14
17	**DB15**		PG15
18	**LCD_RS**	PC8	
19	**LCD_WR**	PC7	
20	**LCD_CS**	PC9	
21	**LCD_RST**	PC10	

22	LCD_RD	PC6
23	NC	no need to connect
24	TP_IRQ	PC1
25	SD_CS	no need to connect
26	MISO	PC2
27	MOSI	PC3
28	TP_CS	PC13
29	EX_CLK	PC0
30	GND	GND

STM32F429IGT6、STM32F767IGT6、STM32H743IIT6 microcontroller test program wiring instructions

Number	Module Pin	Corresponding to Apollo STM32F4/F7 development board wiring pin	
		8-bit mode	16-bit mode
1	5V	5V	
2	DB0/NC	PE0	
3	DB1/NC	PE1	
4	DB2/NC	PE2	
5	DB3/NC	PE3	
6	DB4/NC	PE4	
7	DB5/NC	PE5	
8	DB6/NC	PE6	
9	DB7/NC	PE7	
10	DB8	no need to connect	PE8
11	DB9		PE9
12	DB10		PE10
13	DB11		PE11
14	DB12		PE12
15	DB13		PE13
16	DB14		PE14
17	DB15		PE15

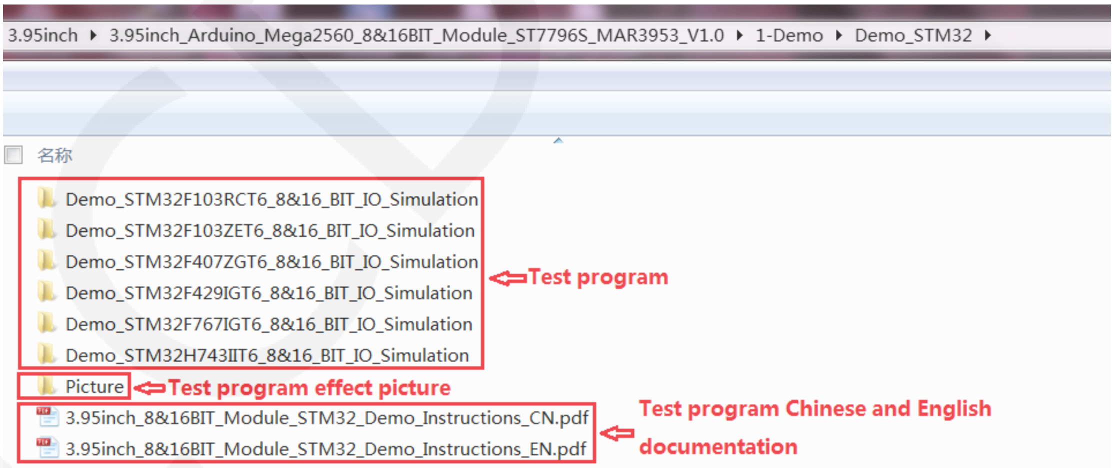
18	**LCD_RS**	PC8
19	**LCD_WR**	PC7
20	**LCD_CS**	PC9
21	**LCD_RST**	PC10
22	**LCD_RD**	PC6
23	**NC**	no need to connect
24	**TP_IRQ**	PH10
25	**SD_CS**	no need to connect
26	**MISO**	PH11
27	**MOSI**	PH12
28	**TP_CS**	PH13
29	**EX_CLK**	PH9
30	**GND**	GND

Operating Steps:

A. Connect the LCD module and the STM32 MCU according to the above wiring instructions, and power on;

B. Open the directory where the STM32 test program is located and select the example to be tested, as shown below:

(Please refer to the test program description document for test program description)



3.95inch ▸ 3.95inch_Arduino_Mega2560_8&16BIT_Module_ST7796S_MAR3953_V1.0 ▸ 1-Demo ▸ Demo_STM32 ▸

□ 名称

Demo_STM32F103RCT6_8&16_BIT_IO_Simulation
Demo_STM32F103ZET6_8&16_BIT_IO_Simulation
Demo_STM32F407ZGT6_8&16_BIT_IO_Simulation
Demo_STM32F429IGT6_8&16_BIT_IO_Simulation
Demo_STM32F767IGT6_8&16_BIT_IO_Simulation
Demo_STM32H743IIT6_8&16_BIT_IO_Simulation
Picture
3.95inch_8&16BIT_Module_STM32_Demo_Instructions_CN.pdf
3.95inch_8&16BIT_Module_STM32_Demo_Instructions_EN.pdf

