

4.3inch MCU Display Module 用户手册

产品概述

该产品为一款 4.3 寸 TFT LCD 显示模块，其由两部分组成：4.3 寸显示屏和 SSD1963 转接板。该显示模块使用 800x480 分辨率，支持 RGB565 65K 色显示，其内部使用 SSD1963 驱动 IC。该显示模块还支持电容触摸屏和电阻触摸屏切换功能。

产品特点

- 4.3 寸高清彩屏，支持 RGB565 65K 色显示，显示色彩非常丰富
- 支持 800x480 分辨率，显示效果非常清晰
- 支持 8 位或 16 位并行总线传输
- 兼容原子开发板 MCU 屏接口连接
- 支持电容触摸屏和电阻触摸屏切换，电容触摸屏最大支持 5 点触摸
- 可直插 ESP32 开发板
- 兼容 STM32 开发板软排线和排针直连
- 支持 PWM 背光亮度调节
- 提供丰富的 STM32 和 ESP32 平台示例程序
- 军工级工艺标准,长期稳定工作
- 提供底层驱动技术支持

产品参数

名称	描述
显示颜色	RGB565 65K 彩色
SKU	MRG4301（无触摸）、MRG4311（带触摸）
尺寸	4.3(inch)
类型	TFT
驱动芯片	SS1963
分辨率	800x480(Pixel)
模块接口	8 位或 16 位并口

触摸屏类型	电容触摸屏或电阻触摸屏
触摸屏 IC	FT5426（电容触摸）、XPT2046（电阻触摸）
有效显示区域	95.04x53.86(mm)
模块尺寸	105.41x86.41(mm)
工作温度	-10~60℃
存储温度	-20~70℃
输入电压	5V
IO 电平	3.3V
功耗	84mA（背光灭）、198mA（背光最亮）
产品重量（净重）	117g

