5.0inch MCU Display Module User Manual

Product Description

The product is a 5.0-inch TFT LCD display module, which consists of two parts: 5.0-inch display and ssd1963 adapter board. The display module uses 800x480 resolution, supports rgb565k color display, and uses ssd1963 driver IC. The display module also supports the switching function of capacitive touch screen and resistive touch screen.

Product Features

- 5.0-inch color screen, support RGB565 65K color display, display rich colors
- Support 800x480 resolution, the display effect is very clear
- Support 8 bit or 16 bit parallel bus transmission
- Compatible with MCU interface connection of atomic development board
- It supports the switching between capacitive touch screen and resistance touch screen, and the capacitive touch screen can support up to 5 touch points
- Directly plug in esp32 development board
- Compatible with STM32 development board flexible cable and straight connection of needle
- Support PWM backlight brightness adjustment
- Provides a rich sample program for STM32 and C51 platforms
- Military-grade process standards, long-term stable work
- Provide underlying driver technical support

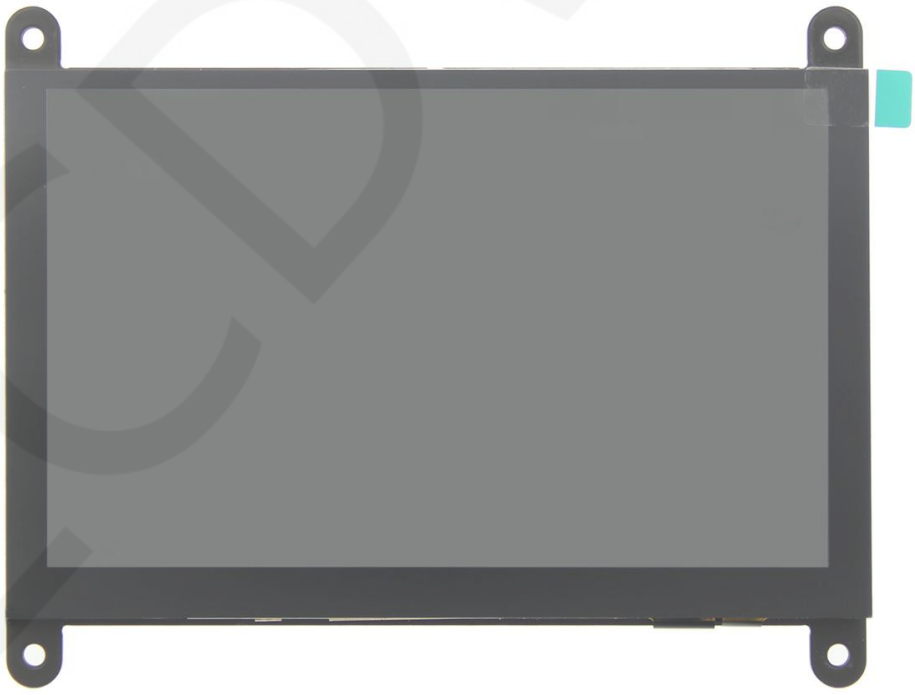
Product Parameters

Name	Description
Display Color	RGB565 65k color
SKU	MRG5101(no touch), MRG5111(have touch)
Screen Size	5.0(inch)
Type	TFT

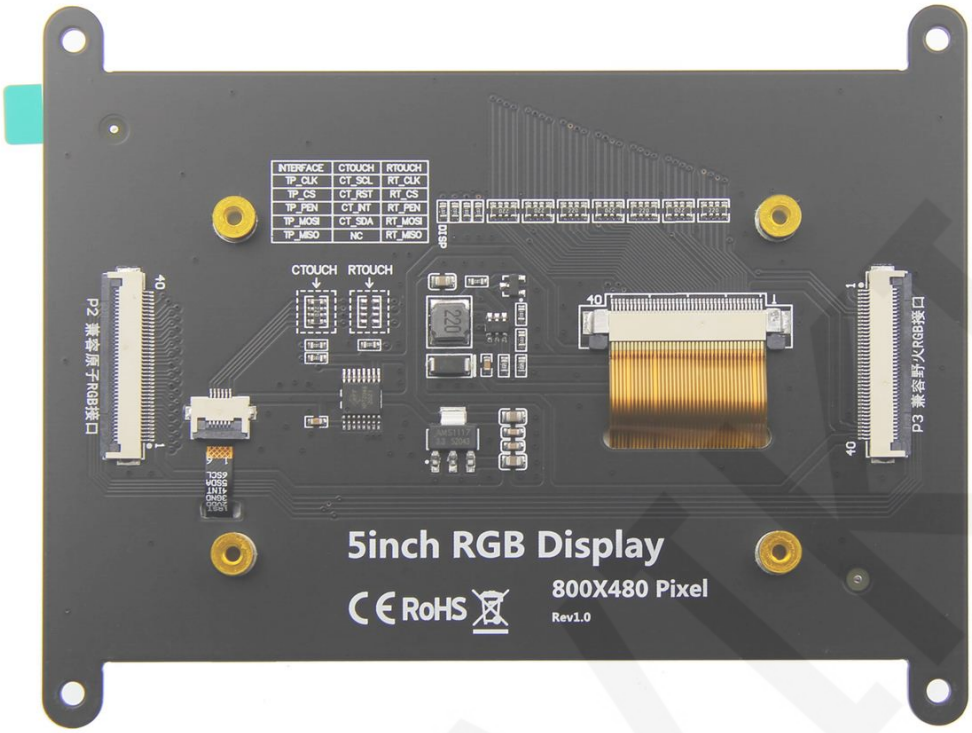
Driver IC	SSD1963
Resolution	800*480 (Pixel)
Module Interface	8Bit or 16Bit parallel interface
Touch Screen Type	Capacitive or Resistive touch screen
Touch IC	FT5426(Capacitive touch), XPT2046(Resistive touch)
Active Area	108.00x64.80(mm)
Module PCB Size	121.11x95.24(mm)
Operating Temperature	-10℃~60℃
Storage Temperature	-20℃~70℃
Input Voltage	5V
IO Voltage	3.3V
Power Consumption	98mA(The backlight is off), 161mA(The backlight is the brightest)
Product Weight(Net weight)	133g