Test program

Test program effect picture

Test program Chinese and English documentation

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

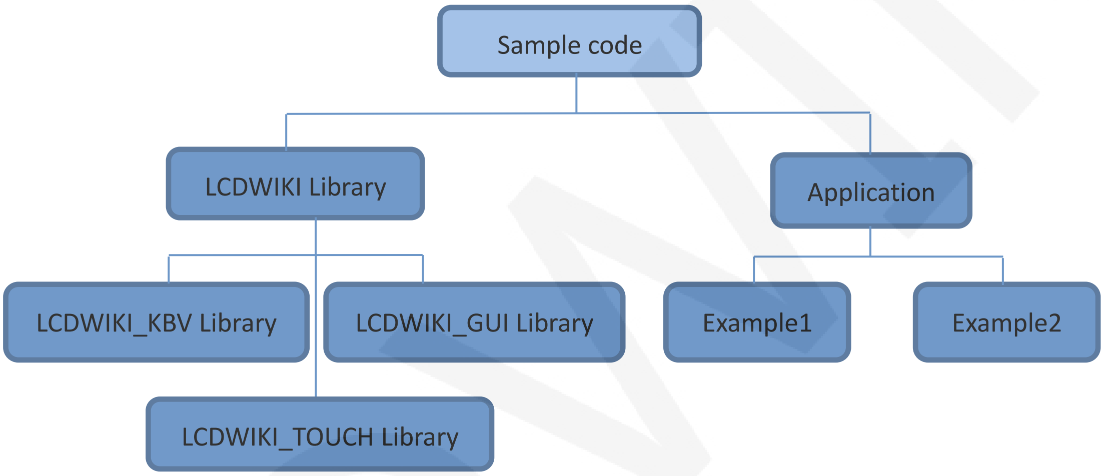
D. If the LCD module displays characters and graphics normally, the program runs successfully;

Software Description

1. Code Architecture

A. Arduino code architecture description

The code architecture is shown below:



Arduino's test program code consists of two parts: the LCDWIKI library and application code.

The LCDWIKI library contains three parts: LCDWIKI_KBV library, LCDWIKI_GUI library, and LCDWIKI_TOUCH library.

The application contains several test examples, each with different test content;

LCDWIKI_KBV is the underlying library, which is associated with hardware. It is mainly responsible for operating registers, including hardware module initialization, data and command transmission, pixel coordinates and color settings, display mode configuration, etc;

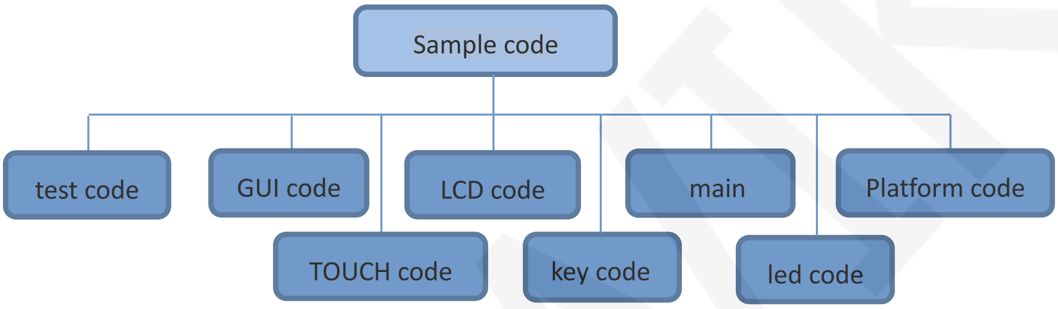
LCDWIKI_GUI is the middle layer library, which is responsible for drawing graphics and displaying characters using the API provided by the underlying library;

LCDWIKI_TOUCH is the underlying library of touch screens, mainly responsible for touch interrupt detection, touch data sampling and AD conversion, and touch data transmission.