接口说明

产品的外观如图 1~图 5 所示：

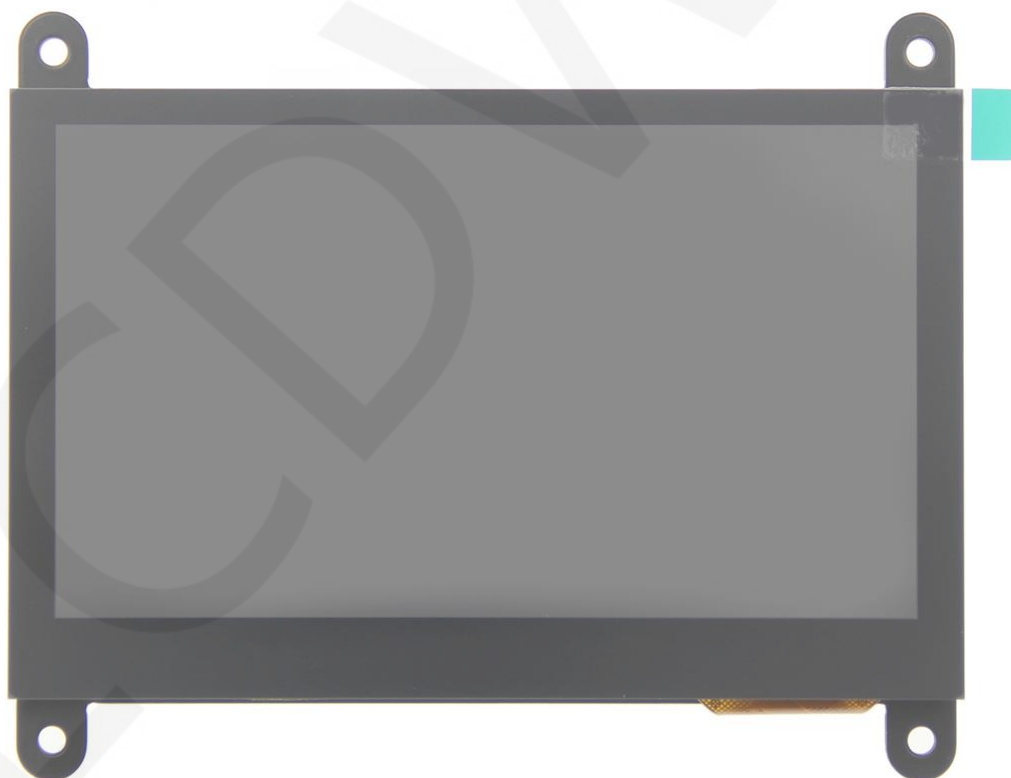


图 1. 显示模块正面图

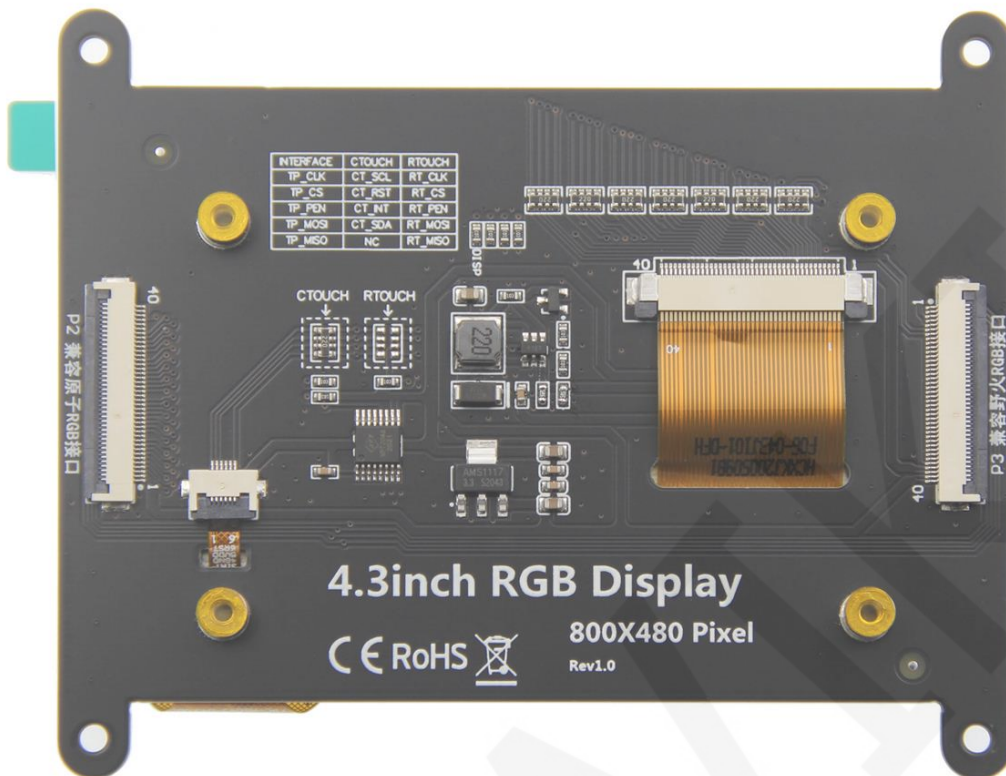


图 2. 显示模块背面图

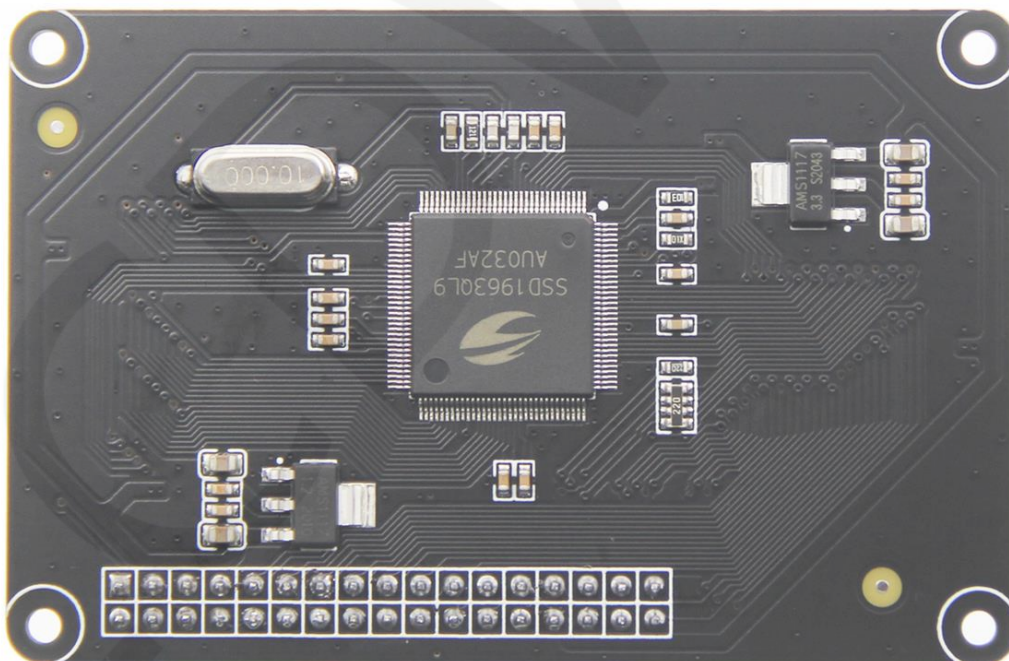


图 3. 转接板正面图

显示模块接口及选择电路如图 6 所示：

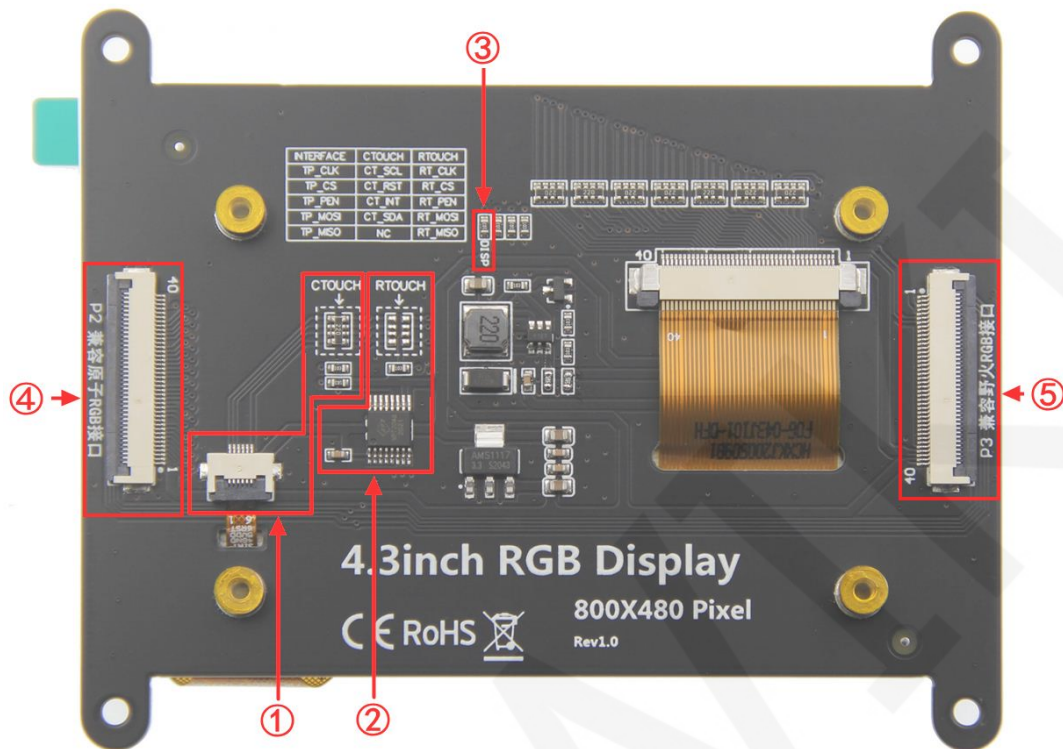


图 6. 显示模块接口及选择电路图

图 6 中各个标识说明如下：

- ①--电容触摸屏电路
- ②--电阻触摸屏电路
- ③--DISP 电阻
- ④--P2 接口（用来连接转接板）
- ⑤--P3 接口（没有使用）

转接板接口如图 7 所示：

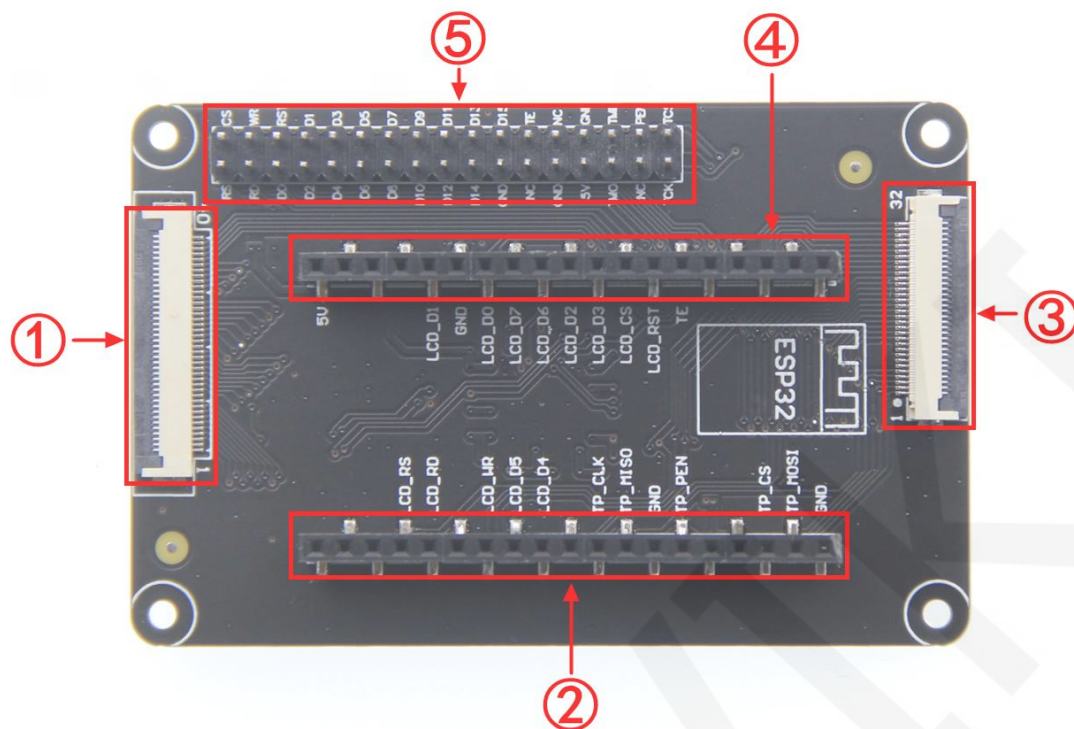


图 7. 转接板接口图

图 7 中各个标识说明如下：

- ①--40PIN 软排线输出接口
- ②④--ESP32 开发板直插接口
- ③--STM32 开发板 32PIN 软排线输入接口
- ⑤--STM32 开发板 34PIN 排针输入接口

1、该模块支持电容触摸屏和电阻触摸屏切换。使用电容触摸屏时，请将电容触摸屏电路焊接好；使用电阻触摸屏时，请将电阻触摸屏电路焊接好。如果经常要切换触摸屏，最简单的做法就是，将其他电路焊接好，只切换虚线框里的排阻。

2、使用该产品时，必须焊接 DISP 电阻，否则程序运行后，屏无显示。

3、转接板各接口引脚说明如下：

STM32开发板排针输入接口引脚说明		
序号	引脚名称	引脚说明
1	CS	液晶屏片选控制引脚（低电平使能）
2	RS	液晶屏寄存器/数据选择控制引脚（低电平：寄存器，高电平：数据）