Interface Description

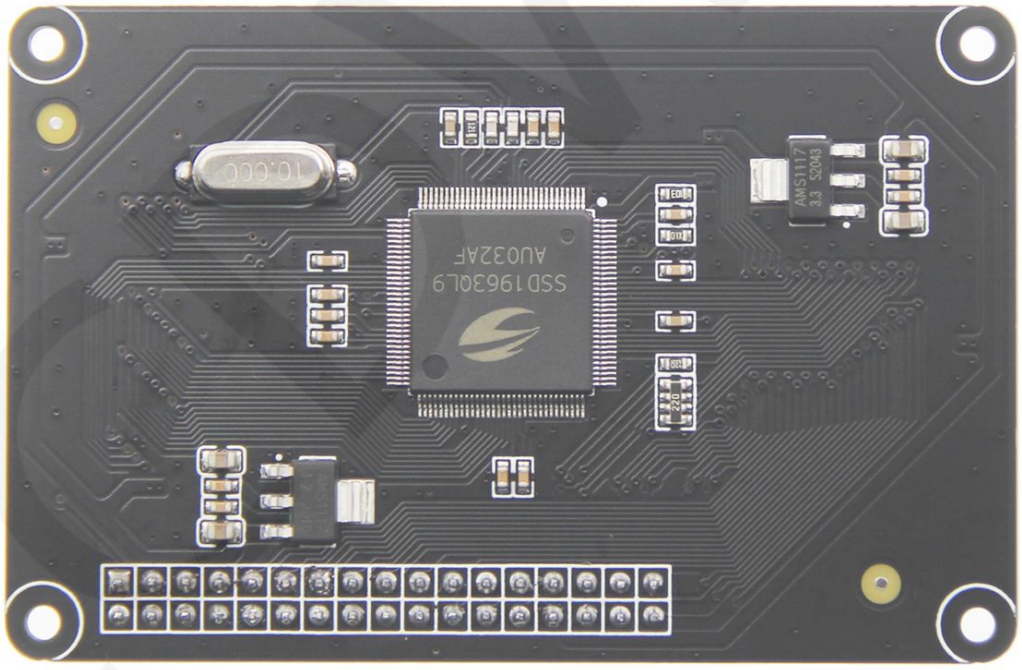
The appearance of the product is shown in Picture 1 ~ Picture 5.



