3.8inch Arduino 8BIT Module MAR3803 User Manual

Product Description

The Arduino Mega2560 module is a 3.8-inch TFT LCD module with 480x320 resolution and 65K color display. It uses 8-bit line parallel port communication, and the driver IC is ILI9486. The module includes an LCD display, 5V~3.3V level to circuit, touch screen control circuit, can be directly plugged into the Arduino mega2560 development board, and also supports SD card expansion.

Product Features

- 3.8-inch color screen, support 16BIT RGB 65K color display, display rich colors
- 320x480 HD resolution for clear display
- 8-bit parallel bus transmission for 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
- Military-grade process standards, long-term stable work
- Provide underlying driver technical support

Product Parameters

Name	Description
Display Color	RGB 65K color
SKU	MAR3803
Screen Size	3.8(inch)
Type	TFT
Driver IC	ILI9486
Resolution	480*320 (Pixel)

Luminance	310Cd/m²
Module Interface	8Bit parallel interface
Active Area	52.55x79.04(mm)
Module PCB Size	61.54x105.69 (mm)
Back Light	6 chip HighLight white LEDs
Operating Temperature	-20℃~60℃
Storage Temperature	-30℃~70℃
Operating Voltage	3.3V / 5V
Power Consumption	0.62w
Product Weight	64(g)

Interface Description

Module Pin silkscreen picture

Number	Module Pin	Pin Description
1	**5V**	Power pin
2	**DB0/NC**	Data bus low 8-bit pin
3	**DB1/NC**	

4	DB2/NC	
5	DB3/NC	
6	DB4/NC	
7	DB5/NC	
8	DB6/NC	
9	DB7/NC	
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 four parts: an LCD display control circuit, a level shift circuit, an SD card control circuit, and a touch screen control 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.

working principle

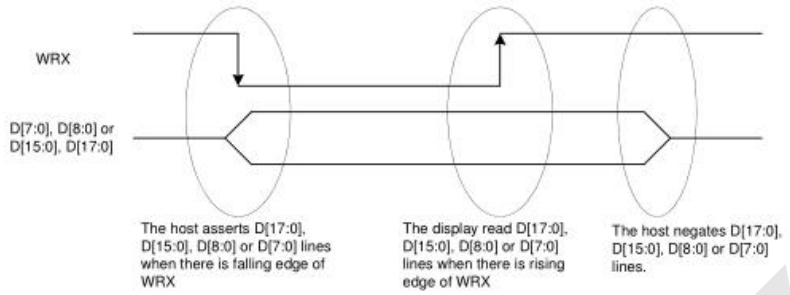
1. Introduction to ITI9486 Controller

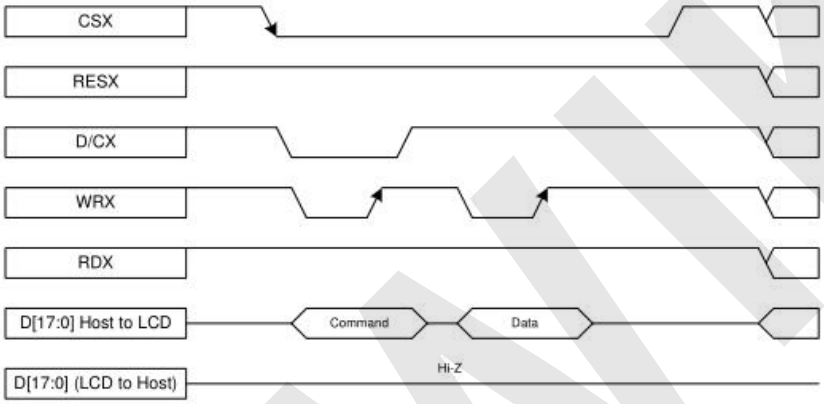
The ITI9486 controller supports a maximum resolution of 320*480 and has a 345600-byte GRAM. It also supports 8-bit, 9-bit, 16-bit, and 18-bit parallel port data buses. It also supports 3-wire and 4-wire SPI serial ports. Since the supported resolution is relatively large and the amount of data transmitted is large, the parallel port transmission is adopted, and the transmission speed is fast. ITI9486 also supports 65K, 262K 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 ITI9486 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 ITI9486 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:





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.

