

# 2.8inch 16BIT Module MRB2801 用户手册

## 产品概述

该产品为一款 2.8 寸 TFT LCD 模块，其拥有 320x240 分辨率，支持 16BIT RGB 65K 色显示，内部驱动 IC 为 ILI9341。其硬件支持 8 位和 16 位并口数据总线模式切换，默认使用 16 位并口数据总线模式。该模块包含有 LCD 显示屏、电阻触摸屏、SD 卡插槽以及 PCB 底板等部件，可以直插到 STM32 系列开发板上使用，还支持 SD 卡扩展功能。

## 产品特点

- 2.8 寸彩屏，支持 16BIT RGB 65K 色显示，显示色彩丰富
- 240x320 分辨率，显示效果清晰
- 支持 8 位和 16 位并行数据总线模式切换，默认采用 16 位并行总线传输，传输速度快
- 支持正点原子 STM32 Mini、精英、战舰、探索者以及阿波罗开发板程序和直插式使用
- 支持触摸功能
- 支持 SD 卡功能扩展
- 提供丰富的 STM32 和 C51 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	RGB 65K 彩色
SKU	MRB2801
屏幕尺寸	2.8(inch)
屏幕类型	TFT
驱动芯片	ILI9341
分辨率	320*240 (Pixel)
模块接口	8bit or 16Bit parallel interface
有效显示区域	43.2x57.6 (mm)

触摸屏类型	电阻触摸屏
触摸 IC	XPT2046
模块尺寸	51.00x82.55 (mm)
工作温度	-20℃~70℃
存储温度	-40℃~70℃
工作电压	3.3V / 5V
功耗	待定
产品重量	待定

## 接口说明

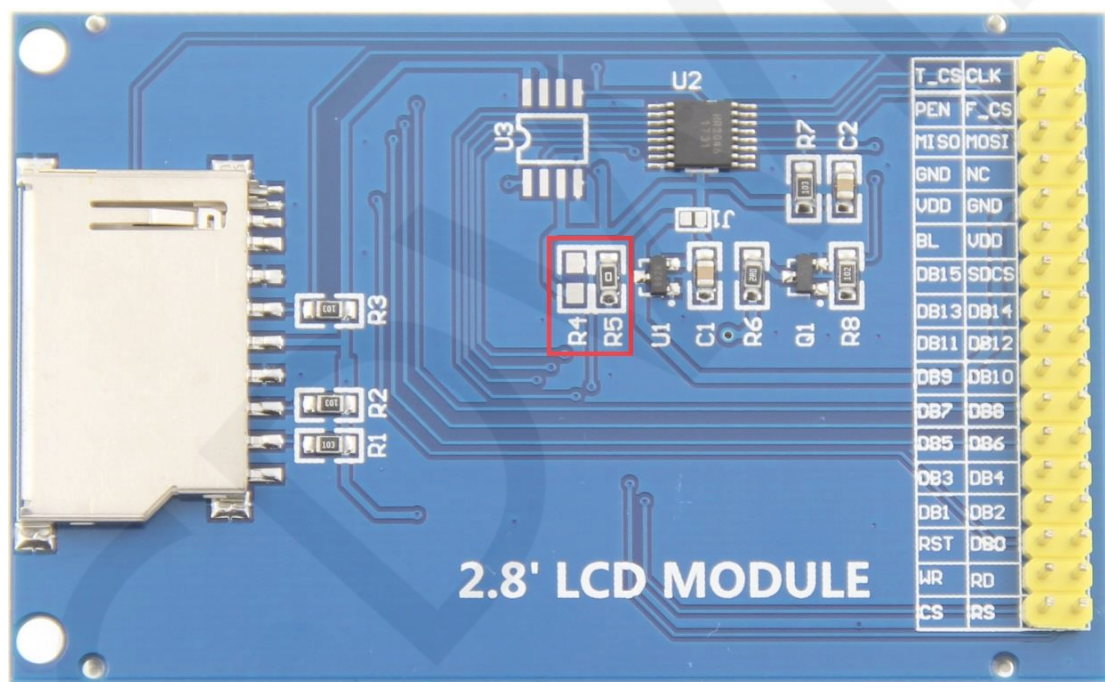


图 1. 模块引脚丝印图

### 注意：

1. 该模块硬件支持8位和16位并口数据总线模式切换（如上面图1中红框所示），具体说明如下：
  - A. 将R5焊接0欧电阻或者直接短接，并将R4断开：选择16位数据总线模式（默认），使用DB0~DB15数据引脚