3	WR	液晶屏写控制引脚

4	RD	液晶屏读控制引脚
5	RST	液晶屏复位控制引脚（低电平复位）
6~21	D0~D15	液晶屏16位数据总线引脚（8位模式下使用D0~D7）
22	GND	电源地
23	TE	液晶屏撕裂效应信号引脚（只读）
24	NC	无定义，保留
25	NC	无定义，保留
26	GND	电源地
27	GND	电源地
28	5V	模块电源正极引脚（接5V）
29	TMI	电阻触摸屏SPI总线读数据引脚（电容触摸屏没使用）
30	TMO	电容触摸屏IIC总线数据引脚（电阻触摸屏SPI总线写数据引脚）
31	PEN	触摸屏中断检测引脚（发生触摸时为低电平）
32	NC	无定义，保留
33	TCS	电容触摸屏复位引脚（电阻触摸屏片选引脚）
34	TCK	电容触摸屏IIC总线时钟引脚（电阻触摸屏SPI总线时钟引脚）

STM32开发板软排线输入接口引脚说明

序号	引脚名称	引脚说明
1	TCK	电容触摸屏IIC总线时钟引脚（电阻触摸屏SPI总线时钟引脚）
2	TCS	电容触摸屏复位引脚（电阻触摸屏片选引脚）
3	PEN	触摸屏中断检测引脚（发生触摸时为低电平）
4	TMO	电容触摸屏IIC总线数据引脚（电阻触摸屏SPI总线写数据引脚）
5	TMI	电阻触摸屏SPI总线读数据引脚（电容触摸屏没使用）
6	5V	模块电源正极引脚（接5V）
7	GND	电源地

8	GND	电源地
9	NC	无定义，保留
10	NC	无定义，保留
11	TE	液晶屏撕裂效应信号引脚（只读）
12~27	D15~D0	液晶屏16位数据总线引脚（8位模式下使用D0~D7）
28	RST	液晶屏复位控制引脚（低电平复位）
29	RD	液晶屏读控制引脚
30	WR	液晶屏写控制引脚
31	RS	液晶屏寄存器/数据选择控制引脚（低电平：寄存器，高电平：数据）
32	CS	液晶屏片选控制引脚（低电平使能）

