

3.95inch Arduino 8BIT Module MAR3954 用户手册

产品概述

该产品为一款 3.95 寸 TFT LCD 模块，拥有 480x320 分辨率，支持 16BIT RGB 65K 色显示，内部驱动 IC 为 ST7796S，采用 8 位并口通信方式。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，支持 SD 卡扩展功能，可以直插到 Arduino UNO 和 MEGA2560 开发板上使用，还可以用于 C51 和 STM32 平台。

产品特点

- 3.95 寸彩屏，支持 16BIT RGB 65K 色显示，显示色彩丰富
- 480x320 分辨率，显示效果清晰
- 采用 8 位并行总线传输，传输速度快
- 板载 5V/3.3V 电平转换 IC，兼容 5V/3.3V 工作电压
- 支持 Arduino UNO 和 MAGE2560 直插式使用
- 支持触摸功能
- 支持 SD 卡功能扩展
- 提供 Arduino 库和丰富的示例程序
- 可用于 C51 和 STM32 平台并提供丰富的示例程序
- 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	MAR3954
尺寸	3.95(inch)
类型	TFT
驱动芯片	ST7796S
分辨率	480*320 (Pixel)

模块接口	8Bit parallel interface
有效显示区域	83.52x55.68(mm)
模块尺寸	96.52x61.47 (mm)
工作温度	-10℃~60℃
存储温度	-20℃~70℃
工作电压	3.3V / 5V
功耗	待定
产品重量	待定

接口说明

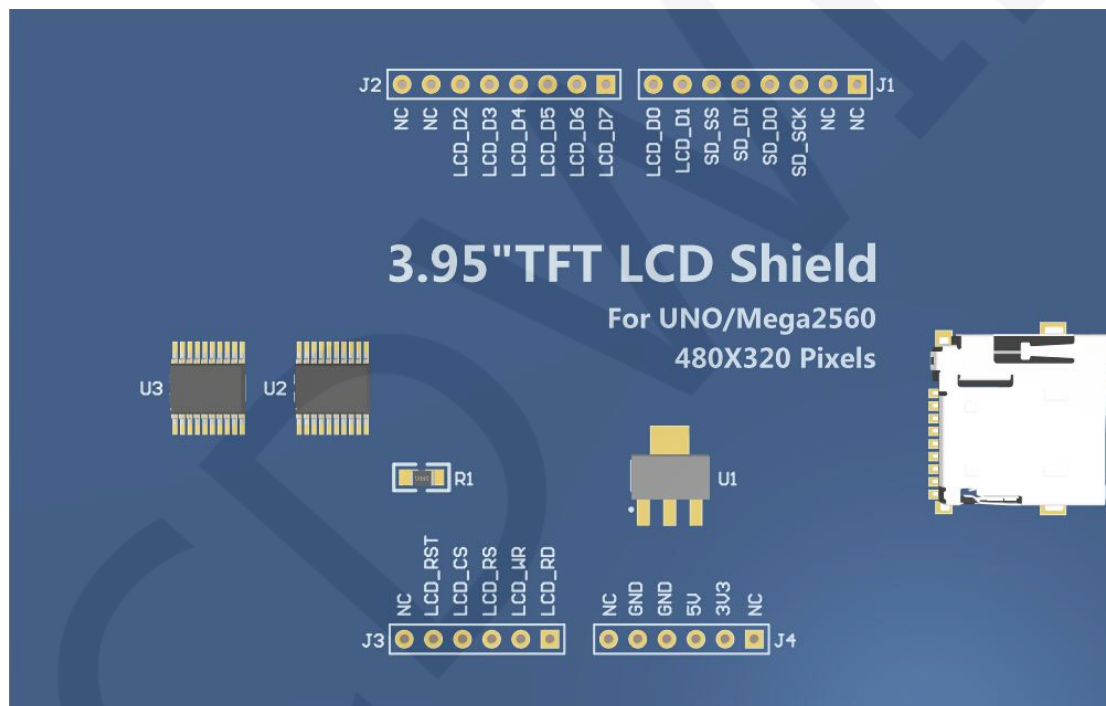


图 1. 模块引脚丝印图

重要说明:

1. 以下引脚序号1~20是指我司带PCB底板的模块引脚编号，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：LCD_CS在我们模块上是13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供电需要接5V（模块已集