B. 将R4焊接0欧电阻或者直接短接，并将R5断开：选择8位数据总线模式，使用

### DB8~DB15数据引脚

#### 重要说明：

1. 以下引脚序号 1~34 是指我司带 PCB 底板的模块排针引脚编号，如果您购买的是裸屏，请参考裸屏规格书的引脚定义，按照信号类型来参考接线而不是直接根据下面的模块引脚编号来接线，举例：CS 在我们模块上是 1 脚，可能在不同尺寸裸屏上是 x 脚。
2. 关于 VCC 供电电压：如果您购买的是带 PCB 底板模块，VCC/VDD 供电可接 5V 或 3.3V（模块已集成超低压差 5V 转 3V 电路），但是建议接3.3V，因为接5V会导致电路发热量增加，影响模块使用寿命；如果您购买的是液晶屏裸屏，切记只能接 3.3V。
3. 关于背光电压：带 PCB 底板的模块均已集成三极管背光控制电路，只需 BL 引脚输入高电平或者PWM波则背光点亮。如果您购买的是裸屏，则LED<sub>Ax</sub> 接 3.0V~3.3V，LED<sub>Kx</sub> 接地即可。

序号	模块引脚	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	DB0	液晶屏数据总线低8位引脚（如果选择8位模式，低8位数据引脚没有使用）
7	DB1	
8	DB2	
9	DB3	
10	DB4	
11	DB5	
12	DB6	
13	DB7	
14	DB8	液晶屏数据总线高8位引脚
15	DB9	
16	DB10	

17	DB11	
18	DB12	
19	DB13	
20	DB14	
21	DB15	
22	SDCS	SD卡片选控制引脚(使用SD卡扩展功能时用到, 本测试程序未用到)
23	BL	液晶屏背光控制引脚(高电平点亮)
24	VDD	模块电源正极引脚(模块已集成稳压IC, 所以电源可接5V也可以接3.3V)
25	VDD	
26	GND	模块电源地引脚
27	GND	
28	NC	无定义, 保留
29	MISO	触摸屏SPI总线数据输入引脚
30	MOSI	触摸屏SPI总线数据输出引脚
31	PEN	触摸屏中断检测引脚(发生触摸时为低电平)
32	F_CS	Flash片选控制引脚(使用Flash扩展功能时用到, 本测试程序未用到)
33	T_CS	触摸屏IC片选控制引脚(低电平使能)
34	CLK	触摸屏SPI总线时钟控制引脚