The application is to use the API provided by the LCDWIKI library to write some test examples and implement Some aspect of the test function;

B. 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;

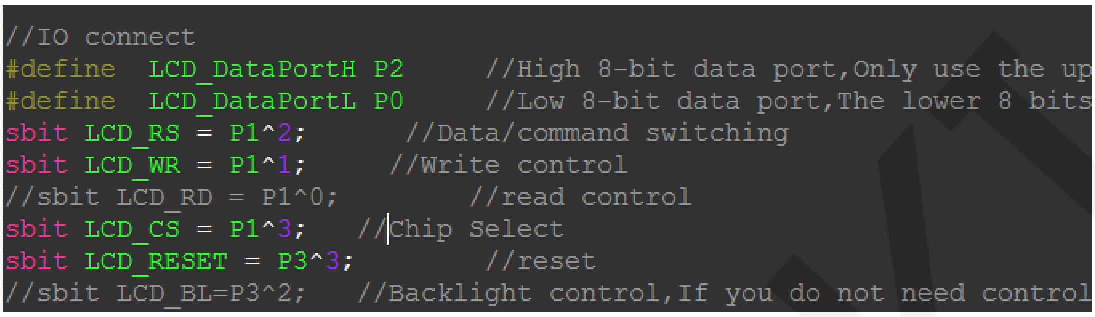
2. GPIO definition description

A. Arduino test program GPIO definition description

The module is plugged into the Arduino Mage2560, so it is not allowed to modify the GPIO port definition.

B. C51 test program GPIO definition description

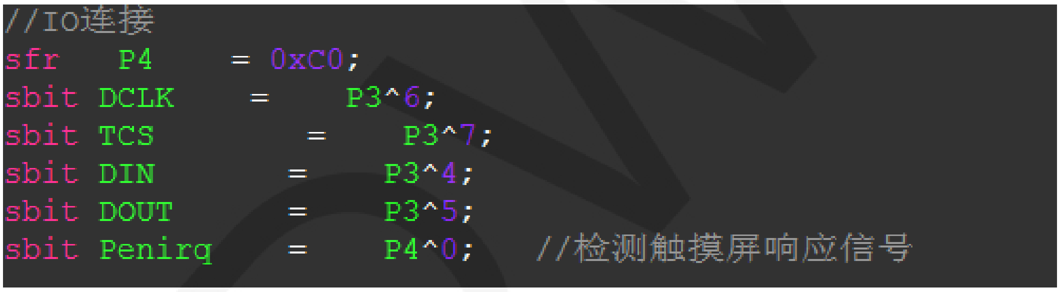
The C51 test program GPIO definition is placed in the lcd.h file as shown below(Take the STC12C5A60S2 microcontroller test program as an example):



```
//IO connect
#define  LCD_DataPortH P2      //High 8-bit data port,Only use the up
#define  LCD_DataPortL P0      //Low 8-bit data port,The lower 8 bits
sbit LCD_RS = P1^2;        //Data/command switching
sbit LCD_WR = P1^1;       //Write control
//sbit LCD_RD = P1^0;          //read control
sbit LCD_CS = P1^3;    //Chip Select
sbit LCD_RESET = P3^3;         //reset
//sbit LCD_BL=P3^2;    //Backlight control,If you do not need control
```

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 GPIO port definition is placed in touch.h, as shown below (only 12C5A60S2 can test touch)



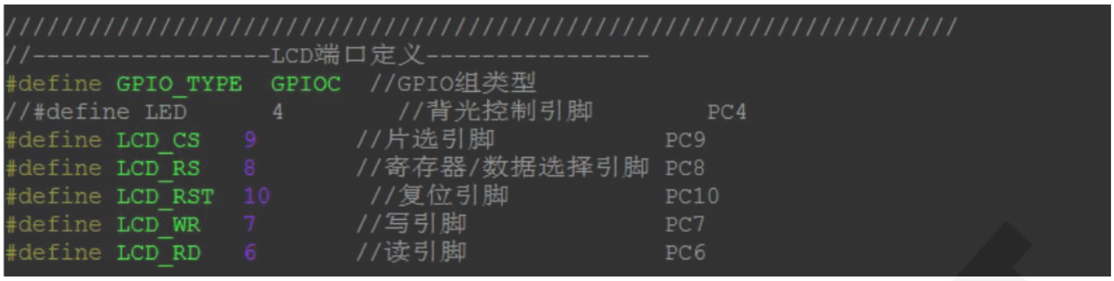
```
//IO连接
sfr   P4     = 0xC0;
sbit DCLK     =    P3^6;
sbit TCS         =    P3^7;
sbit DIN        =    P3^4;
sbit DOUT       =    P3^5;
sbit Penirq     =    P4^0;   //检测触摸屏响应信号
```