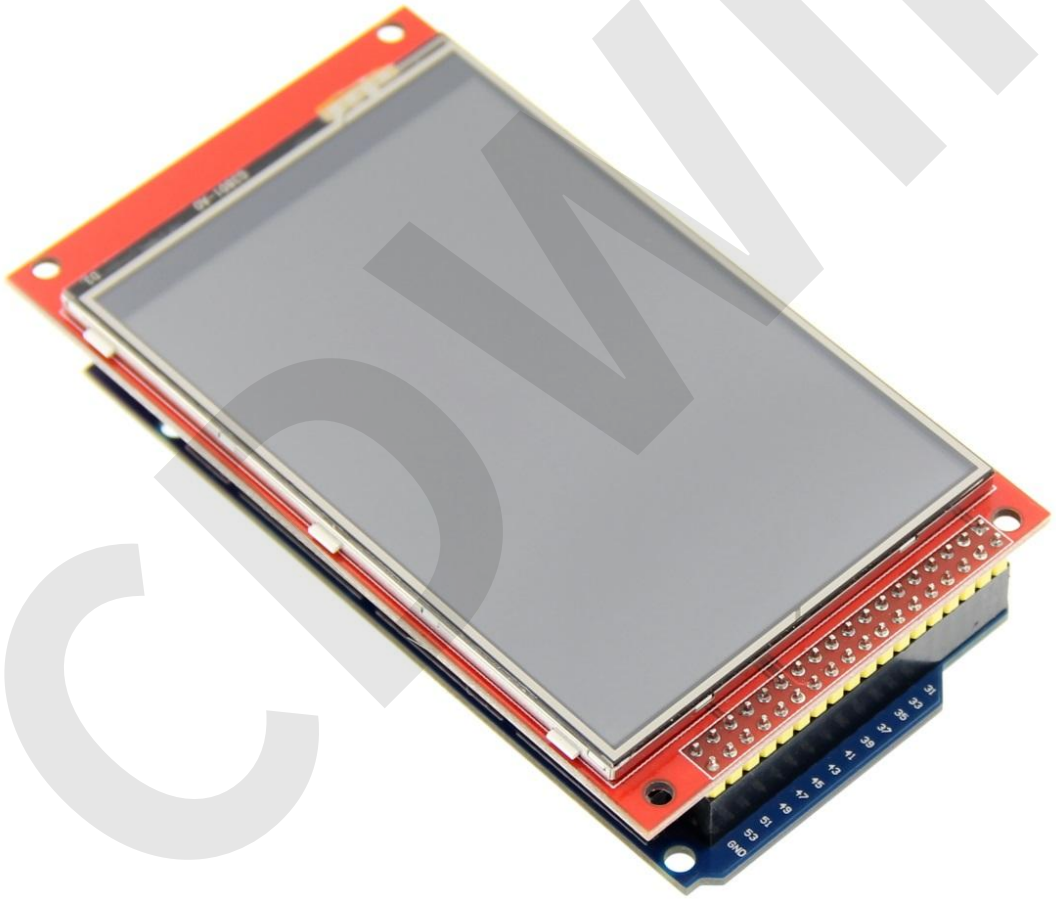
Instructions for use

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**
1	**5V**	5V
2	**DB0/NC**	no need to connect
3	**DB1/NC**	
4	**DB2/NC**	
5	**DB3/NC**	
6	**DB4/NC**	
7	**DB5/NC**	
8	**DB6/NC**	
9	**DB7/NC**	
10	**DB8**	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**	no need to connect
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