成超低压差5V转3.3V电路），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底板的模块均已接入3.3V，不需要再手动接入。如果您购买的是裸屏，则 LEDA接3.0V-3.3V，LEDKx接地即可。

序号	模块引脚	引脚说明
1	5V	电源正极5V引脚
2	3V3	电源正极3.3V引脚
3	GND	电源地引脚
4	LCD_D0	8位数据总线引脚
5	LCD_D1	
6	LCD_D2	
7	LCD_D3	
8	LCD_D4	
9	LCD_D5	
10	LCD_D6	
11	LCD_D7	
12	LCD_RST	液晶屏复位控制引脚
13	LCD_CS	液晶屏片选控制引脚
14	LCD_RS	液晶屏寄存器/数据选择控制引脚
15	LCD_WR	液晶屏写控制引脚
16	LCD_RD	液晶屏读控制引脚
17	SD_SS	扩展功能：SD卡片选控制引脚
18	SD_DI	扩展功能：SD卡输入引脚
19	SD_DO	扩展功能：SD卡输出引脚
20	SD_SCK	扩展功能：SD卡时钟控制引脚

硬件配置

该 LCD 模块硬件电路包含三大部分：LCD 显示控制电路、电平转换电路、SD 卡控制电路。

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。

电平转换电路用于进行 5V/3.3V 转换，使模块可以兼容 3.3V/5V 电源。

SD 卡控制电路用于 SD 卡功能扩展，控制 SD 卡的识别，读取及写入。

触摸屏控制采用引脚复用的方法。

工作原理

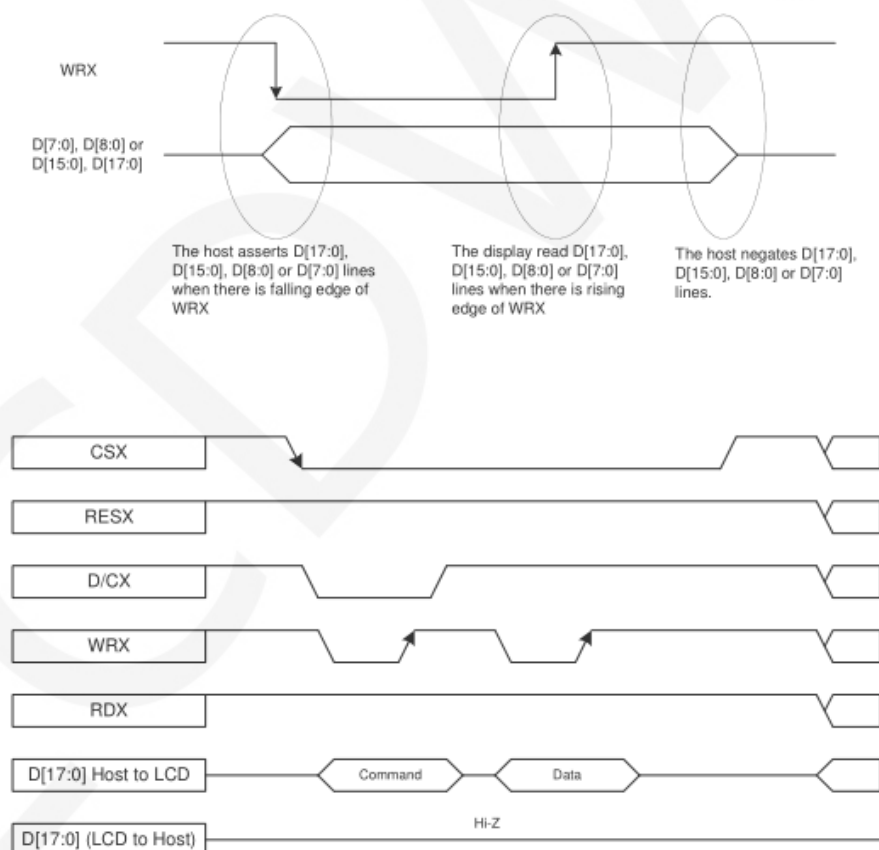
1、ST7796S 控制器简介

ST7796S 是一种用于 262 K 彩色 TFT-LCD 的单片控制器，支持的最大分辨率为 320*480，拥有一个 345600 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。ST7796S 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。