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

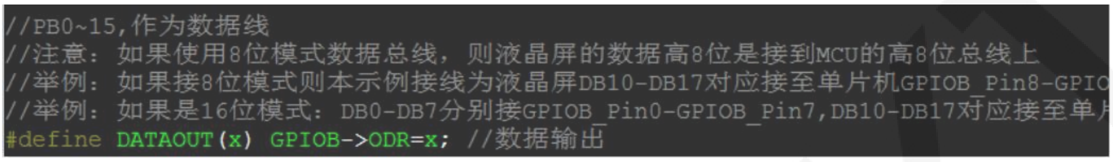
If the microcontroller does not have a P4 GPIO group, you can define penirq as another GPIO.

C. STM32 test program GPIO definition description

STM32 IO simulation test program lcd screen GPIO definition is placed in the lcd.h file, as shown below (take STM32F103RCT6 test program as an example



```
/////////////////////////////////////////////////////////////////////
//-----------------LCD端口定义----------------
#define GPIO_TYPE  GPIOC  //GPIO组类型
//#define LED      4        //背光控制引脚        PC4
#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   10       //复位引脚            PC10
#define LCD_WR    7       //写引脚               PC7
#define LCD_RD    6       //读引脚               PC6
```

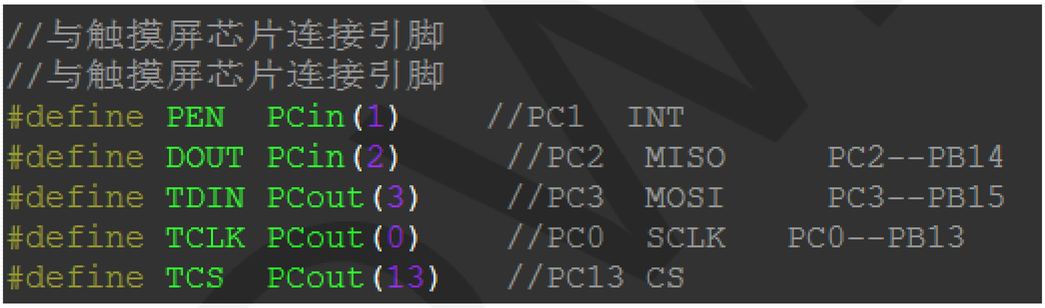


```
//PB0~15,作为数据线
//注意: 如果使用8位模式数据总线, 则液晶屏的数据高8位是接到MCU的高8位总线上
//举例: 如果接8位模式则本示例接线为液晶屏DB10-DB17对应接至单片机GPIOB_Pin8-GPIO
//举例: 如果是16位模式: DB0-DB7分别接GPIOB_Pin0-GPIOB_Pin7,DB10-DB17对应接至单片
#define DATAOUT(x) GPIOB->ODR=x; //数据输出
```

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 touch screen GPIO port is defined in the touch.h file as shown below (take the STM32F103RCT6 test program as an example)



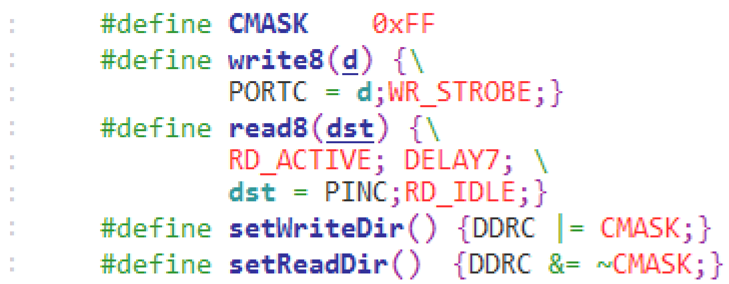
```
//与触摸屏芯片连接引脚
//与触摸屏芯片连接引脚
#define PEN  PCin(1)      //PC1  INT
#define DOUT PCin(2)       //PC2  MISO      PC2--PB14
#define TDIN PCout(3)      //PC3  MOSI      PC3--PB15
#define TCLK PCout(0)      //PC0  SCLK    PC0--PB13
#define TCS  PCout(13)     //PC13 CS
```

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.