ESP32开发板直插接口引脚说明

序号	模块引脚	引脚说明
1	5V	电源引脚
2	LCD_RS	液晶屏寄存器/数据选择引脚 （高电平：数据；低电平：寄存器）
3	LCD_RD	液晶屏读控制引脚
4	LCD_D1	8位并口数据总线第2引脚
5	GND	电源地
6	LCD_WR	液晶屏写控制引脚
7	LCD_D0	8位并口数据总线第1引脚
8	LCD_D5	8位并口数据总线第6引脚
9	LCD_D7	8位并口数据总线第8引脚
10	LCD_D4	8位并口数据总线第5引脚
11	LCD_D6	8位并口数据总线第7引脚
12	LCD_D2	8位并口数据总线第3引脚
13	TP_CLK	电容触摸屏IIC总线时钟控制引脚 （电阻触摸屏SPI总线时钟控制引脚）
14	LCD_D3	8位并口数据总线第4引脚
15	TP_MISO	电阻触摸屏SPI总线读数据引脚（电容触摸屏没使用）

16	LCD_CS	液晶屏片选控制引脚（低电平使能）
17	GND	电源地
18	LCD_RST	液晶屏复位控制引脚（低电平有效）
19	TP_PEN	触摸屏中断检测引脚（发生触摸时为低电平）
20	TE	撕裂效应信号引脚（只读）
21	TP_CS	电容触摸屏复位引脚（电阻触摸屏片选引脚）
22	TP_MOSI	电容触摸屏IIC总线数据引脚（电阻触摸屏SPI总线写数据引脚）
23	GND	电源地

软排线输出接口引脚说明

序号	引脚名称	引脚说明
1	VCC5	电源输入引脚（接5V）
2	VCC5	电源输入引脚（接5V）
3~10	R0 ~ R7	8位RED数据引脚
11	GND	电源地
12~19	G0 ~ G7	8位GREEN数据引脚
20	GND	电源地
21~28	B0 ~ B7	8位BLUE数据引脚
29	GND	电源地
30	PCLK	像素时钟控制引脚
31	HSYNC	水平同步信号控制引脚
32	VSNC	垂直同步信号控制引脚
33	DE	数据使能信号控制引脚
34	BL	液晶屏背光控制引脚
35	TP_CS	电容触摸屏复位引脚（电阻触摸屏片选引脚）
36	TP_MOSI	电容触摸屏IIC总线数据引脚（电阻触摸屏SPI总线写数据引脚）
37	TP_MISO	电阻触摸屏SPI总线读数据引脚（电容触摸屏没使用）
38	TP_CLK	电容触摸屏IIC总线时钟控制引脚（电阻触摸屏SPI总线时钟控制引脚）

39	TP_PEN	触摸屏中断控制引脚
40	RST	液晶屏复位控制引脚（低电平有效）

硬件配置

该产品的硬件电路包含两部分：显示屏和转接板。

显示屏硬件电路包含十大部分：背光控制电路、屏分辨率选择电路、40PIN 显示屏接口、排阻电路、P2 用户接口、P3 用户接口、电容触摸屏接口电路、电阻触摸屏控制电路、触摸屏选择电路以及电源电路。具体介绍如下：

- 1、背光控制电路用于给显示屏提供背光电压以及调节背光亮度。
- 2、屏分辨率选择电路用于选择显示屏类型（根据分辨率区分）。其原理就是在分别在 R7、G7、B7 数据线上接上上拉或者下拉电阻，然后通过读取这三根数据线的状态（相当于读取显示屏 ID）来确定所使用显示屏的分辨率，从而选择不同的配置。这样在软件上可以达到一个测试示例兼容多款显示屏的目的。当然该模块只支持一种分辨率，所以 R7、G7、B7 数据线所接电阻都是固定的。
- 3、40PIN 显示屏接口用来接入并控制显示屏。
- 4、排阻电路用来平衡显示屏和用户接口之间的数据线路阻抗。
- 5、P2、P3 用户接口用来外接开发板。
- 6、电容触摸屏接口电路用来介入电容触摸屏并控制 IIC 引脚上拉。
- 7、电阻触摸屏控制电路用来检测触摸信号并采集触摸屏坐标数据，然后进行 ADC 转换。
- 8、触摸屏选择电路用来选择接入的触摸屏，通过焊接排阻来切换。
- 9、电源电路用于将输入的 5V 电源转换为 3.3V。

转接板硬件电路包含八大部分：32Pin LCD FPC 接口、触摸平衡阻抗电路、SSD1963 复位电路、电源电路、SSD1963 控制电路、40Pin 显示屏接口、34Pin LCD 排针接口、ESP32 直插接口。具体介绍如下：

- 1、32Pin LCD FPC 接口用来通过软排线连接 STM32 开发板。
- 2、触摸平衡阻抗电路用来平衡触摸屏数据线路阻抗。
- 3、SSD1963 复位电路用来对 SSD1963 进行复位控制。
- 4、电源电路用来进行电平转换（5V 转 3.3V 和 5V 转 1.2V）和稳压。

- 5、SSD1963 控制电路用来控制 SSD1963 IC。
- 6、40Pin 显示屏接口用来通过软排线连接显示模块。
- 7、34Pin LCD 排针接口用来通过硬排线连接 STM32 开发板。
- 8、ESP32 直插接口用来直插 ESP32 开发板。