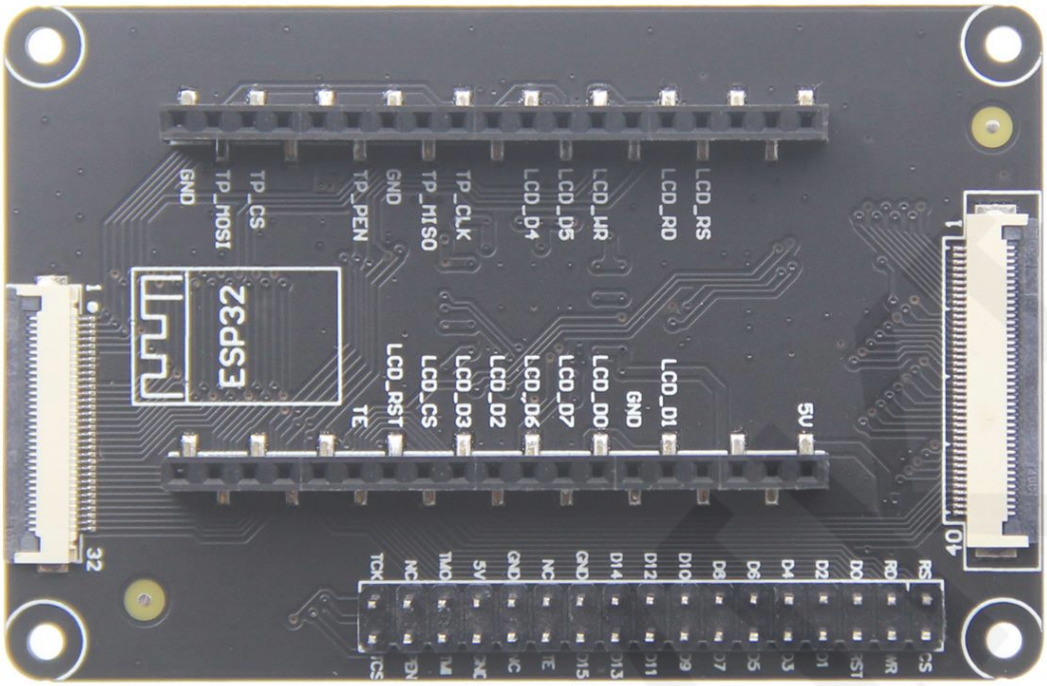
Picture1. Front view of module



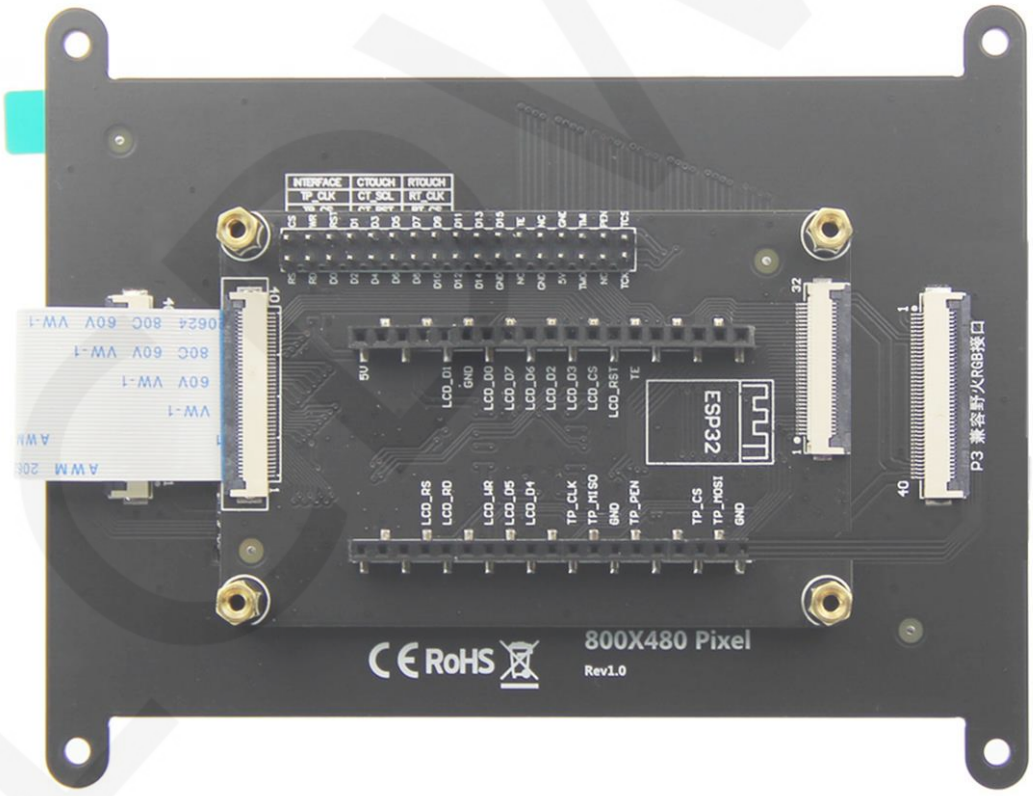
Picture2. Back view of module



Picture3. Front view of adapter board

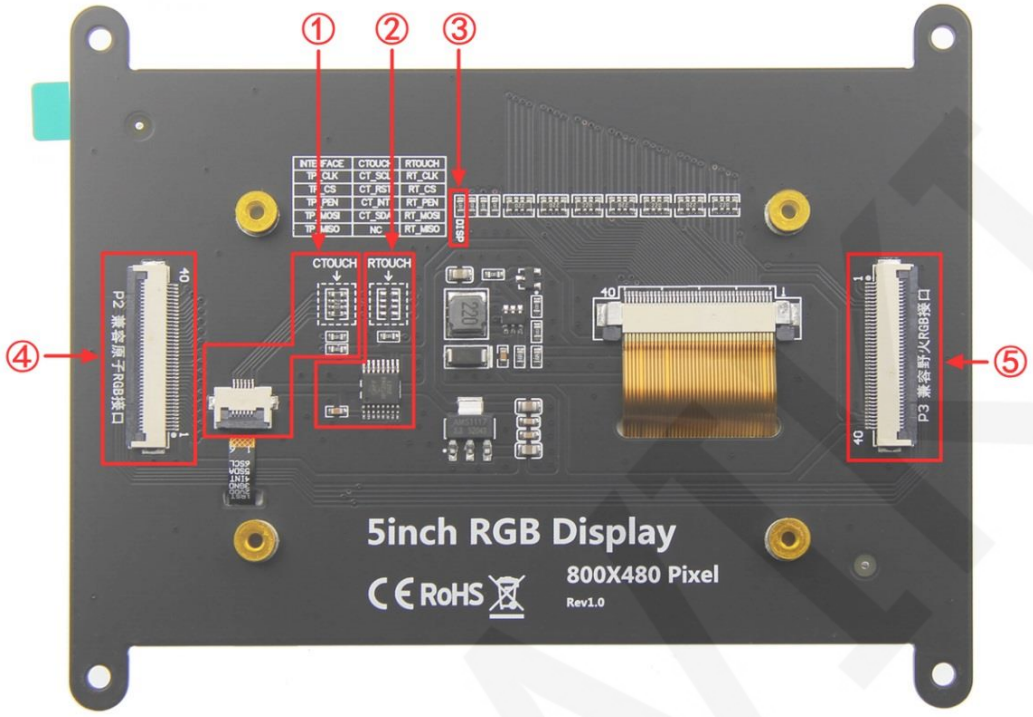


Picture 4. Back view of adapter board



Picture 5. Combination picture

The module interface and selection circuit are shown in Picture 6:

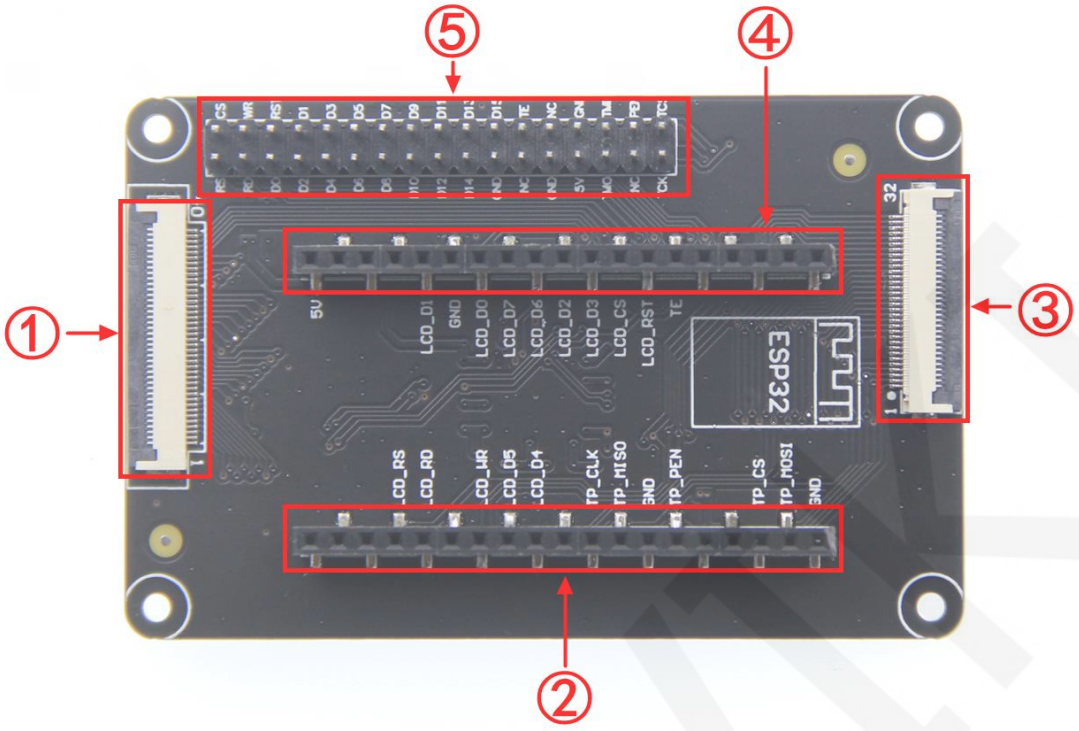


Picture 6. The module interface and selection circuit

Each identification circuit in Picture 6 is described as follows:

①--Capacitive touch screen circuit
②--Resistance touch screen circuit
③--Disp resistance
④--P2 interface (Used to connect the adapter board)
⑤--P3 interface (Not used)

The adapter board interface is shown in Picture 7:



The descriptions of the marks in Picture 7 are as follows:

①--40pin flexible cable output interface

②④--In line interface of esp32 development board

③--STM32 development board 32pin flexible cable input interface

⑤--STM32 development board 34pin pin input interface

1. The module supports switching between capacitive touch screen and resistive touch screen. When using capacitive touch screen, please weld Capacitive touch screen circuit; when using resistance touch screen, please weld Resistance touch screen circuit. If you often need to switch the touch screen, the simplest way is to solder other circuits and switch only the drain in the dotted line box.
2. **When using the product, the DISP resistor must be welded, otherwise the screen will not display after the program runs.**
3. The interface pins of the adapter board are described as follows:

STM32 development board row pin input interface pin description		
Number	**Pin name**	**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)
6~21	D0~D15	LCD 16 bit data bus pin (use d0 ~ D7 in 8-bit mode)
22	GND	Module power ground pin
23	TE	LCD tearing effect signal pin (read only)
24	NC	No definition, reserved
25	NC	No definition, reserved
26	GND	Module power ground pin
27	GND	Module power ground pin
28	5V	Module power supply positive pin (connected to 5V)
29	TMI	Resistance touch screen SPI bus read data pin (capacitor touch screen not used)
30	TMO	IIC bus data pin of capacitive touch screen (SPI bus write data pin of resistance touch screen)
31	PEN	Touch screen interrupt detection pin(Low level when a touch occurs)
32	NC	No definition, reserved
33	TCS	Capacitor touch screen reset pin (resistance touch screen chip selection pin)
34	TCK	IIC bus clock pin of capacitive touch screen (SPI bus clock pin of resistance touch screen)

STM32 development board flexible cable input interface pin description

Number	Pin name	Pin description
1	TCK	IIC bus clock pin of capacitive touch screen (SPI bus clock pin of resistance touch screen)
2	TCS	Capacitor touch screen reset pin (resistance touch screen chip selection pin)

3	PEN	Touch screen interrupt detection pin(Low level when a touch occurs)
4	TMO	IIC bus data pin of capacitive touch screen (SPI bus write data pin of resistance touch screen)
5	TMI	Resistance touch screen SPI bus read data pin (capacitor touch screen not used)
6	5V	Module power supply positive pin (connected to 5V)
7	GND	Module power ground pin
8	GND	Module power ground pin
9	NC	No definition, reserved
10	NC	No definition, reserved
11	TE	LCD tearing effect signal pin (read only)
12~27	D15~D0	LCD 16 bit data bus pin (use d0 ~ D7 in 8-bit mode)
28	RST	LCD reset control pin(low level reset)
29	RD	LCD read control pin
30	WR	LCD write control pin
31	RS	LCD register / data selection control pin(high level: register, low level: data)
32	CS	LCD reset control pin(low level enable)

ESP32 test program directly insert instructions

Number	Module Pin	Remarks
1	5V	Power pin
2	LCD_RS	LCD register / data selection pin High level:data; Low level:register
3	LCD_RD	LCD read control pin
4	LCD_D1	Pin 2 of 8-bit parallel data bus
5	GND	Power ground pin
6	LCD_WR	LCD write control pin
7	LCD_D0	Pin 1 of 8-bit parallel data bus
8	LCD_D5	Pin 6 of 8-bit parallel data bus

9	LCD_D7	Pin 8 of 8-bit parallel data bus
10	LCD_D4	Pin 5 of 8-bit parallel data bus
11	LCD_D6	Pin 7 of 8-bit parallel data bus
12	LCD_D2	Pin 3 of 8-bit parallel data bus
13	TP_CLK	IIC bus clock control pin of capacitive touch screen(SPI bus clock control pin of resistance touch screen)
14	LCD_D3	Pin 4 of 8-bit parallel data bus
15	TP_MISO	SPI bus read data pin of resistance touch screen(capacitive touch screen not used)
16	LCD_CS	LCD chip select control pin
17	GND	Power ground pin
18	LCD_RST	LCD reset control pin
19	TP_PEN	Touch screen interrupt control pin
20	TE	Tearing Effect Signal pin(read-only)
21	TP_CS	Capacitive touch screen reset pin (resistance touch screen chip selection pin)
22	TP_MOSI	IIC bus data pin of capacitive touch screen (resistance touch screen SPI bus write data pin)
23	GND	Power ground pin

Pin description of flexible cable output interface		
Number	**Pin name**	**Pin description**
1	VCC5	Power input pin（connect to 5V）
2	VCC5	Power input pin（connect to 5V）
3~10	R0 ~ R7	8-bit RED data pin
11	GND	power ground pin
12~19	G0 ~ G7	8-bit GREEN data pin
20	GND	power ground pin
21~28	B0 ~ B7	8-bit BLUE data pin
29	GND	power ground pin
30	PCLK	Pixel clock control pin

31	HSYNC	Horizontal synchronous signal control pin
32	VSYNC	Vertical synchronous signal control pin
33	DE	Data enable signal control pin
34	BL	LCD backlight control pin
35	TP_CS	Capacitor touch screen reset pin (resistance touch screen chip selection pin)
36	TP_MOSI	Data pin of IIC bus of capacitance touch screen (write data pin of SPI bus of resistance touch screen)
37	TP_MISO	Resistance touch screen SPI bus read data pin (capacitance touch screen not used)
38	TP_CLK	IIC bus clock control pin of capacitive touch screen (SPI bus clock control pin of resistance touch screen)
39	TP_PEN	Touch screen interrupt control pin
40	RST	LCD reset control pin (effective at low level)

Hardware Configuration

The hardware circuit of the product consists of two parts: display screen and adapter board.