3. Parallel port communication code implementation

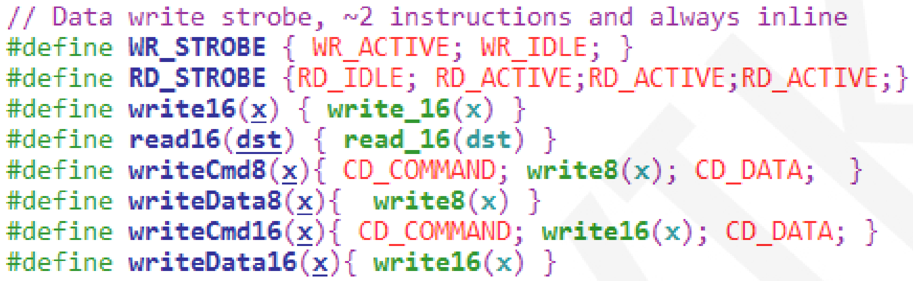
A. Arduino test program parallel port communication code implementation

If the 8-bit mode related code is used in the mcu_8bit_magic.h file of the LCDWIKI_KBV library, as shown below:

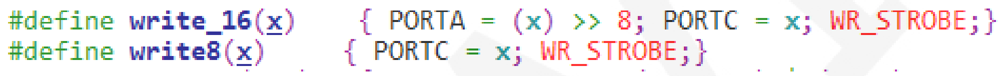


```
#define CMASK    0xFF
#define write8(d) {\
        PORTC = d;WR_STROBE;}
#define read8(dst) {\
        RD_ACTIVE; DELAY7; \
        dst = PINC;RD_IDLE;}
#define setWriteDir() {DDRC |= CMASK;}
#define setReadDir()  {DDRC &= ~CMASK;}
```

If the 16-bit mode related code is used in the mcu_16bit_magic.h file of the LCDWIKI_KBV library, as shown below:



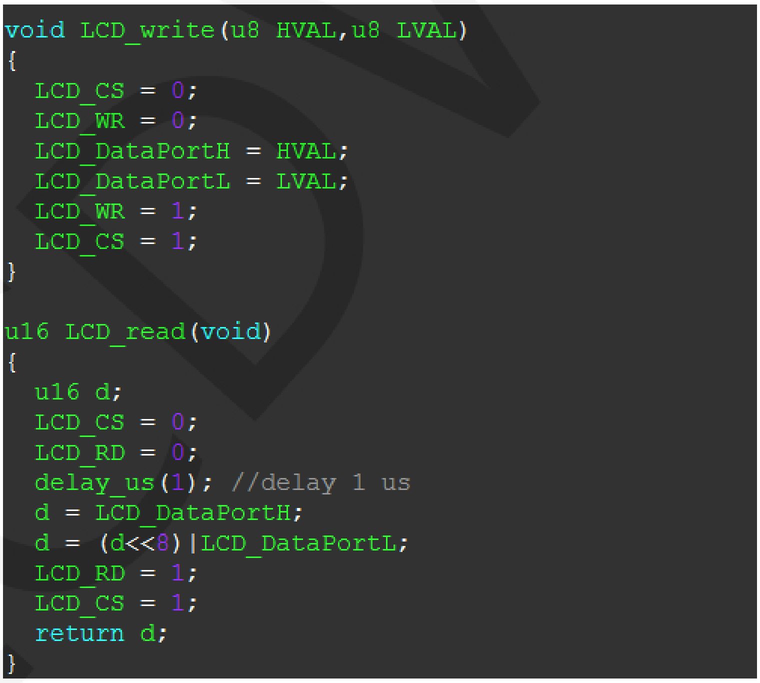
```
// Data write strobe, ~2 instructions and always inline
#define WR_STROBE { WR_ACTIVE; WR_IDLE; }
#define RD_STROBE {RD_IDLE; RD_ACTIVE;RD_ACTIVE;RD_ACTIVE;}
#define write16(x) { write_16(x) }
#define read16(dst) { read_16(dst) }
#define writeCmd8(x){ CD_COMMAND; write8(x); CD_DATA;  }
#define writeData8(x){  write8(x) }
#define writeCmd16(x){ CD_COMMAND; write16(x); CD_DATA; }
#define writeData16(x){ write16(x) }
```



```
#define write_16(x)    { PORTA = (x) >> 8; PORTC = x; WR_STROBE;}
#define write8(x)     { PORTC = x; WR_STROBE;}
```