工作原理

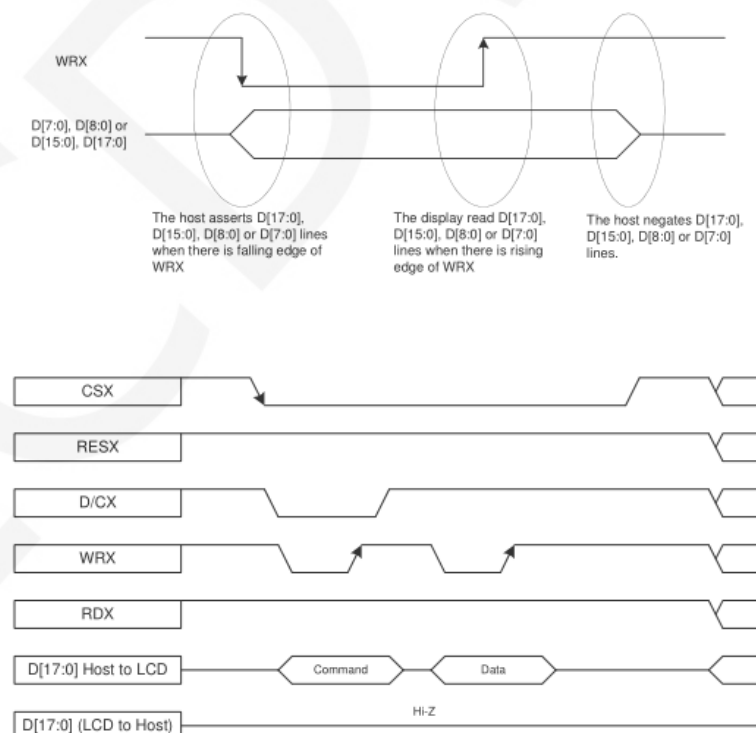
1、SSD1963 控制器简介

SSD1963 是一种用于 16.7M 彩色 TFT-LCD 的单片控制器，支持的最大分辨率为 864*480，拥有一个 1215K 字节大小的 GRAM。同时支持 8 位、9 位、16 位、18 位、24 位并口数据总线。由于支持的分辨率比较大，传输的数据量大，所以采用并口传输，传输速度快。SSD1963 还支持 65K、262K、16M RGB 颜色显示，显示色彩很丰富，同时支持旋转显示和滚动显示以及视频播放，显示方式多样。

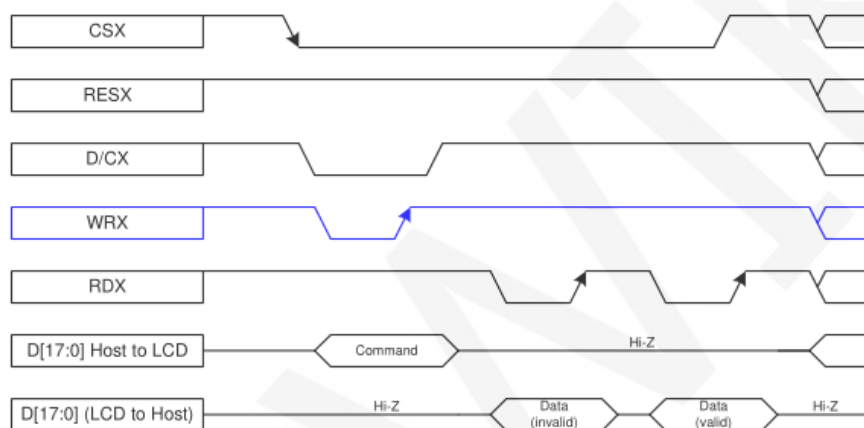
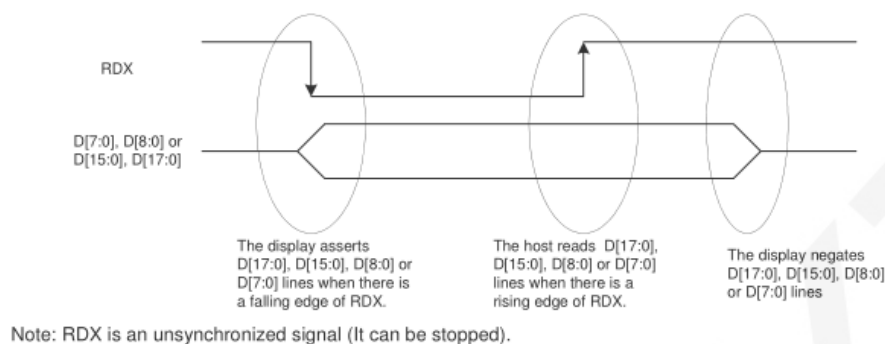
SSD1963 控制器使用 16bit (RGB565) 来控制一个像素点显示，因此可以每个像素点显示颜色多达 65K 种。像素点地址设置按照行列的顺序进行，递增递减方向由扫描方式决定。SSD1963 显示方法按照先设置地址再设置颜色值进行。

2、并口通信简介

并口通信写模式时序如下图所示：



并口通信读模式时序如下图所示：



CSX 为片选信号，用于开启和禁止并口通信，低电平有效

RESX 为外部复位信号，低电平有效

D/CX 为数据或者命令选择信号，1-写数据或者命令参数，0-写命令

WRX 为写数据控制信号

RDX 为读数据控制信号

D[X:0]为并口数据位，共有 8 位、9 位、16 位、18 位四种类型

当进行写入操作时，在已经复位的基础上，先设置数据或者命令选择信号，然后将片选信号拉低，接下来从主机输入需要写入的内容，然后将写数据控制信号拉低再拉高，数据在写控制信号的上升沿会被写入到液晶屏控制 IC，最后将片选信号拉高，一次数据写入操作完成。

当进入读操作时，在已经复位的基础上，先将片选信号拉低，然后将数据或者命令选择信号拉高，接下来将读数据控制信号拉低，然后从液晶屏控制 IC 读取数据，再将读数据控制信号拉高，数据在读数据控制信号上升沿会被读取出来，最后将片选信号拉