The hardware circuit of the display screen consists of ten parts: backlight control circuit, screen resolution selection circuit, 40pin display interface, drain circuit, P2 user interface, P3 user interface, capacitive touch screen interface circuit, resistance touch screen control circuit,Touch screen selection circuit and power supply circuit. The details are as follows:

1. The backlight control circuit is used to provide backlight voltage to display screen and adjust backlight brightness.
2. The screen resolution selection circuit is used to select the display type (distinguished according to the resolution). Its principle is to connect pull-up or pull-down resistors on R7, G7 and B7 data lines respectively, and then determine the resolution of the display screen used by reading the status of the three data lines (equivalent to reading the display screen ID), so as to select different configurations. In this way, a test

example can be compatible with multiple displays in software. Of course, the module only supports one resolution, so the resistance of R7, G7 and B7 data lines is fixed.

3. The 40pin display interface is used to access and control the display screen.
4. The drain circuit is used to balance the data line impedance between the display and the user interface.
5. P2, P3 user interface is used for external development board.
6. Capacitive touch screen interface circuit is used to intervene capacitive touch screen and control IIC pin pull-up.
7. The resistance touch screen control circuit is used to detect the touch signal and collect the coordinate data of the touch screen, and then carry out ADC conversion.
8. The touch screen selection circuit is used to select the connected touch screen and switch through welding resistance.
9. The power circuit is used to convert the input 5V power supply to 3.3V.

The hardware circuit of adapter board consists of eight parts: 32pin LCD FPC interface, touch balance impedance circuit, ssd1963 reset circuit, power circuit, ssd1963 control circuit, 40pin display interface, 34pin LCD pin interface and esp32 in-line interface. The details are as follows:

1. 32pin LCD FPC interface is used to connect STM32 development board through flexible cable.
2. The touch balance impedance circuit is used to balance the touch screen data line impedance.
3. Ssd1963 reset circuit is used to reset ssd1963.
4. The power supply circuit is used for level conversion (5V to 3.3V and 5V to 1.2V) and voltage stabilization.
5. Ssd1963 control circuit is used to control ssd1963 IC.
6. 40pin display interface is used to connect display module through flexible cable.
7. 34pin LCD pin interface is used to connect STM32 development board through hard wired.
8. Esp32 plug in interface is used to plug in esp32 development board.

working principle

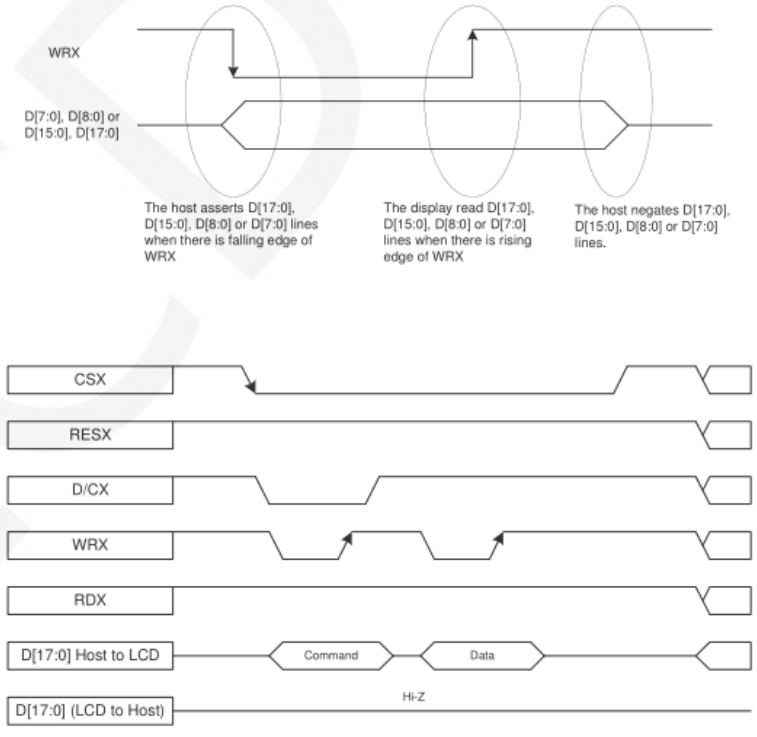
1. Introduction to SSD1963 Controller

The SSD1963 is a single-chip controller for 16.7M color TFT-LCDs. It supports a maximum resolution of 864*480 and has a GRAM of 1215K bytes. It also supports 8-bit, 9-bit, 16-bit, 18-bit and 24bit parallel port data buses. 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. SSD1963 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.

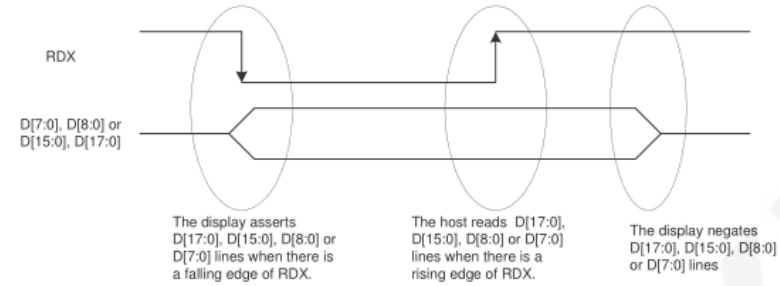
The SSD1963 controller uses 16bit (RGB565) to control a pixel display, so it can display up to 65K colors per pixel. The pixel address setting is performed in the order of rows and columns, and the incrementing and decreasing direction is determined by the scanning mode. The SSd1963 display method is performed by setting the address and then setting the color value.