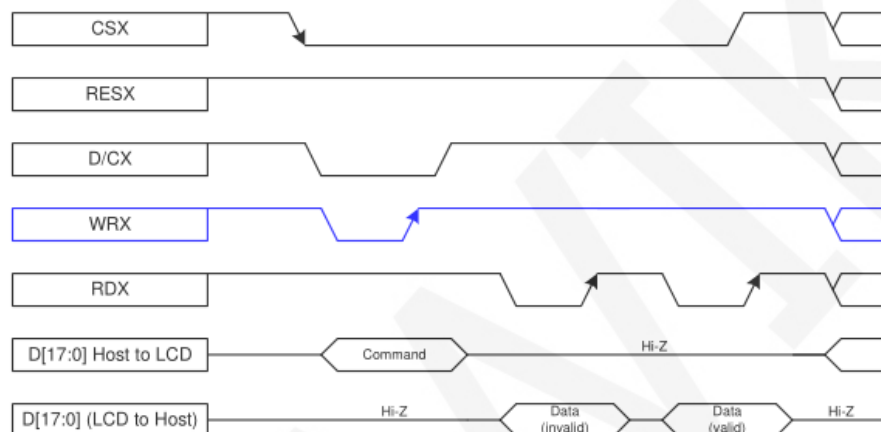
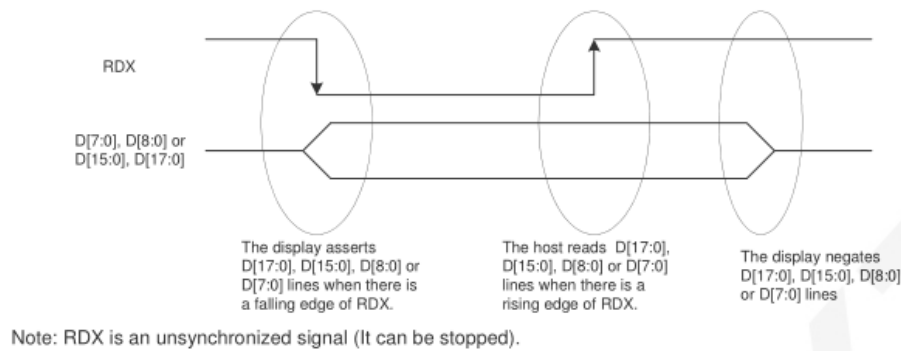
ST7796S 控制器使用 16bit (RGB565) 来控制一个像素点显示，因此可以每个像素点显示颜色多达 65K 种。像素点地址设置按照行列的顺序进行，递增递减方向由扫描方式决定。ST7796S 显示方法按照先设置地址再设置颜色值进行。