高，一次数据读取操作完成。

使用说明

1、STM32 使用说明

接线说明：

引脚标注见接口说明。

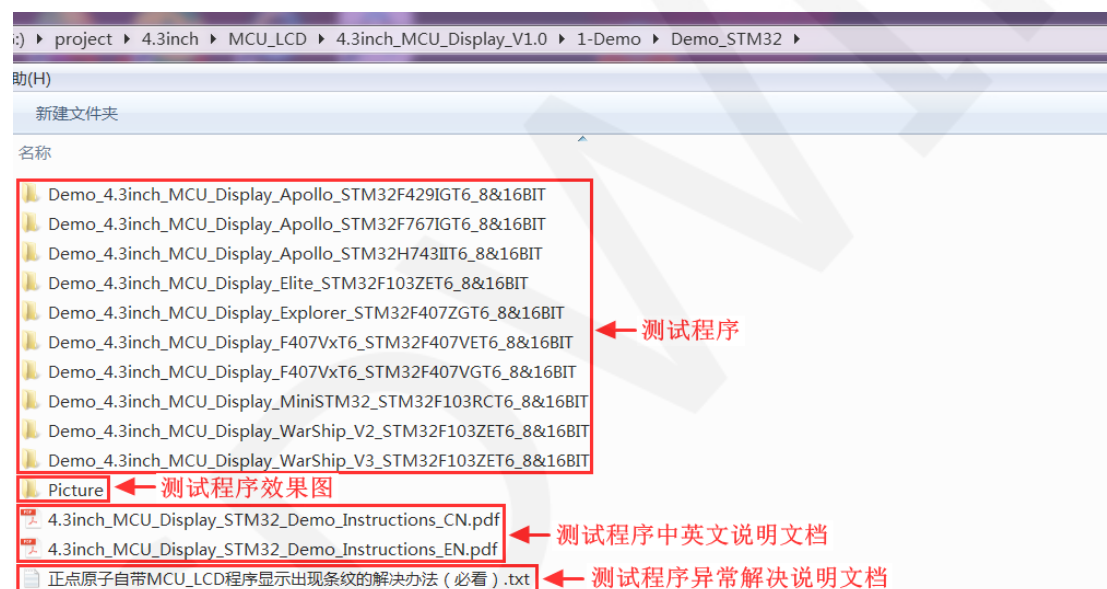
具体的接线说明请参考 STM32 测试程序说明文档。

操作说明：

A、按照接线说明将 LCD 模块和 STM32 单片机连接起来，并上电；

B、选择需要测试的 STM32 测试程序，如下图所示：

（测试程序说明请查阅测试程序说明文档）



C、打开所选的测试程序工程，进行编译和下载；

关于 STM32 测试程序编译和下载的详细说明见如下文档：

http://www.lcdwiki.com/res/PublicFile/STM32_Keil_Use_Illustration_CN.pdf

D、LCD 模块如果正常显示字符和图形，则说明程序运行成功；

2、ESP32 使用说明

接线说明：

引脚标注见接口说明。

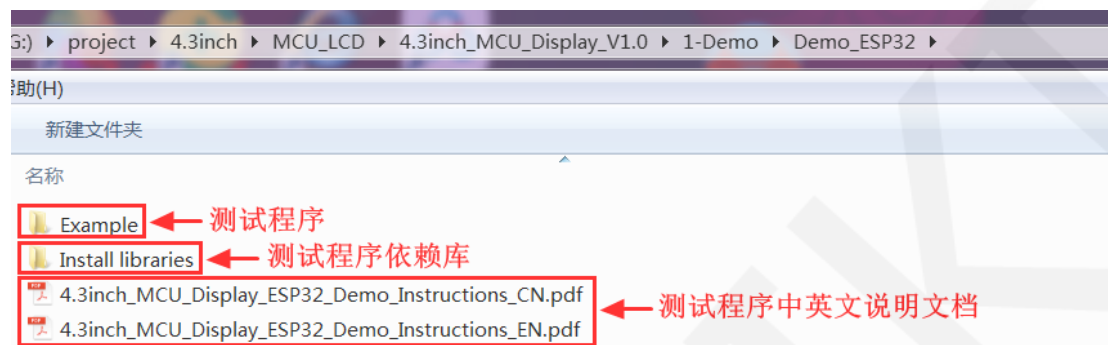
具体的接线说明请参考 ESP32 测试程序说明文档。

操作步骤:

A、按照接线说明将 LCD 模块和 ESP32 开发板连接起来，并上电；

B、选择需要测试的 ESP32 测试程序，如下图所示：

（测试程序说明请查阅测试程序包中测试程序说明文档）



C、打开所选的测试程序工程，进行编译和下载；

关于 ESP32 开发环境搭建以及测试程序编译、下载的详细说明见如下文档：

[http://www.lcdwiki.com/res/PublicFile/Arduino For ESP32 Installation Instructions CN.pdf](http://www.lcdwiki.com/res/PublicFile/Arduino%20For%20ESP32%20Installation%20Instructions%20CN.pdf)

[http://www.lcdwiki.com/res/PublicFile/Arduino IDE Use Illustration CN.pdf](http://www.lcdwiki.com/res/PublicFile/Arduino%20IDE%20Use%20Illustration%20CN.pdf)