## 硬件配置

该 LCD 模块硬件电路包含五大部分: LCD 显示控制电路、电阻触摸屏采样电路、SD 卡接口电路、数据总线模式切换电路以及背光控制电路。

LCD 显示控制电路用于控制 LCD 的引脚, 包括控制引脚和数据传输引脚。

电阻触摸屏采样电路用于检测触摸事件、将触摸数据进行 AD 转换、发送触摸坐标值。

SD 卡接口电路用于 SD 卡功能扩展, 控制 SD 卡的识别, 读取及写入。

数据总线模式切换电路用于切换 8 位或者 16 位并口数据总线模式切换。

背光控制电路用于控制背光亮度和供电电源选择。

## 工作原理

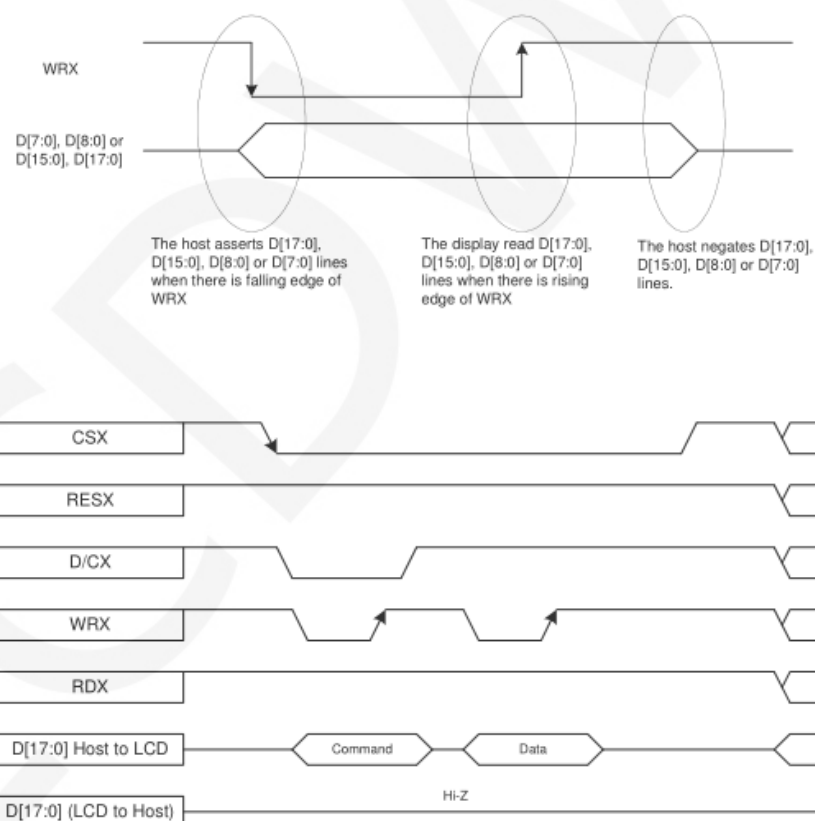
### 1、ITI9341 控制器简介

ITI9341 控制器支持的最大分辨率为 240\*320, 拥有一个 172800 字节大小的 GRAM。同时支持 8 位、9 位、16 位、18 位并口数据总线, 还支持 3 线制和 4 线制 SPI 串口。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。ITI9341 还支持 65K、262K 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。

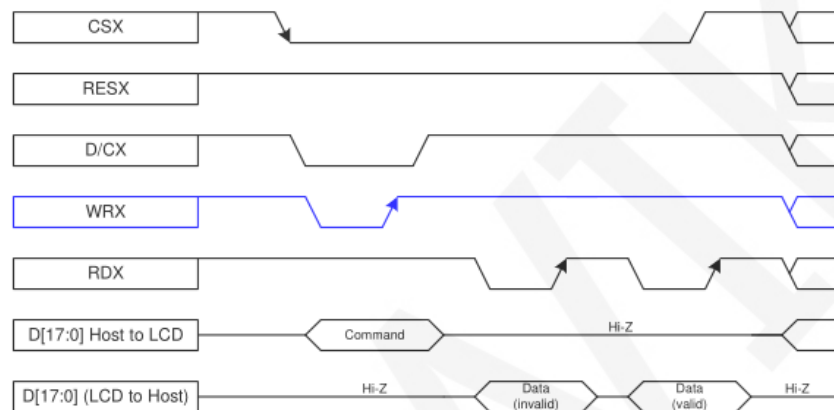
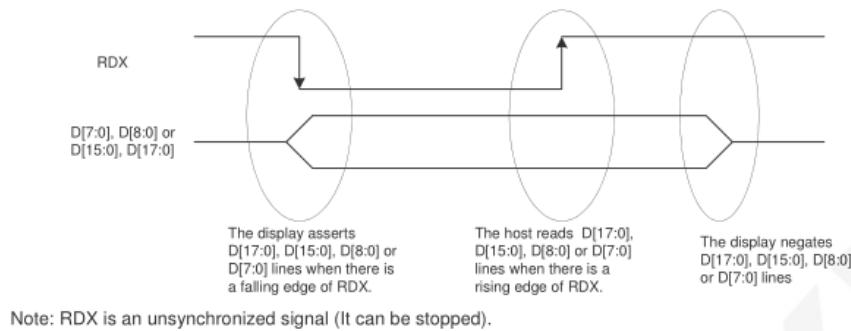
ITI9341 控制器使用 16bit (RGB565) 来控制一个像素点显示, 因此可以每个像素点显示颜色多达 65K 种。像素点地址设置按照行列的顺序进行, 递增递减方向由扫描方式决定。ITI9341 显示方法按照先设置地址再设置颜色值进行。