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

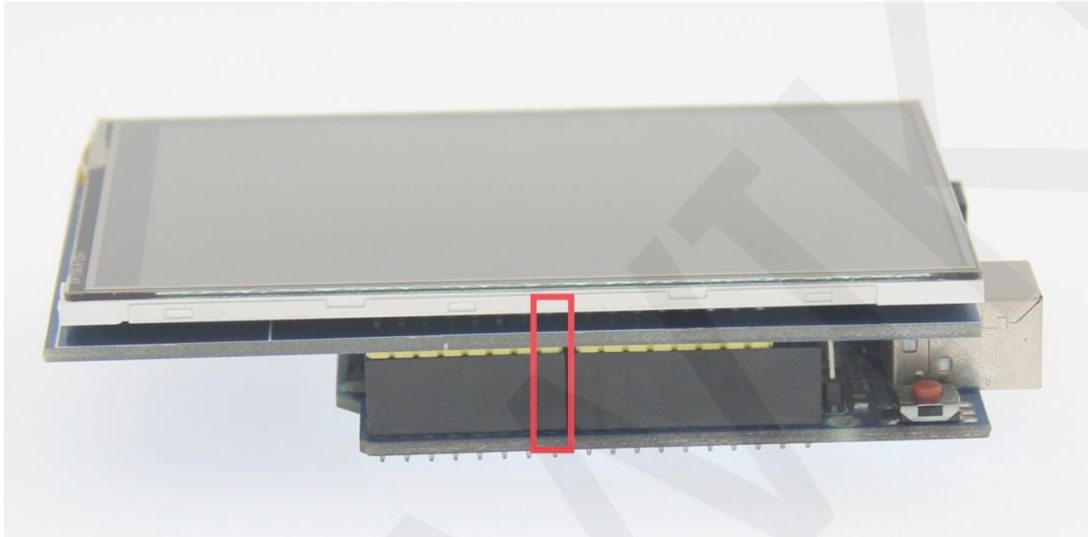
1、Arduino 使用说明

接线说明：

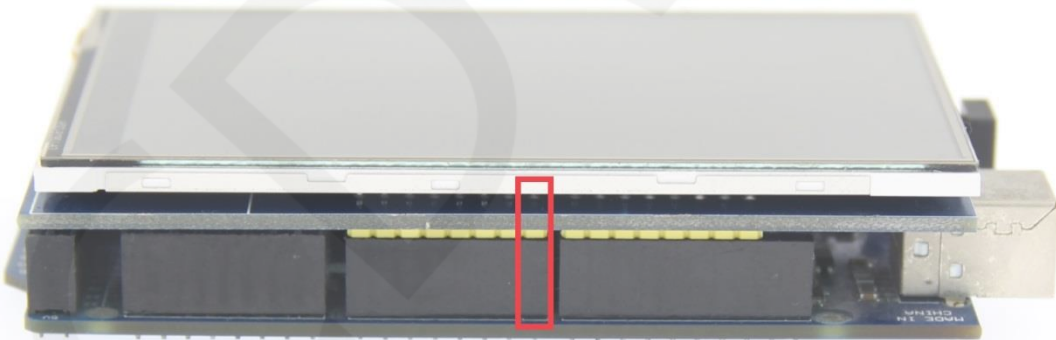
引脚标注见接口说明。

此模块可直接插入 Arduino UNO 和 Mega2560 中使用，不需要再手动接线，如下图所示，

将模块的 **LCD_D7** 引脚对齐 Arduino（UNO 和 Mega2560）的 **7** 引脚，然后直插：



UNO直插图



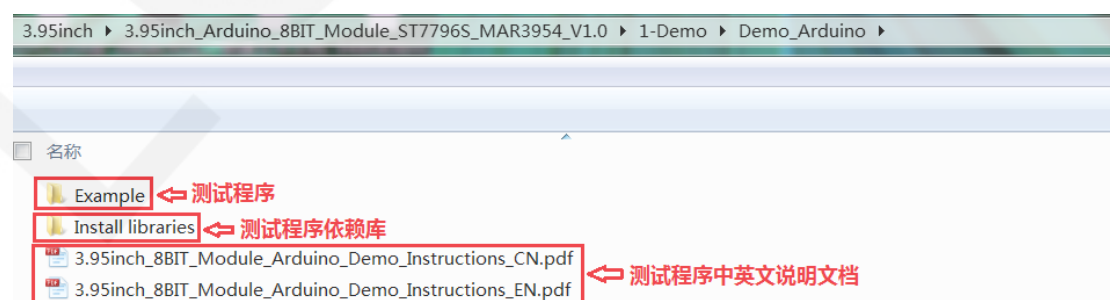
Mega2560直插图

Arduino UNO和MEGA2560单片机测试程序引脚直插说明		
序号	模块引脚	对应UNO和MEGA2560开发板直插引脚
1	5V	5V

2	3V3	3.3V
3	GND	GND
4	LCD_D0	8
5	LCD_D1	9
6	LCD_D2	2
7	LCD_D3	3
8	LCD_D4	4
9	LCD_D5	5
10	LCD_D6	6
11	LCD_D7	7
12	LCD_RST	A4
13	LCD_CS	A3
14	LCD_RS	A2
15	LCD_WR	A1
16	LCD_RD	A0
17	SD_SS	10
18	SD_DI	11
19	SD_DO	12
20	SD_SCK	13

操作步骤:

- 按照上述接线说明将 LCD 模块直插到 Arduino 单片机上，并上电；
- 将测试程序包中 **Install libraries** 目录下的依赖库拷贝到 Arduino 工程目录的 **libraries** 文件夹下（如果不需要依赖库，则不需要拷贝）；
- 选择需要测试的 Arduino 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)