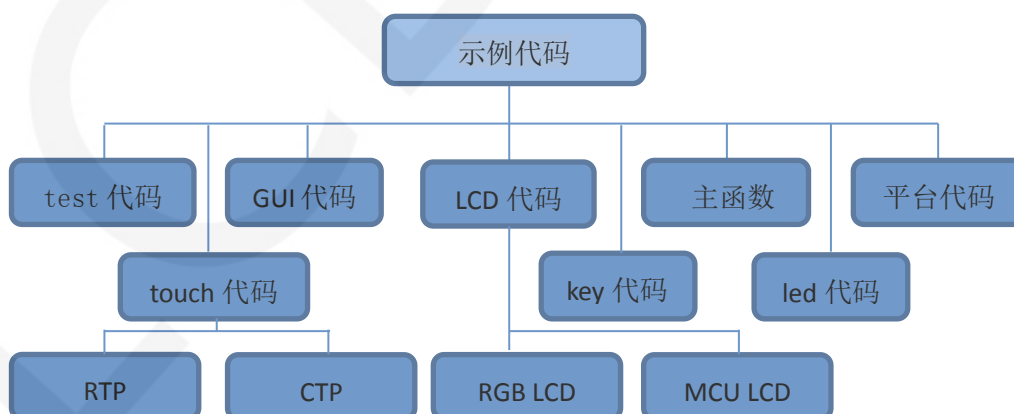
D、LCD 模块如果正常显示字符和图形，则说明程序运行成功；

软件说明

1、代码架构

A、STM32 代码架构说明

代码架构如下图所示：



主程序运行时的 Demo API 代码包含在 test 代码中；

LCD 初始化以及相关的并口读写数据操作都包含在 LCD 代码中，包括 MCU LCD 和 RGB LCD；

画点、线、图形以及中英文字符显示相关的操作都包含在 GUI 代码中；

主函数实现应用程序运行；

平台代码因平台而异；

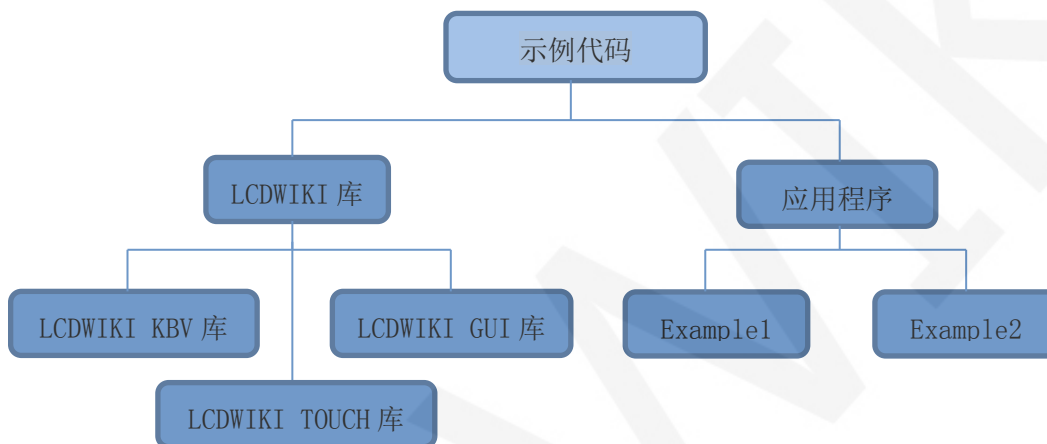
触摸屏相关的操作都包含在 touch 代码中, 包括电阻触摸和电容触摸；

按键处理相关的代码都包含在 key 代码中；

led 配置操作相关的代码都包含在 led 代码中；

B、ESP32 代码架构说明

代码架构如下图所示：



Arduino 的测试程序代码由两部分组成：LCDWIKI 库和应用代码。

LCDWIKI 库包含三部分内容：LCDWIKI_KBV 库、LCDWIKI_GUI 库以及 LCDWIKI_TOUCH 库。

应用程序包含几个测试示例，每个测试示例包含不同的测试内容。

LCDWIKI_KBV 为底层库，和硬件有关联，主要负责操作寄存器，包括硬件模块初始化，数据和命令传输，像素点坐标和颜色设置，显示方式配置等。

LCDWIKI_GUI 为中间层库，负责使用底层库提供的 API 实现图形的绘制，字符显示。

LCDWIKI_TOUCH 为触摸屏底层库，主要负责触摸中断检测，触摸数据采样和 AD 转换以及触摸数据发送。

应用程序是利用 LCDWIKI 库提供的 API，编写一些测试示例，实现某方面的测试功能。

2、GPIO 定义说明

A、STM32 MCU LCD 测试程序 GPIO 定义说明

因为模块是用软排线直连的，原则上不需要修改 GPIO 定义。

LCD 屏的 GPIO 定义放在 lcd.h 或者 mcd_lcd.h 文件里，其定义采用两种方式：

- 1) STM32F103RCT6 单片机测试程序采用 IO 模拟方式（其不支持 FSMC 总线）
- 2) 其他 STM32 单片机测试程序采用 FSMC 总线方式

STM32F103RCT6 单片机 IO 模拟测试程序 LCD 屏的 GPIO 定义如下图所示：

```

////////////////////////////////////
//-----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_1
//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 测试程序 LCD 屏 GPIO 定义如下图所示（以 STM32F103ZET6 单片机 FSMC 测试程序为例）：

```

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

//qdttech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)

```

STM32 触摸屏相关的 GPIO 定义放在 ft5426.h 和 ctpiic.h 文件里面，如下图所示（以 STM32F103RCT6 单片机 IO 模拟测试程序为例）：

```
//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 测试程序 GPIO 定义说明

因为模块是直插到转接板上的，原则上不建议修改 GPIO 口定义。

LCD 的 GPIO 定义放在 mcu_8bit_magic.h 里，如下图所示：

```
#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
```

触摸屏的 GPIO 定义放在每个测试示例开头，如下图所示：

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

3、并口通信代码实现

A、STM32 MCU LCD 测试程序并口通信代码实现

LCD 并口通信代码都放在 LCD.c 或者 mcu_lcd.c 文件里，其实现采用两种方式：

- 1) STM32F103RCT6 单片机测试程序采用 I0 模拟方式（其不支持 FSMC 总线）
- 2) 其他 STM32 单片机测试程序采用 FSMC 总线方式

I0 模拟测试程序实现如下图所示：

```
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;
}
```

FSMC 测试程序实现如下图所示：

```
u16 LCD_read(void)
{
    u16 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; //写入要写的的数据
}
```