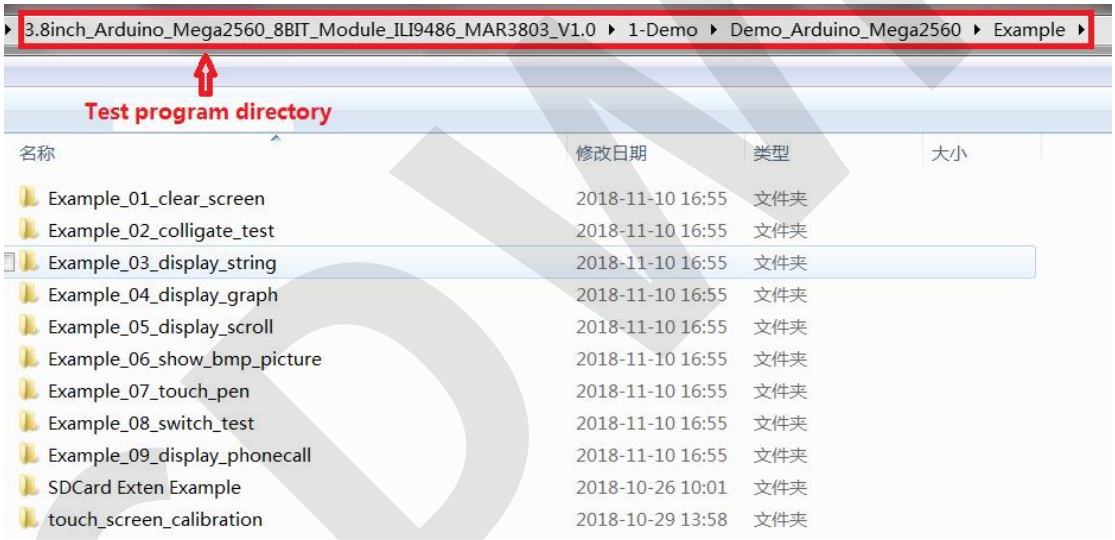
Operating Steps：

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)



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**
1	**5V**	5V
2	**DB0/NC**	no need to connect
3	**DB1/NC**	
4	**DB2/NC**	
5	**DB3/NC**	
6	**DB4/NC**	
7	**DB5/NC**	
8	**DB6/NC**	
9	**DB7/NC**	
10	**DB8**	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**
1	**5V**	5V
2	**DB0/NC**	no need to connect
3	**DB1/NC**	
4	**DB2/NC**	
5	**DB3/NC**	
6	**DB4/NC**	
7	**DB5/NC**	
8	**DB6/NC**	
9	**DB7/NC**	
10	**DB8**	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

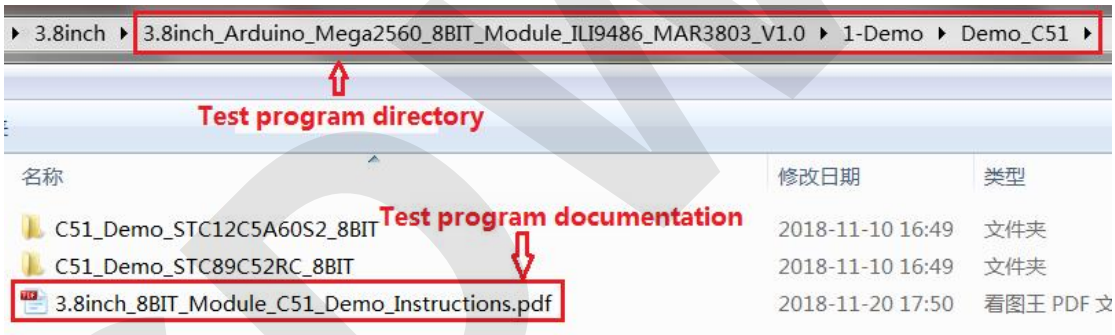
26	**MISO**	P35
27	**MOSI**	P34
28	**TP_CS**	P37
29	**EX_CLK**	P36
30	**GND**	GND

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)



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**
1	**5V**	5V
2	**DB0/NC**	no need to connect
3	**DB1/NC**	
4	**DB2/NC**	
5	**DB3/NC**	
6	**DB4/NC**	
7	**DB5/NC**	
8	**DB6/NC**	
9	**DB7/NC**	
10	**DB8**	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**
1	**5V**	5V
2	**DB0/NC**	no need to connect
3	**DB1/NC**	
4	**DB2/NC**	
5	**DB3/NC**	
6	**DB4/NC**	
7	**DB5/NC**	
8	**DB6/NC**	
9	**DB7/NC**	
10	**DB8**	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**
1	**5V**	5V
2	**DB0/NC**	no need to connect
3	**DB1/NC**	
4	**DB2/NC**	
5	**DB3/NC**	
6	**DB4/NC**	
7	**DB5/NC**	
8	**DB6/NC**	
9	**DB7/NC**	
10	**DB8**	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 microcontroller test program wiring instructions		
Number	**Module Pin**	**Corresponding to Apollo STM32F4/F7 development board wiring pin**
1	**5V**	5V
2	**DB0/NC**	no need to connect
3	**DB1/NC**	
4	**DB2/NC**	
5	**DB3/NC**	
6	**DB4/NC**	
7	**DB5/NC**	
8	**DB6/NC**	
9	**DB7/NC**	
10	**DB8**	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