D、打开所选的测试程序，进行编译和下载。

关于 Arduino 测试程序依赖库拷贝、编译和下载的具体操作方法见如下文档：

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

E、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。

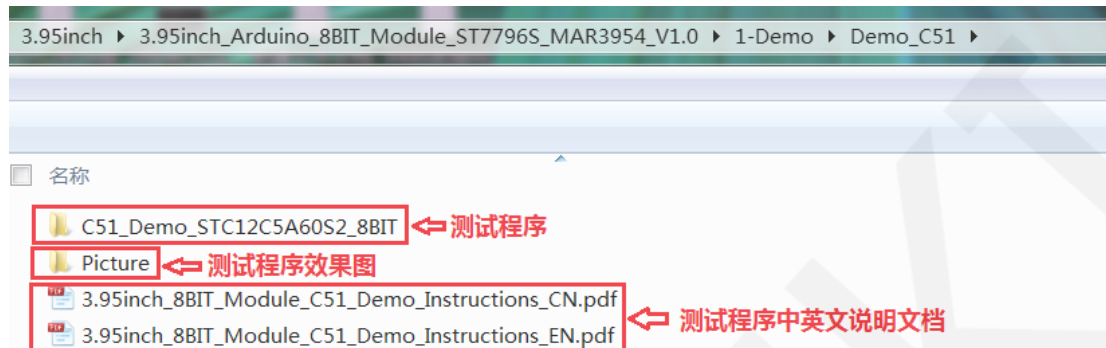
STC12C5A60S2单片机测试程序接线说明		
序号	模块引脚	对应STC12开发板接线引脚
1	5V	5V
2	3V3	3V3
3	GND	GND
4	LCD_D0	P00
5	LCD_D1	P01
6	LCD_D2	P02
7	LCD_D3	P03
8	LCD_D4	P04
9	LCD_D5	P05
10	LCD_D6	P06
11	LCD_D7	P07
12	LCD_RST	P33
13	LCD_CS	P13
14	LCD_RS	P12
15	LCD_WR	P11
16	LCD_RD	P10
17	SD_SS	不需要接
18	SD_DI	不需要接
19	SD_DO	不需要接
20	SD_SCK	不需要接

操作步骤：

A、按照上述接线说明将 LCD 模块和 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

D、LCD 模块如果正常显示字符和图形，则说明程序运行成功；

3、STM32 使用说明

接线说明：

引脚标注见接口说明。

STM32F103RCT6单片机测试程序接线说明		
序号	模块引脚	对应MiniSTM32开发板接线引脚
1	5V	5V
2	3V3	3.3V
3	GND	GND
4	LCD_D0	PB0
5	LCD_D1	PB1
6	LCD_D2	PB2
7	LCD_D3	PB3
8	LCD_D4	PB4
9	LCD_D5	PB5
10	LCD_D6	PB6
11	LCD_D7	PB7
12	LCD_RST	PC10

13	LCD_CS	PC9
14	LCD_RS	PC8
15	LCD_WR	PC7
16	LCD_RD	PC6
17	SD_SS	不需要接
18	SD_DI	不需要接
19	SD_DO	不需要接
20	SD_SCK	不需要接

STM32F103ZET6单片机测试程序接线说明

序号	模块引脚	对应Elite STM32开发板接线引脚
1	5V	5V
2	3V3	3.3V
3	GND	GND
4	LCD_D0	PF0
5	LCD_D1	PF1
6	LCD_D2	PF2
7	LCD_D3	PF3
8	LCD_D4	PF4
9	LCD_D5	PF5
10	LCD_D6	PF6
11	LCD_D7	PF7
12	LCD_RST	PC10
13	LCD_CS	PC9
14	LCD_RS	PC8
15	LCD_WR	PC7
16	LCD_RD	PC6
17	SD_SS	不需要接
18	SD_DI	不需要接
19	SD_DO	不需要接
20	SD_SCK	不需要接

STM32F407ZGT6单片机测试程序接线说明

序号	模块引脚	对应Explorer STM32F4开发板接线引脚
1	5V	5V
2	3V3	3.3V
3	GND	GND
4	LCD_D0	PG0
5	LCD_D1	PG1
6	LCD_D2	PG2
7	LCD_D3	PG3
8	LCD_D4	PG4
9	LCD_D5	PG5
10	LCD_D6	PG6
11	LCD_D7	PG7
12	LCD_RST	PC10
13	LCD_CS	PC9
14	LCD_RS	PC8
15	LCD_WR	PC7
16	LCD_RD	PC6
17	SD_SS	不需要接
18	SD_DI	不需要接
19	SD_DO	不需要接
20	SD_SCK	不需要接