2. Introduction to parallel port communication

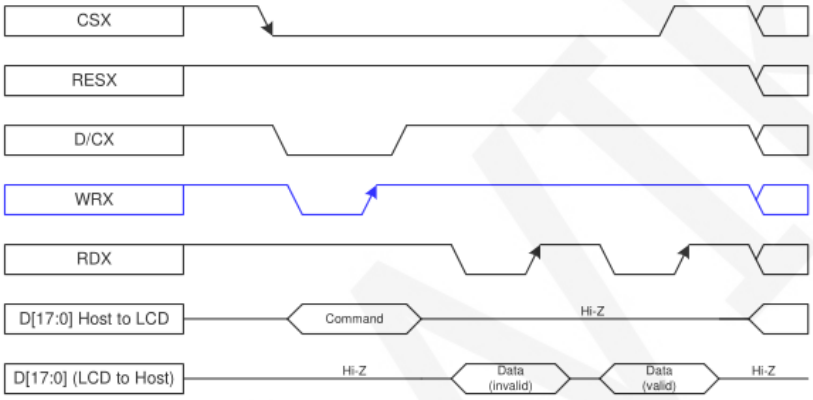
The parallel port communication write mode timing is as shown below:



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. STM32 instructions

Wiring instructions:

See the interface description for pin assignments.

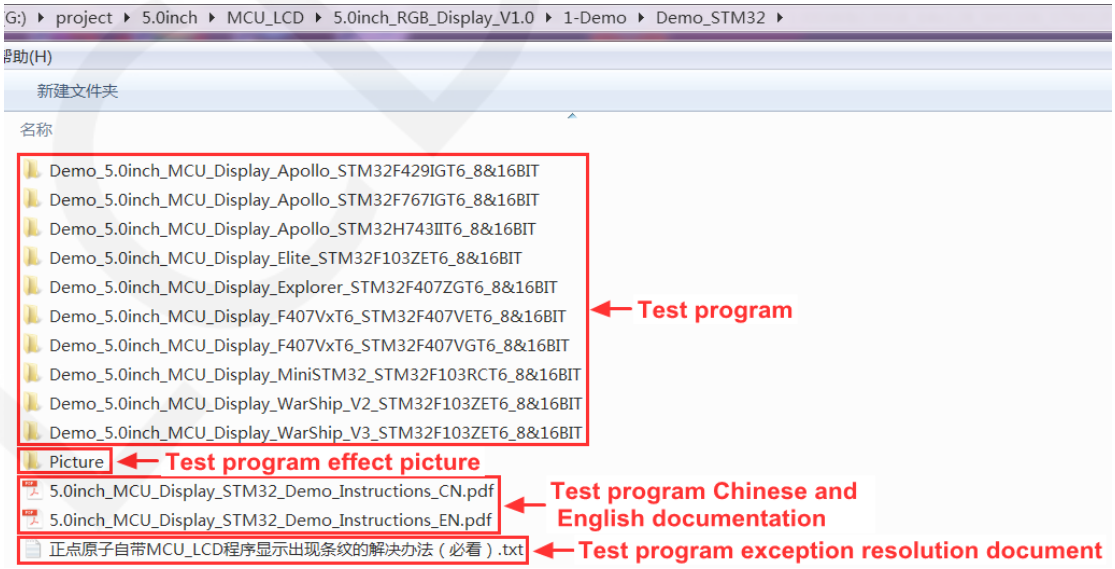
For specific wiring instructions, please refer to the STM32 test procedure description document.

Operating Steps：

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

B. Select the STM32 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;

2. ESP32 instructions

Wiring instructions:

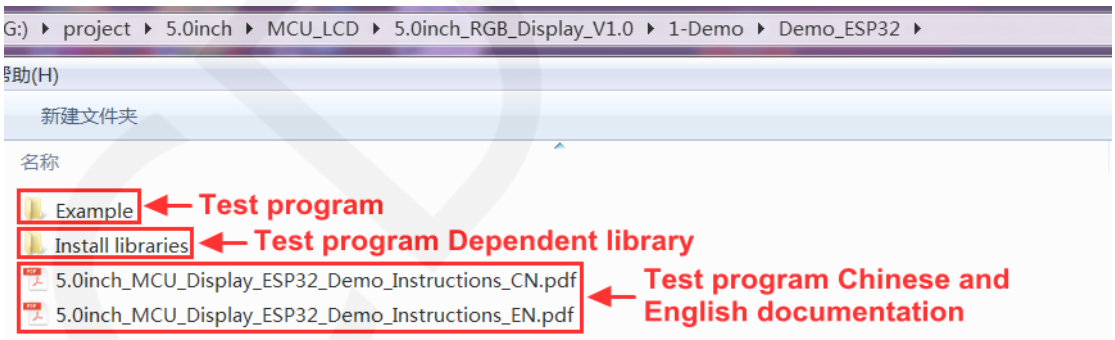
See the interface description for pin assignments.

For specific wiring instructions, please refer to the ESP32 test procedure description document.

Operating Steps:

A. Connect the LCD module and the ESP32 development board according to the above wiring instructions, and power on;

B. Select the ESP32 development board 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;

See the following document for detailed instructions on the construction of esp32 development environment and test program compilation and download:

http://www.lcdwiki.com/res/PublicFile/Arduino_For_ESP32_Installation_Instructions_EN.pdf

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

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

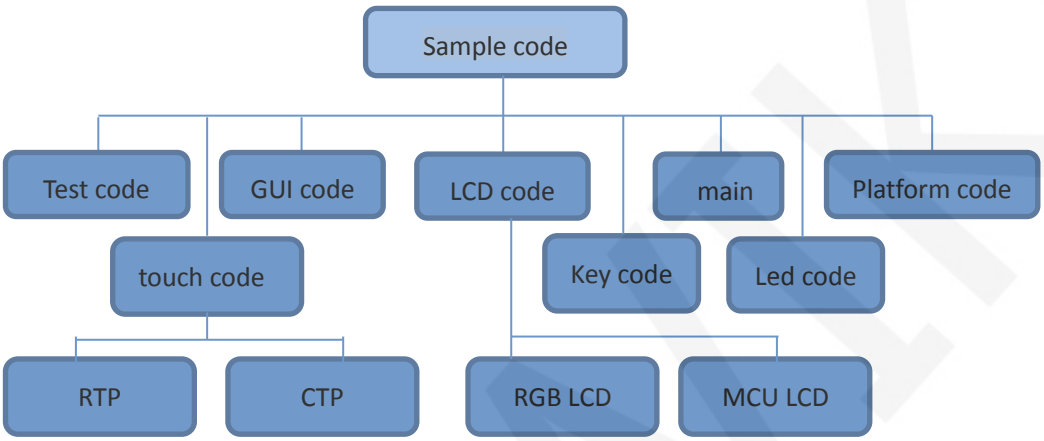
successfully;