### 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。

**注意：**

1. 本模块可以直插到正点原子开发板的TFTLCD插槽中使用，不需要手动接线；
2. 以下直插说明中对应单片机内部连接引脚是指TFTLCD插槽在开发板内部所直连的单片机引脚，仅供参考。

MiniSTM32开发板TFTLCD插槽直插说明			
序号	模块引脚	对应TFTLCD插槽直插引脚	对应STM32F103RCT6单片机内部连接引脚
1	CS	CS	PC9
2	RS	RS	PC8
3	WR	WR	PC7
4	RD	RD	PC6
5	RST	RST	PC4
6	DB0	D0	PB0
7	DB1	D1	PB1
8	DB2	D2	PB2
9	DB3	D3	PB3
10	DB4	D4	PB4
11	DB5	D5	PB5
12	DB6	D6	PB6
13	DB7	D7	PB7
14	DB8	D8	PB8
15	DB9	D9	PB9
16	DB10	D10	PB10
17	DB11	D11	PB11
18	DB12	D12	PB12
19	DB13	D13	PB13



20	DB14	D14	PB14
21	DB15	D15	PB15
22	SDCS	没使用	GND
23	BL	BL	PC10
24	VDD	3.3	3.3V
25	VDD	3.3	3.3V
26	GND	GND	GND
27	GND	GND	GND
28	NC	没使用	5V
29	MISO	MISO	PC2
30	MOSI	MOSI	PC3
31	PEN	PEN	PC1
32	F_CS	没使用	NC
33	T_CS	TCS	PC13
34	CLK	CLK	PC0

### Elite STM32开发板TFTLCD插槽直插说明

序号	模块引脚	对应TFTLCD 插槽直插引脚	对应STM32F103ZET6单片机内 部连接引脚
1	CS	CS	PG12
2	RS	RS	PG0
3	WR	WR	PD5
4	RD	RD	PD4
5	RST	RST	复位引脚
6	DB0	D0	PD14
7	DB1	D1	PD15
8	DB2	D2	PD0
9	DB3	D3	PD1
10	DB4	D4	PE7
11	DB5	D5	PE8
12	DB6	D6	PE9
13	DB7	D7	PE10

14	DB8	D8	PE11
15	DB9	D9	PE12
16	DB10	D10	PE13
17	DB11	D11	PE14
18	DB12	D12	PE15
19	DB13	D13	PD8
20	DB14	D14	PD9
21	DB15	D15	PD10
22	SDCS	没使用	GND
23	BL	BL	PB0
24	VDD	VDD	3.3V
25	VDD	VDD	3.3V
26	GND	GND	GND
27	GND	GND	GND
28	NC	没使用	5V
29	MISO	MISO	PB2
30	MOSI	MOSI	PF9
31	PEN	PEN	PF10
32	F_CS	没使用	NC
33	T_CS	TCS	PF11
34	CLK	CLK	PB1

### WarShip STM32开发板TFTLCD插槽直插说明

序号	模块引脚	对应TFTLCD 插槽直插引脚	对应STM32F103ZET6单片机内 部连接引脚	
			V2	V3
1	CS	CS	PG12	
2	RS	RS	PG0	
3	WR	WR	PD5	
4	RD	RD	PD4	
5	RST	RST	复位引脚	
6	DB0	D0	PD14	

7	DB1	D1	PD15
8	DB2	D2	PD0
9	DB3	D3	PD1
10	DB4	D4	PE7
11	DB5	D5	PE8
12	DB6	D6	PE9
13	DB7	D7	PE10
14	DB8	D8	PE11
15	DB9	D9	PE12
16	DB10	D10	PE13
17	DB11	D11	PE14
18	DB12	D12	PE15
19	DB13	D13	PD8
20	DB14	D14	PD9
21	DB15	D15	PD10
22	SDCS	没使用	GND
23	BL	BL	PB0
24	VDD	VDD	3.3V
25	VDD	VDD	3.3V
26	GND	GND	GND
27	GND	GND	GND
28	NC	没使用	5V
29	MISO	MISO	PF8 PB2
30	MOSI	MOSI	PF9
31	PEN	PEN	PF10
32	F_CS	没使用	NC
33	T_CS	TCS	PB2 PF11
34	CLK	CLK	PB1