STM32F429IGT6、STM32F7671GT6、STM32H743IIT6 单片机测试程序接线说明

序号	模块引脚	对应Apollo STM32F4/F7开发板接线引脚
1	5V	5V
2	3V3	3.3V
3	GND	GND
4	LCD_D0	PE0
5	LCD_D1	PE1

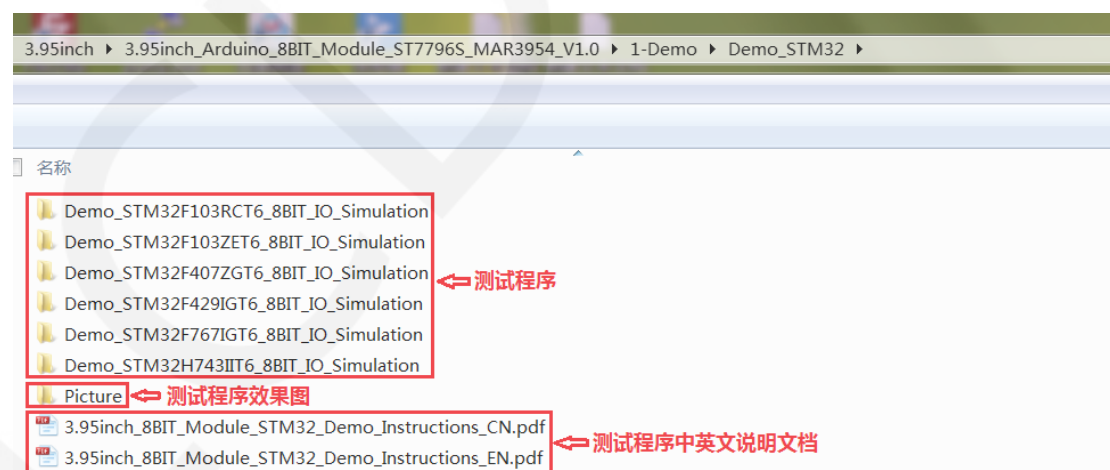
6	LCD_D2	PE2
7	LCD_D3	PE3
8	LCD_D4	PE4
9	LCD_D5	PE5
10	LCD_D6	PE6
11	LCD_D7	PE7
12	LCD_RST	PC10
13	LCD_CS	PC9
14	LCD_RS	PC8
15	LCD_WR	PC7
16	LCD_RD	PC6
17	SD_SS	不需要接
18	SD_DI	不需要接
19	SD_DO	不需要接
20	SD_SCK	不需要接

操作说明:

A、按照上述接线说明将 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

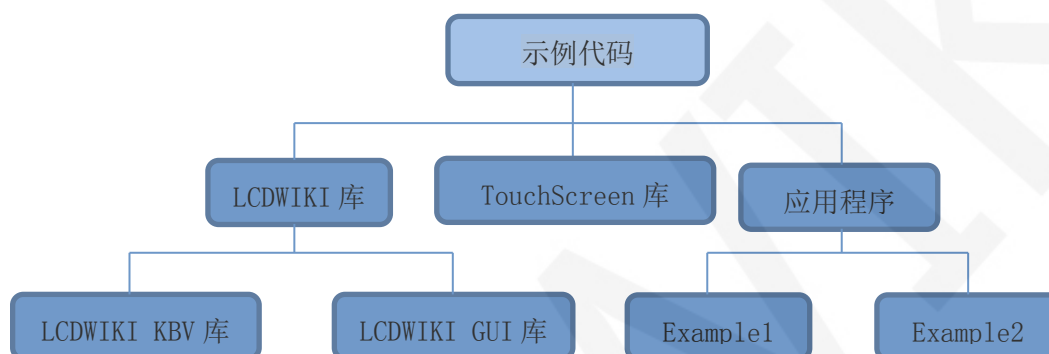
D、LCD 模块如果正常显示字符和图形，则说明程序运行成功；

软件说明

1、代码架构

A、Arduino 代码架构说明

代码架构如下图所示：



Arduino 的测试程序代码由三部分组成：LCDWIKI 库、TouchScreen 库和应用代码；

LCDWIKI 库包含两部分内容：LCDWIKI_KBV 库和 LCDWIKI_GUI 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；