都实现了 8、16 位命令写入以及 8、16 位数据写入和读取。

B、ESP32 测试程序并口通信代码实现

LCD 并口通信代码放在 mcu_8bit_magic.h 里，如下图所示：

```
// 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))

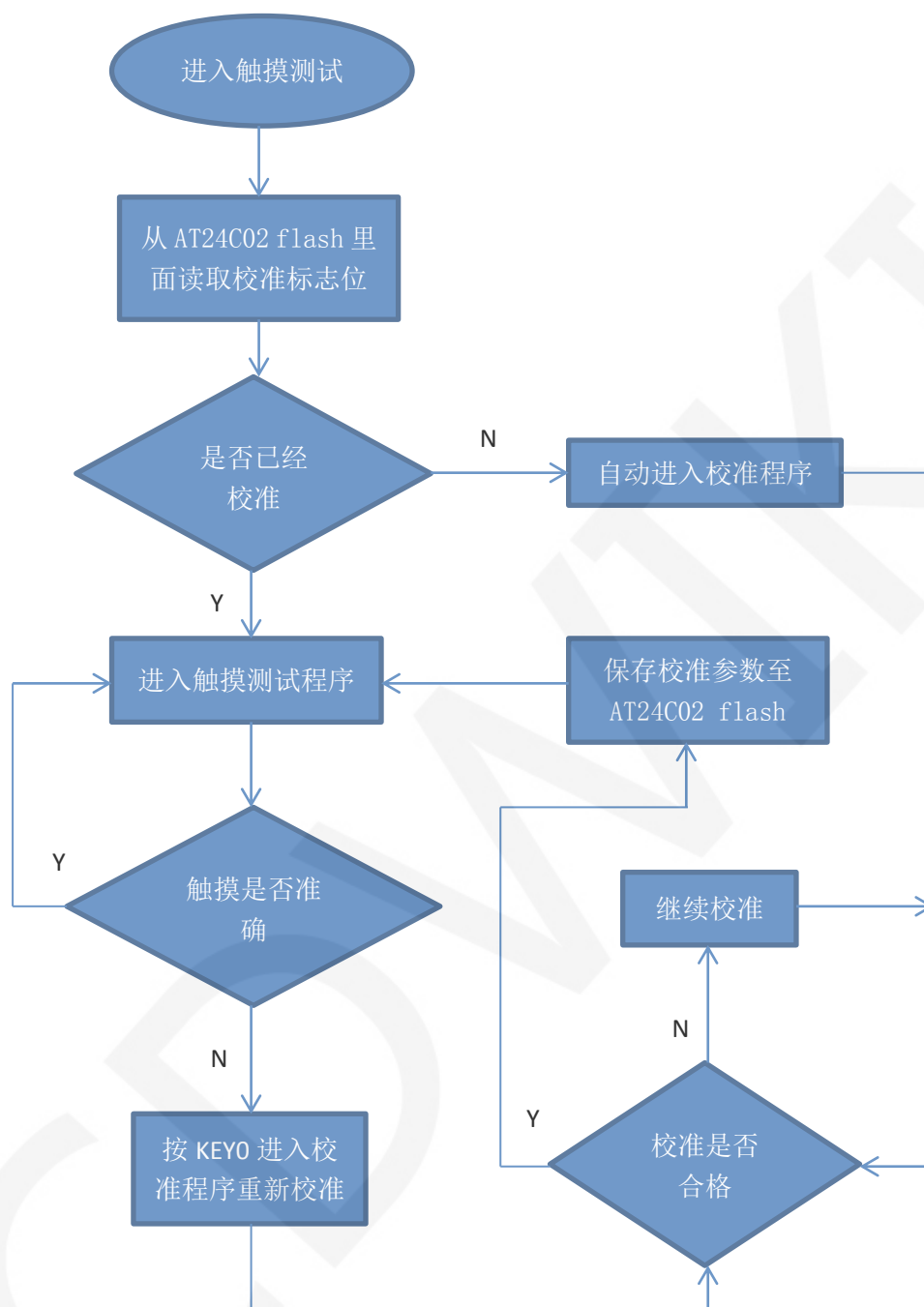
// A lookup table is used to set the different bit patterns, this uses 1kByte of RAM
#define set_mask(C) set_mask_buffer[C] // 63fps Sprite rendering test 33% faster, graphicstest only 1.8% faster than shifting in real time

// Write 8 bits to TFT
#define write8(C) {GPIO.out_wlts = clr_mask; GPIO.out_wlts = set_mask((uint8_t)C); WR_STROBE;}
#define read8(dst)
```

4、触摸屏校准说明

A、STM32 测试程序触摸屏校准说明

STM32 触摸屏校准程序可以自动识别是否需要校准或者手动通过按键进入校准，此过程包含在触摸屏测试项中，校准标志和校准参数保存在 AT24C02 flash 里，需要时要从 flash 里面读取，校准流程如下图所示：



B、Arduino 测试程序触摸屏校准说明

Arduino 触摸屏校准需要先运行 touch_screen_calibration 程序, 然后根据提示进行校准, 校准合格后, 需要将屏幕显示的校准参数写入 LCDWIKI_TOUCH 库的 cali_para.h 文件里面, 如下图所示:

```

3:
4: #define XFAC      852
5: #define XOFFSET   (-14)
6: #define YFAC      1284
7: #define YOFFSET   (-30)
8:

```

常用软件

本套测试示例需要显示中英文、符号以及图片，所以要用到取模软件。取模软件有两种：Image2Lcd 和 PCtoLCD2002。这里只针对该套测试程序说明一下取模软件的设置。

PCtoLCD2002 取模软件设置如下：

点阵格式选择**阴码**

取模方式选择**逐行式**

取模走向选择**顺向（高位在前）**

输出数制选择**十六进制数**

自定义格式选择**C51 格式**

具体设置方法见如下网页：

<http://www.lcdwiki.com/zh/%E3%80%90%E6%95%99%E7%A8%8B%E3%80%91%E4%B8%AD%E8%8B%B1%E6%96%87%E6%98%BE%E7%A4%BA%E5%8F%96%E6%A8%A1%E8%AE%BE%E7%BD%AE>

Image2Lcd 取模软件设置如下图所示：



Image2Lcd 软件需要设置为水平、自左向右、自上向下、低位在前扫描方式。