B. C51 test program parallel port communication code implementation

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



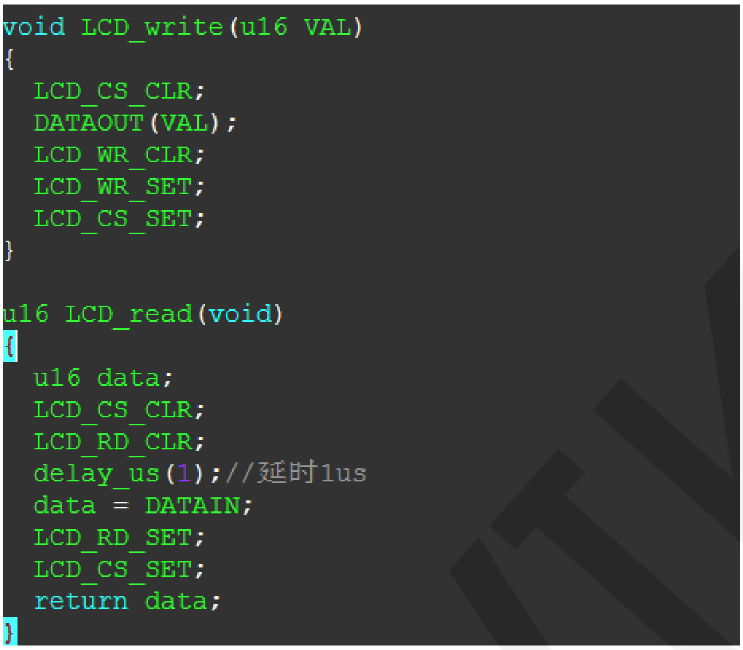
```
void LCD_write(u8 HVAL,u8 LVAL)
{
  LCD_CS = 0;
  LCD_WR = 0;
  LCD_DataPortH = HVAL;
  LCD_DataPortL = LVAL;
  LCD_WR = 1;
  LCD_CS = 1;
}

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

C. STM32 test program parallel port communication code implementation

The STM32 test program parallel port communication code is implemented in the

LCD.c file.The IO simulation test program is implemented as shown below:



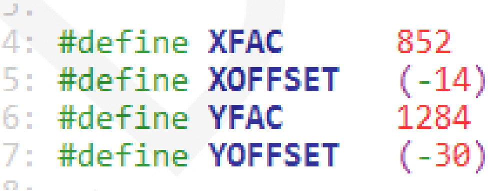
```
void LCD_write(u16 VAL)
{
  LCD_CS_CLR;
  DATAOUT(VAL);
  LCD_WR_CLR;
  LCD_WR_SET;
  LCD_CS_SET;
}

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;
}
```

Both 8 and 16-bit commands and 8, 16-bit data transfers are implemented.

4. touch screen calibration instructions

A. Arduino test program touch screen calibration instructions

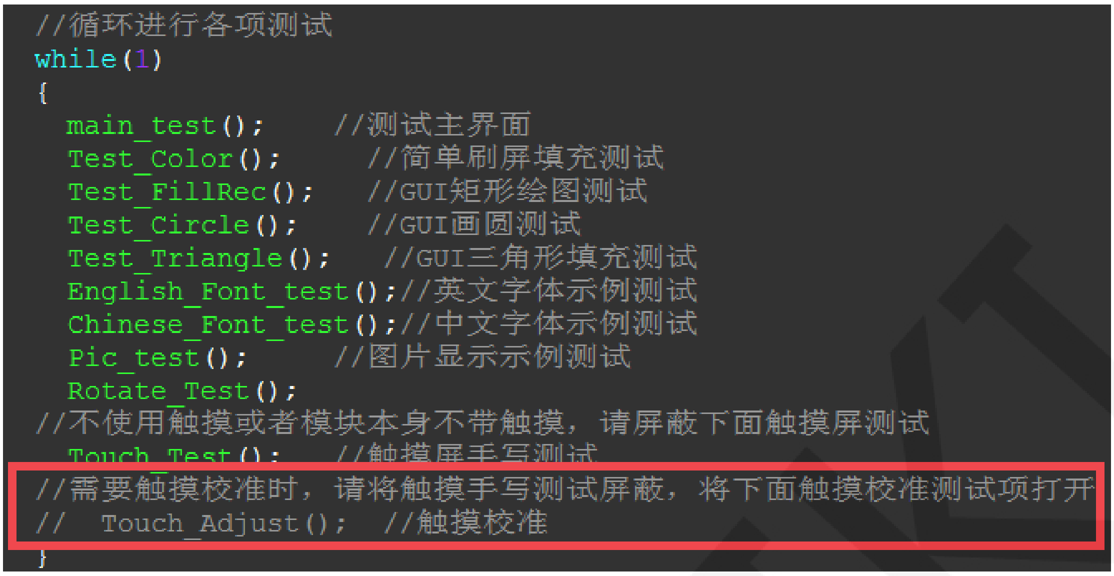
Arduino touch screen calibration needs to run the touch_screen_calibration program first, and then calibrate according to the prompts. After the calibration is passed, the calibration parameters displayed on the screen need to be written into the cali_para.h file of the LCDWIKI_TOUCH library, as shown below:



```
#define XFAC       852
#define XOFFSET    (-14)
#define YFAC       1284
#define YOFFSET    (-30)
```

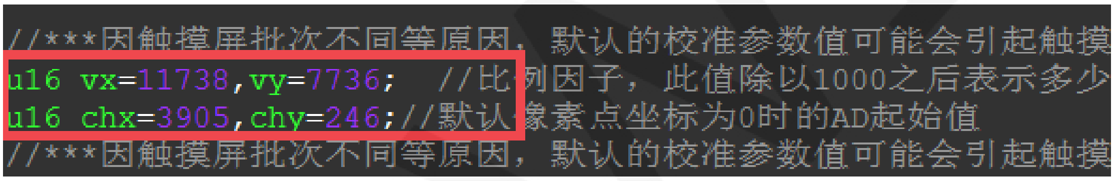
B. C51 test program touch screen calibration instructions

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:

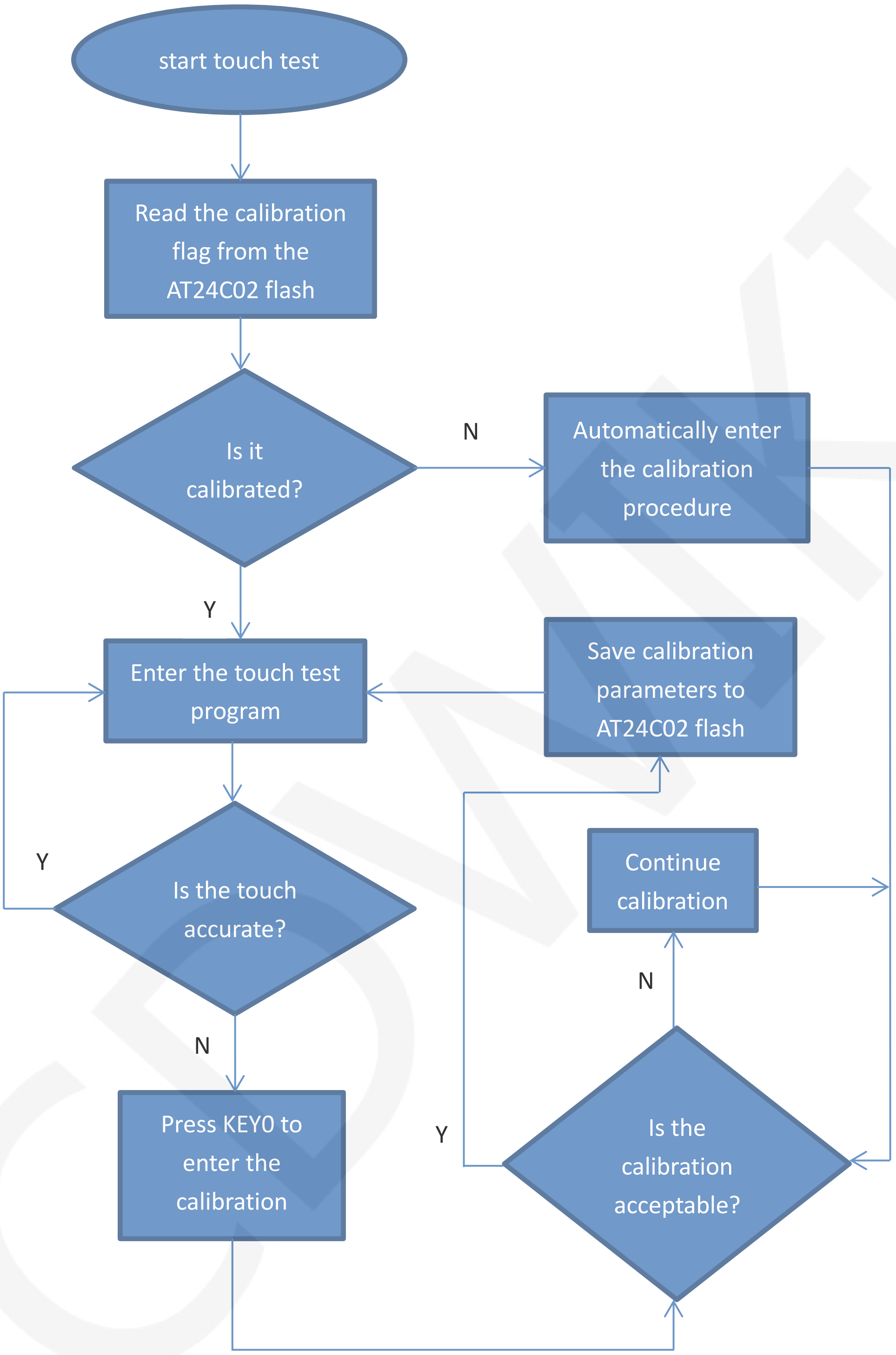