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, Including MCD LCD and RGB LCD;

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;

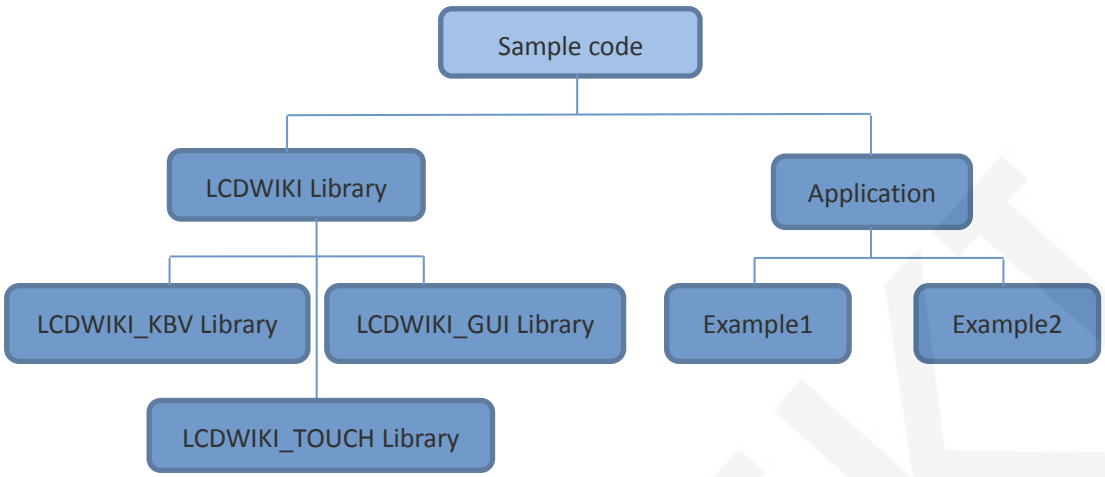
Touch screen related operations are included in the touch code, Including resistance touch and capacitance touch;

The key processing related code is included in the key code;

The code related to the led configuration operation is included in the led code;

B. 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;

2. GPIO definition description

A. STM32 MCU LCD test program GPIO definition description

Because the module is directly connected with flexible cable, it is not necessary to modify the GPIO definition in principle

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 GPIO definition related to STM32 touch screen is placed in the files ft5426. h and ctpiic. H, as shown below (take the STM32F103RCT6 microcontroller IO simulation test program as an example):

```
//与电容触摸屏连接的芯片引脚(未包含IIC引脚)
//IO操作函数
#define FT_RST              PCout(13) //FT5206复位引脚
#define FT_INT              PCin(1)   //FT5206中断引脚
```

```
//IIC IO方向设置
#define CTP_SDA_IN()  {GPIOC->CRL&=0XFFFF0FFF;GPIOC->CRL|=8<<4*3;}
#define CTP_SDA_OUT() {GPIOC->CRL&=0XFFFF0FFF;GPIOC->CRL|=3<<4*3;}

//IO操作函数
#define CTP_IIC_SCL    PCout(0)        //SCL
#define CTP_IIC_SDA    PCout(3)        //SDA
#define CTP_READ_SDA   PCin(3)         //输入SDA
```

B. ESP32 test program GPIO definition description

Since the module is directly inserted into the adapter board, it is not recommended to modify the GPIO port definition in principle .

The GPIO definition of LCD is placed in mcu_ 8bit_ magic. h, as shown in the figure below:

```
#define CS_PIN   33  // Chip select control pin (library pulls permanently low
#define CD_PIN   15  // Data Command control pin - must use a pin in the range
0-31
//#define RST_PIN  32  // Reset pin, toggles on startup
#define WR_PIN    4  // Write strobe control pin - must use a pin in the range
0-31
#define RD_PIN    2  // Read strobe control pin
extern uint32_t set_mask_buffer[256];

#define TFTLCD_D0   12  // Must use pins in the range 0-31 for the data bus
#define TFTLCD_D1   13  // so a single register write sets/clears all bits.
#define TFTLCD_D2   26  // Pins can be randomly assigned, this does not affect
#define TFTLCD_D3   25  // TFT screen update performance.
#define TFTLCD_D4   17
#define TFTLCD_D5   16
#define TFTLCD_D6   27
#define TFTLCD_D7   14
```

The GPIO definition of the touch screen is placed at the beginning of each test example, as shown in the following figure::

```
#define INT 21
#define CRST 22
#define SCL 18
#define SDA 23
```

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 or mcu_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. ESP32 test program parallel port communication code implementation

The relevant code is implemented in the mcu_8bit_magic.h file as shown below:

```
// Mask for the 8 data bits to set pin directions
 #define dir_mask ((1 << TFTLCD_D0) | (1 << TFTLCD_D1) | (1 << TFTLCD_D2) | (1
<< TFTLCD_D3) | (1 << TFTLCD_D4) | (1 << TFTLCD_D5) | (1 << TFTLCD_D6) | (1 <<
TFTLCD_D7))

 // Data bits and the write line are cleared to 0 in one step
 #define clr_mask (dir_mask | (1 << WR_PIN))

 // A lookup table is used to set the different bit patterns, this uses 1kByte
of RAM
 #define set_mask(C) set_mask_buffer[C] // 63fps Sprite rendering test 33%
faster, graphicstest only 1.8% faster than shifting in real time

 // Write 8 bits to TFT
#define write8(C)    {GPIO.out_w1tc = clr_mask; GPIO.out_w1ts = set_mask((uint8
_t)C); WR_STROBE;}
#define read8(dst)
```