Explorer STM32F4开发板TFTLCD插槽直插说明			
序号	模块引脚	对应TFTLCD 插槽直插引脚	对应STM32F407ZGT6单片机内 部连接引脚
1	CS	CS	PG12
2	RS	RS	PF12
3	WR	WR	PD5
4	RD	RD	PD4
5	RST	RST	复位引脚
6	DB0	D0	PD14
7	DB1	D1	PD15
8	DB2	D2	PD0
9	DB3	D3	PD1
10	DB4	D4	PE7
11	DB5	D5	PE8
12	DB6	D6	PE9
13	DB7	D7	PE10
14	DB8	D8	PE11
15	DB9	D9	PE12
16	DB10	D10	PE13
17	DB11	D11	PE14
18	DB12	D12	PE15
19	DB13	D13	PD8
20	DB14	D14	PD9
21	DB15	D15	PD10
22	SDCS	没使用	GND
23	BL	BL	PB15
24	VDD	VDD	3.3V
25	VDD	VDD	3.3V
26	GND	GND	GND
27	GND	GND	GND
28	NC	没使用	5V

29	MISO	MISO	PB2
30	MOSI	MOSI	PF11
31	PEN	PEN	PB1
32	F_CS	没使用	NC
33	T_CS	TCS	PC13
34	CLK	CLK	PB0

### Apollo STM32F4/F7开发板TFTLCD插槽直插说明

序号	模块引脚	对应TFTLCD 插槽直插引脚	对应STM32F429IGT6、 STM32F767IGT6、 STM32H743IIT6 单片机内部连接引脚
1	CS	CS	PD7
2	RS	RS	PD13
3	WR	WR	PD5
4	RD	RD	PD4
5	RST	RST	复位引脚
6	DB0	D0	PD14
7	DB1	D1	PD15
8	DB2	D2	PD0
9	DB3	D3	PD1
10	DB4	D4	PE7
11	DB5	D5	PE8
12	DB6	D6	PE9
13	DB7	D7	PE10
14	DB8	D8	PE11
15	DB9	D9	PE12
16	DB10	D10	PE13
17	DB11	D11	PE14
18	DB12	D12	PE15
19	DB13	D13	PD8
20	DB14	D14	PD9

21	DB15	D15	PD10
22	SDCS	没使用	GND
23	BL	BL	PB5
24	VDD	VDD	3.3V
25	VDD	VDD	3.3V
26	GND	GND	GND
27	GND	GND	GND
28	NC	没使用	5V
29	MISO	MISO	PG3
30	MOSI	MOSI	PI3
31	PEN	PEN	PH7
32	F_CS	没使用	NC
33	T_CS	TCS	PI8
34	CLK	CLK	PH6

#### 操作步骤:

- A、按照上述接线说明将 LCD 模块（如图 1 所示）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](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、C51 使用说明

接线说明：

引脚标注见接口说明。

STC12C5A60S2单片机测试程序接线说明		
序号	模块引脚	对应STC12开发板接线引脚
1	CS	P13
2	RS	P12
3	WR	P11
4	RD	P10
5	RST	P33
6	DB0	P00
7	DB1	P01
8	DB2	P02
9	DB3	P03
10	DB4	P04
11	DB5	P05
12	DB6	P06
13	DB7	P07
14	DB8	P20
15	DB9	P21
16	DB10	P22
17	DB11	P23
18	DB12	P24
19	DB13	P25
20	DB14	P26
21	DB15	P27
22	SDCS	不需要接
23	BL	P32

24	VDD	3.3V/5V
25	VDD	3.3V/5V
26	GND	GND
27	GND	GND
28	NC	不需要接
29	MISO	P35
30	MOSI	P34
31	PEN	P40
32	F_CS	不需要接
33	T_CS	P37
34	CLK	P36

STC89C52RC单片机测试程序接线说明		
序号	模块引脚	对应STC89开发板接线引脚
1	CS	P13
2	RS	P12
3	WR	P11
4	RD	P10
5	RST	P14
6	DB0	P30
7	DB1	P31
8	DB2	P32
9	DB3	P33
10	DB4	P34
11	DB5	P35
12	DB6	P36
13	DB7	P37
14	DB8	P20
15	DB9	P21
16	DB10	P22
17	DB11	P23