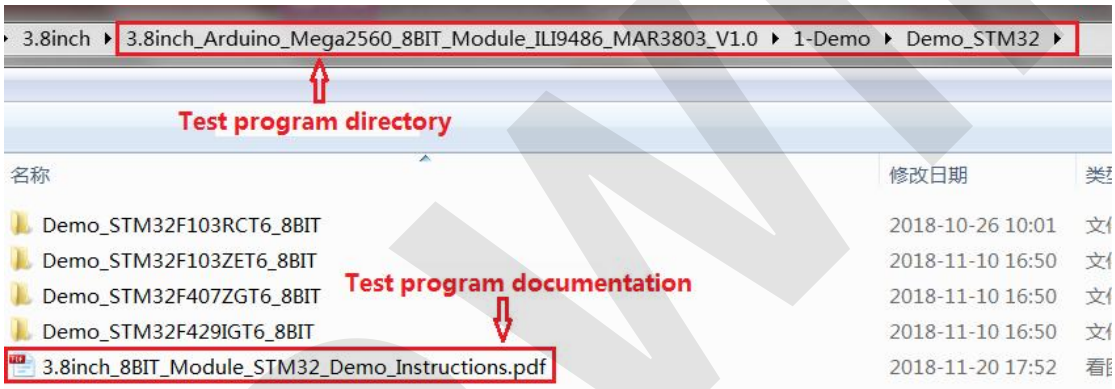
28	**TP_CS**	PH13
29	**EX_CLK**	PH9
30	**GND**	GND

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)



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

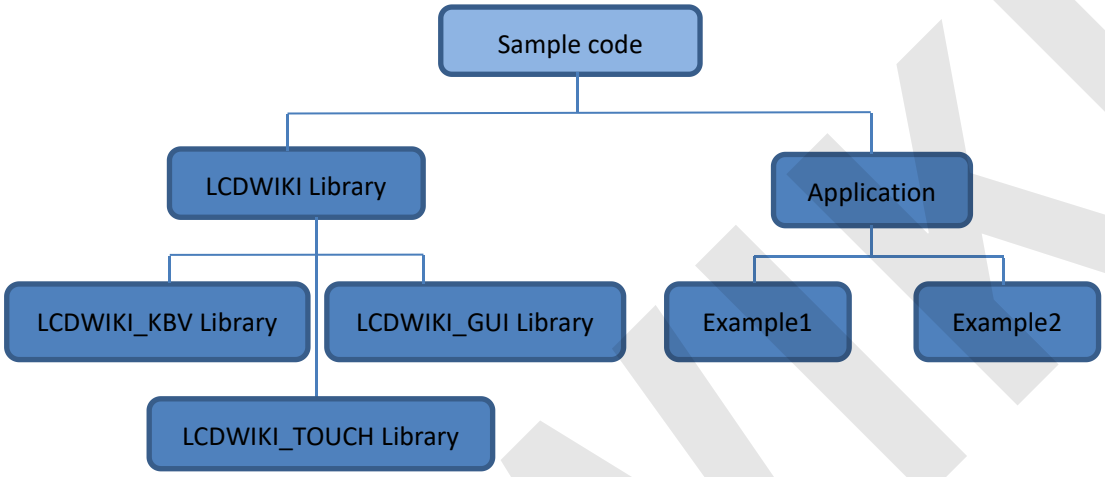
D. If the LCD module displays characters and graphics normally, the program runs successfully；

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;

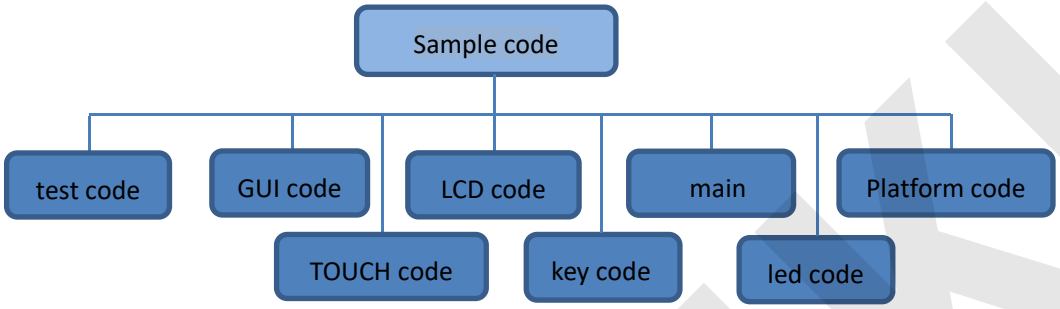
LCDWIKI_GUI is the middle layer library, which is responsible for drawing graphics and displaying characters using the API provided by the underlying library;