```
//***因触摸屏批次不同等原因，默认的校准参数值可能会引起触摸
u16 vx=11738,vy=7736;  //比例因子，此值除以1000之后表示多少
u16 chx=3905,chy=246;//默认像素点坐标为0时的AD起始值
//***因触摸屏批次不同等原因，默认的校准参数值可能会引起触摸
```

C. 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)

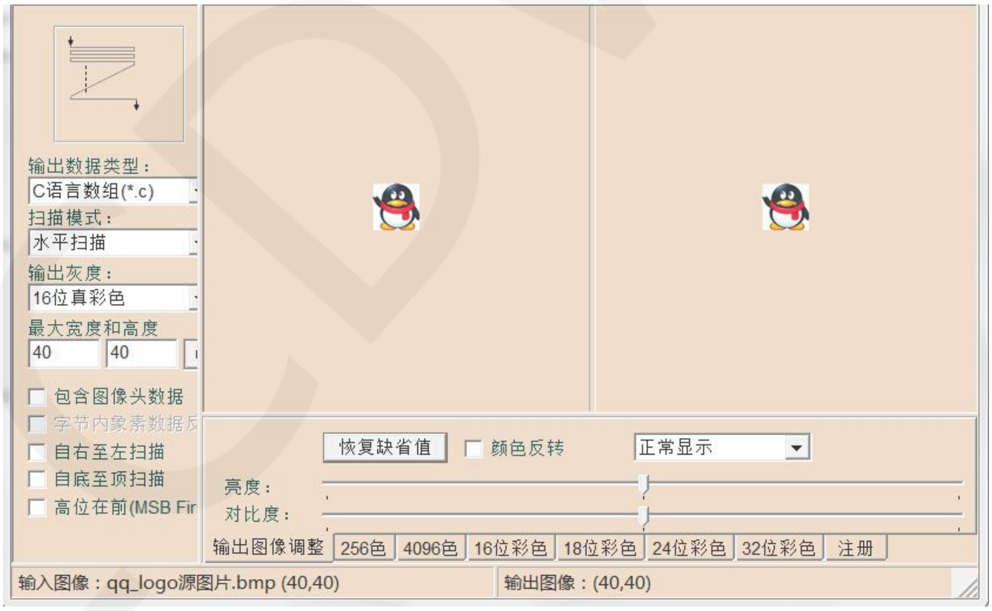
Output number system selects hexadecimal number

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:



输出数据类型：C语言数组(*.c)
扫描模式：水平扫描
输出灰度：16位真彩色
最大宽度和高度 40 40

□ 包含图像头数据
□ 字节内象素数据反
□ 自右至左扫描
□ 自底至顶扫描
□ 高位在前(MSB Fir

□ 颜色反转 正常显示

输入图像：qq_logo源图片.bmp (40,40)
输出图像：(40,40)

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