18	DB12	P24
19	DB13	P25
20	DB14	P26
21	DB15	P27
22	SDCS	不需要接
23	BL	3.3V
24	VDD	3.3V/5V
25	VDD	3.3V/5V
26	GND	GND
27	GND	GND
28	NC	不需要接
29	MISO	不需要接
30	MOSI	不需要接
31	PEN	不需要接
32	F_CS	不需要接
33	T_CS	不需要接
34	CLK	不需要接

**注意：**

1. 由于STC89C52RC单片机没有推挽输出功能，所以背光控制引脚需要接3.3V电源才能正常点亮。
2. 由于STC89C52RC单片机Flash容量太小（小于25KB），无法下载带触摸功能的程序，所以触摸屏不需要接线。

**操作步骤：**

- A、按照上述接线说明将 LCD 模块（如图 1 所示）和 C51 单片机连接起来，并上电；
- B、选择需要测试的 C51 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 C51 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/C51\\_Keil%26stc-isp\\_Use\\_Illustration\\_CN.pdf](http://www.lcdwiki.com/res/PublicFile/C51_Keil%26stc-isp_Use_Illustration_CN.pdf)

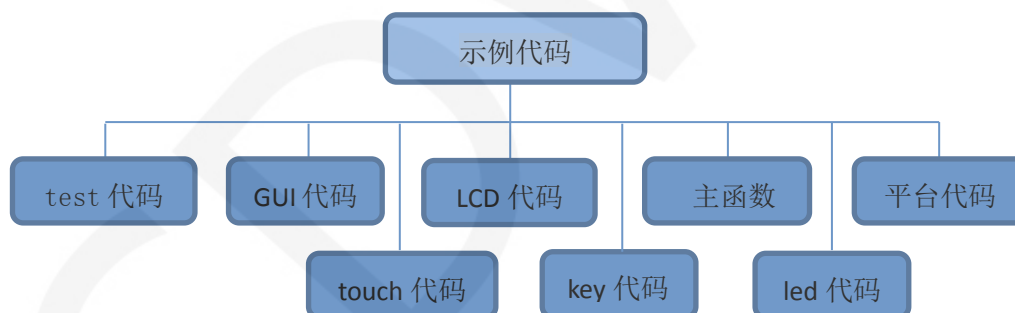
D、LCD 模块如果正常显示字符和图形，则说明程序运行成功；

## 软件说明

### 1、代码架构

#### A、C51 和 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；

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；

按键处理相关的代码都包含在 key 代码中 (C51 平台没有按键处理代码)；

led 配置操作相关的代码都包含在 led 代码中(C51 平台没有 led 处理代码);

## 2、GPIO 定义说明

### A、STM32 测试程序 GPIO 定义说明

STM32 测试程序的 LCD 屏的 GPIO 定义放在 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  
  
//QDtech全系列模块采用了三极管控制背光亮灭, 用户也可以接PWM调节背光亮度  
#define LCD_LED  PBout(LED) //LCD背光  
//如果使用官方库函数定义下列底层, 速度将会下降到14帧每秒, 建议采用我司推荐方法  
//以下IO定义直接操作寄存器, 快速IO操作, 刷屏速率可以达到28帧每秒!  
  
//LCD地址结构体  
typedef struct  
{  
    vu16 LCD_REG;  
    vu16 LCD_RAM;  
} LCD_TypeDef;  
  
//使用NOR/SRAM的 Bank1.sector4,地址位HADDR[27,26]=11 A10作为数据命令区分线  
//注意设置时STM32内部会右移一位对其!  
#define LCD_BASE      ((u32)(0x6C000000 | 0x000007FE))  
#define LCD            ((LCD_TypeDef *) LCD_BASE)
```

STM32 触摸屏相关的 GPIO 定义放在 touch.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
```

## B、C51 测试程序 GPIO 定义说明

C51 测试程序 lcd 屏 GPIO 定义放在 lcd.h 文件里，如下图所示（以 STC12C5A60S2 单片机测试程序为例）：

```
//IO连接
#define LCD_DataPortH P2 //高8位数据口,8位模式下只使用高8位
#define LCD_DataPortL P0 //低8位数据口,8位模式下低8位可以不接线
sbit LCD_RS = P1^2; //数据/命令切换
sbit LCD_WR = P1^1; //写控制
sbit LCD_RD = P1^0; //读控制
sbit LCD_CS = P1^3; //片选
sbit LCD_RESET = P3^3; //复位
sbit LCD_BL=P3^2; //背光控制，如果不需要控制，接3.3v
```