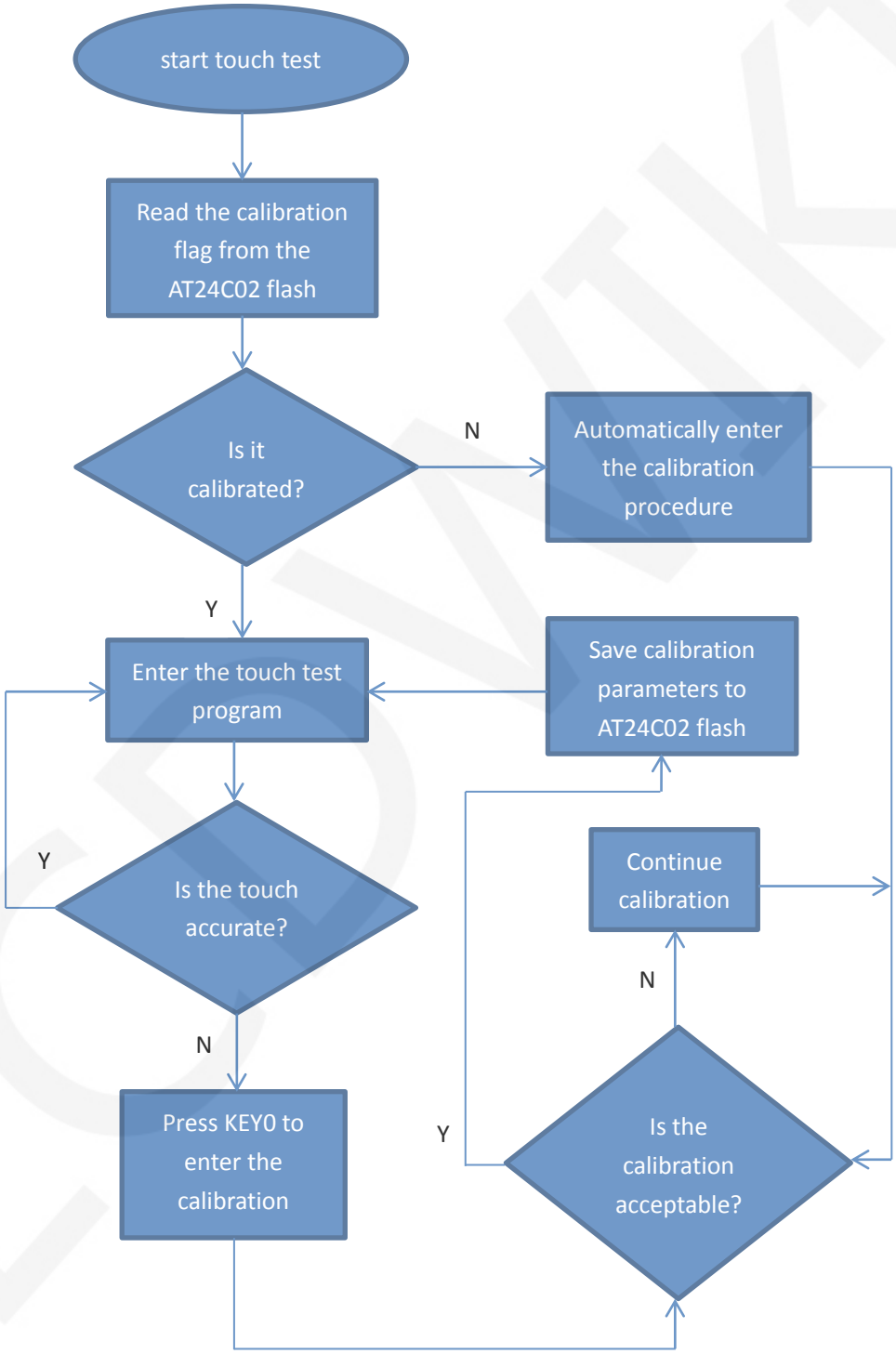
4. touch screen calibration instructions

A. 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:



B. 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:

```
4: #define XFAC       852
5: #define XOFFSET    (-14)
6: #define YFAC       1284
7: #define YOFFSET    (-30)
```

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)

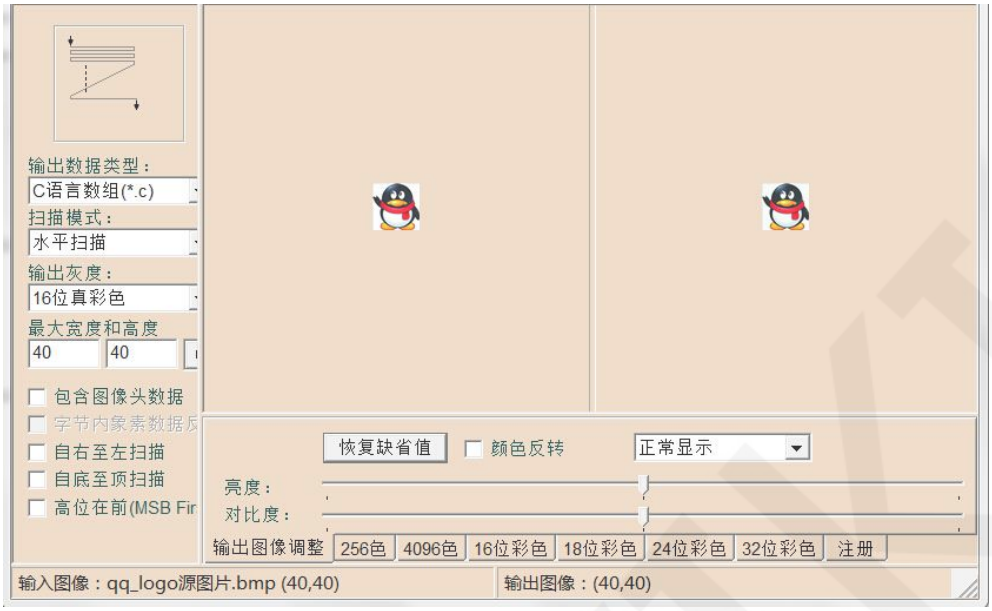
Output number system selects hexadecimal number

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.