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:

```
//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 pins can be defined as any free GPIO.

C. STM32 test program GPIO definition description

The STM32 test program GPIO definition is placed in the lcd.h file as shown below:

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

The pins can be defined as any free GPIO.

3. Parallel port communication code implementation

A. Arduino test program parallel port communication code implementation

The relevant code is implemented in the mcu_16bit_magic.h file of the

LCDWIKI_KBV library, as shown in the figure 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;}
```

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

B. C51 and STM32 test program parallel port communication code implementation

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

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

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

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
}
```

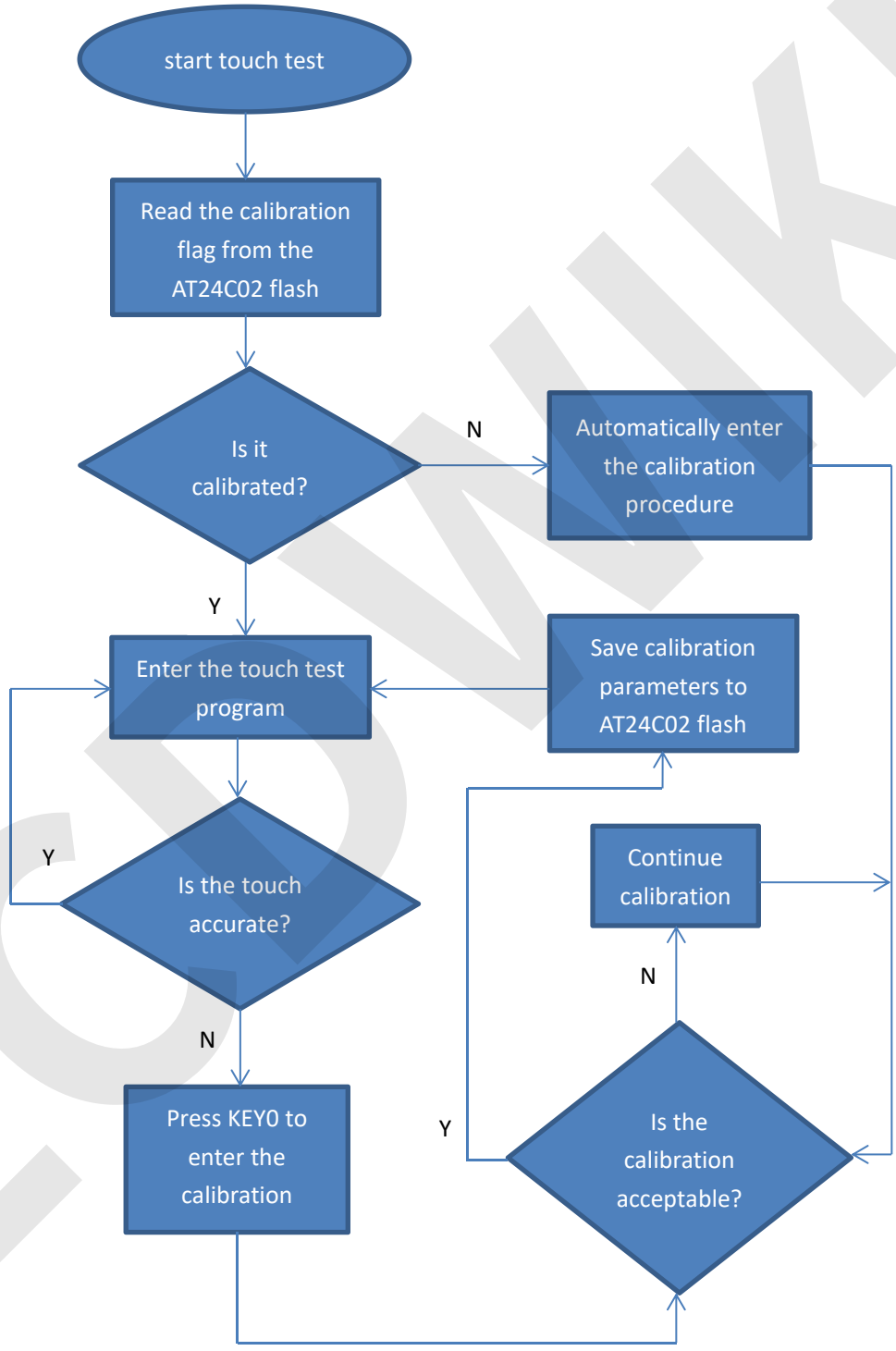
After the touch calibration is passed, you need to save the calibration parameters displayed on the screen in the touch.c file, as shown below:

```
//***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:



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.