并口引脚定义需要选择整套 GPIO 口组，如 P0，P2 等，这样传输数据时，操作方便。

其他引脚可以定义成任何空闲的 GPIO。

触摸屏相关的 GPIO 定义放在 touch.h 文件里面，如下图所示（以 STC12C5A60S2 单片机测试程序为例）：

```
//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
```

触摸屏的 GPIO 定义都可以修改，可以定义成其他任何空闲的 GPIO。

如果单片机没有 P4 GPIO 组，可以把 penirq 定义成其他 GPIO。

## 3、并口通信代码实现

### A、STM32 测试程序并口通信代码实现

STM32 测试程序并口通信代码都放在 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 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

I/O 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;
    #if LCD_USE8BIT_MODEL
    return (data>>8);
    #else
    return data;
    #endif
}

/*****
 * @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)
{
    #if LCD_USE8BIT_MODEL
    LCD->LCD_REG=(data<<8); //写入要写的寄存器序号
    #else
    LCD->LCD_REG=data; //写入要写的寄存器序号
    #endif
}

/*****
 * @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)
{
    #if LCD_USE8BIT_MODEL
    LCD->LCD_RAM=(data<<8); //写入要写的寄存器序号
    #else
    LCD->LCD_RAM=data; //写入要写的寄存器序号
    #endif
}
```

都实现了 8、16 位命令写入以及 8、16 位数据写入和读取。

#### B、C51 测试程序并口通信代码实现

相关的代码在 LCD.c 文件里实现，如下图所示：

```
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;
}

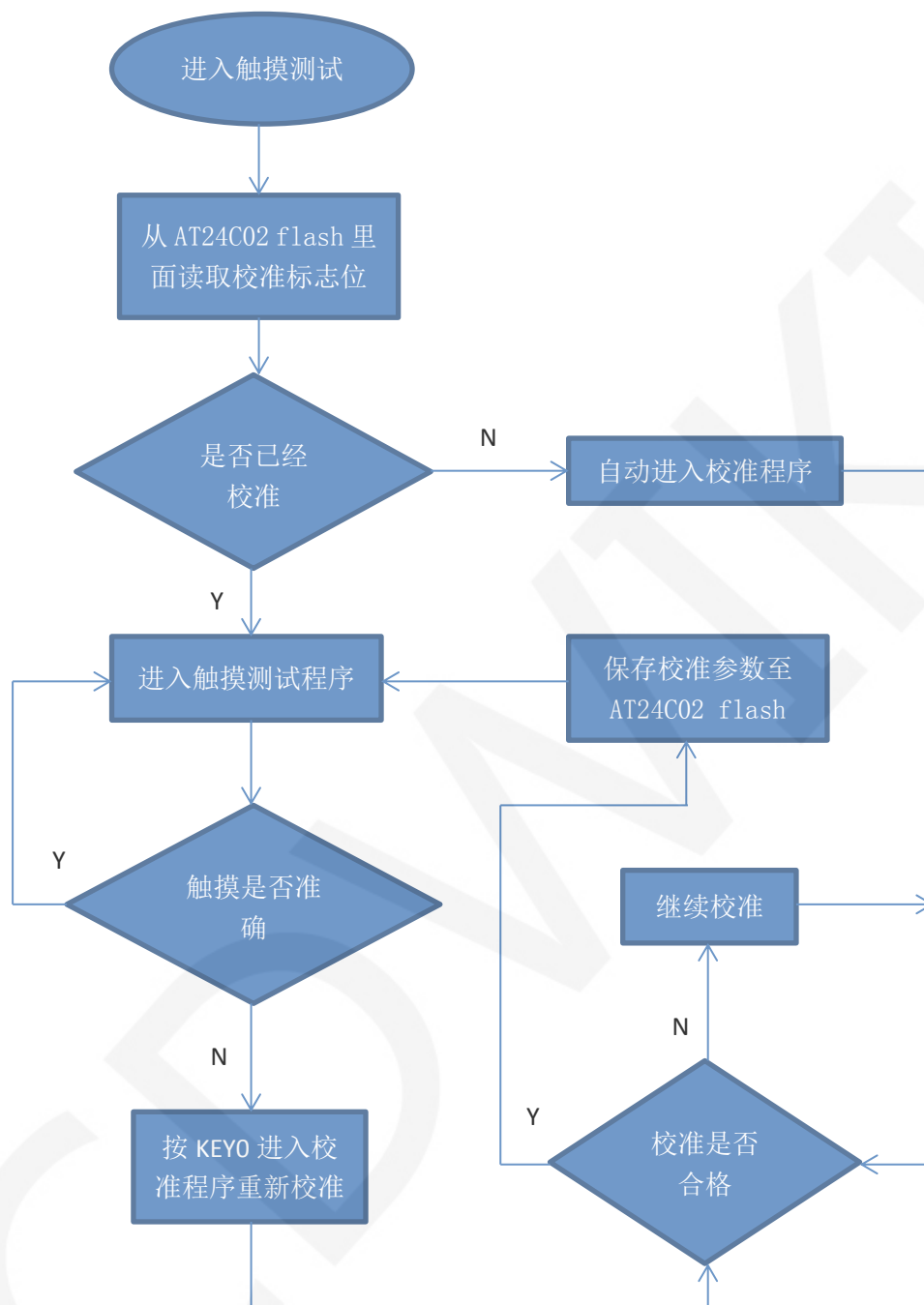
u16 LCD_read(void)
{
    u16 d;
    LCD_CS = 0;
    LCD_RD = 0;
    delay_us(1); //delay 1 us
    d = LCD_DataPortH;
    d = (d<<8) | LCD_DataPortL;
    LCD_RD = 1;
    LCD_CS = 1;
    return d;
}
```

实现了 8、16 位命令写入以及 8、16 位数据写入和读取。

### 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、C51 测试程序触摸屏校准说明

C51 的触摸屏校准需要执行 Touch\_Adjust 测试项(只有 STC12C5A60S2 测试程序才有),

如下图所示:

```
//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
}
```

触摸校准合格后，需要将屏幕显示的校准参数保存在 touch.c 文件中，如下图所示：

```
/**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
```

## 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：



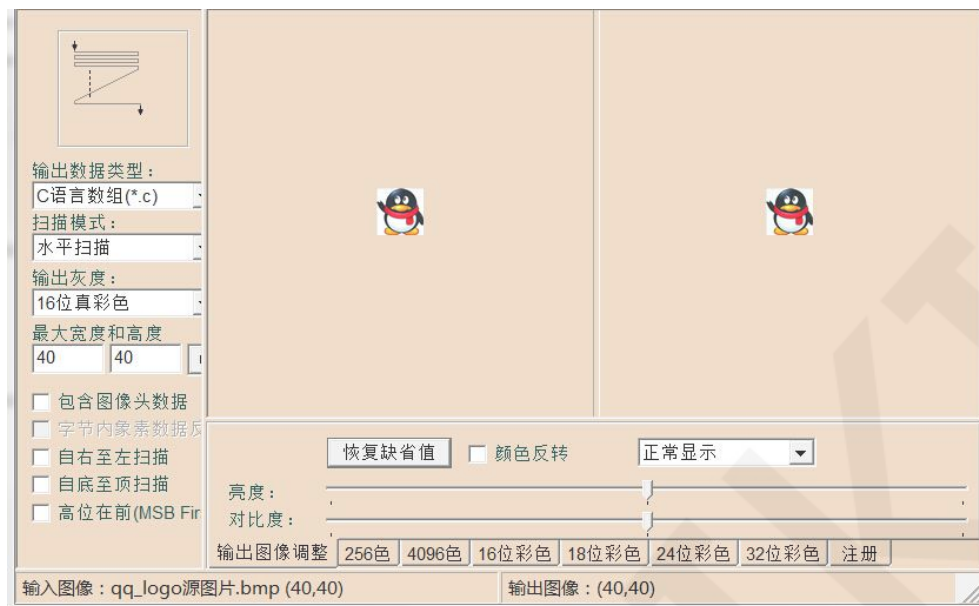


Image2Lcd 软件需要设置为水平、自左向右、自上向下、低位在前扫描方式。