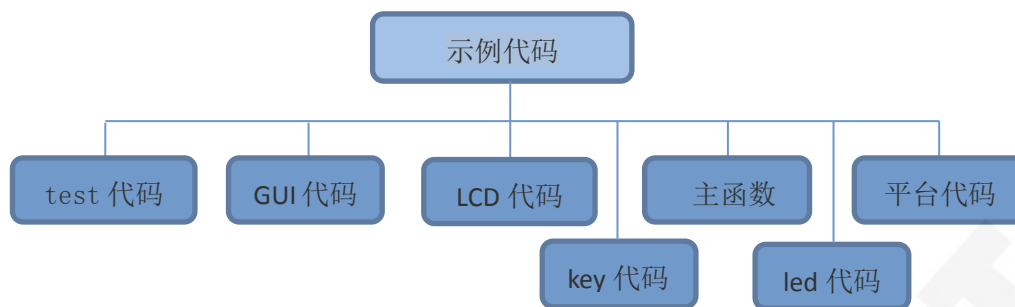
LCDWIKI_GUI 为中间层库，负责使用底层库提供的 API 实现图形的绘制，字符显示；

TouchScreen 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 库和 TouchScreen 库提供的 API，编写一些测试示例，实现某方面的测试功能；

B、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 初始化以及相关的斌并口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;

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;

2、GPIO 定义说明

A、Arduino 测试程序 GPIO 定义说明

模块是直插到 Arduino UNO 和 Mage2560 上使用的, 所以不允许修改 GPIO 口定义。

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。

C、STM32 测试程序 GPIO 定义说明

STM32 IO 模拟测试程序 lcd 屏 GPIO 定义放在 lcd.h 文件里, 如下图所示(以 STM32F103RCT6 测试程序为例):

```

////////////////////////////////////
//-----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-
//举例: 如果是16位模式: DB0-DB7分别接GPIOB_Pin0-GPIOB_Pin7,DB10-DB17对应接
#define DATAOUT(x)  GPIOB->ODR=x; //数据输出
#define DATAIN      GPIOB->IDR;  //数据输入

```

数据并口引脚定义需要选择整套 GPIO 口组, 如 PB, 传输数据时, 操作方便。

其他引脚可以定义成任何空闲的 GPIO。

3、并口通信代码实现

A、Arduino 测试程序并口通信代码实现

相关的代码在 LCDWIKI_KBV 库的 mcu_8bit_magic.h 文件里实现, 如下图所示:

```

#define BMASK      0x03
#define DMASK      0xFC
#define write8(d) { PORTD = (PORTD & ~DMASK) | ((d) & DMASK); PORTB = (PORTB & ~BMASK) | ((d) & BMASK); WR_STROBE; }
// #define write16(d) { uint8_t h = (d)>>8, l = d; write8(h); write8(l); }
#define read8(dst) { RD_ACTIVE; DELAY7; dst = (PIND & DMASK) | (PINB & BMASK); RD_IDLE; }
// #define read16(dst) { uint8_t hi; read8(hi); read8(dst); dst |= (hi << 8); }

#define setWriteDir() { DDRD |= DMASK; DDRB |= BMASK; }
#define setReadDir() { DDRD &= ~DMASK; DDRB &= ~BMASK; }

#define write16(d) { uint8_t h = (d)>>8, l = d; write8(h); write8(l); }
#define read16(dst) { uint8_t hi; read8(hi); read8(dst); dst |= (hi << 8); }

```

实现了 8、16 位命令以及 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 位命令以及 8、16 位数据写入和读取。

C、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 文件里实现。

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;
}
```

都实现了 8、16 位命令以及 8、16 位数据写入和读取。

4、触摸屏校准说明

A、Arduino 测试程序触摸屏校准说明

Arduino 触摸屏校准需要先运行 TouchScreen_Calibr 程序（见测试程序目录），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相应的测试程序里，如下图所示（以 touch_pen 测试程序为例）：

```
//param calibration from kbv
#define TS_MINX 124 ➡ LEFT
#define TS_MAXX 906 ➡ RT

#define TS_MINY 83 ➡ TOP
#define TS_MAXY 893 ➡ BOT
```

B、C51 和 STM32 测试程序触摸屏校准说明

由于此模块不含专门的触摸 IC，所以在 C51 和 STM32 上实现触摸功能比较困难。
故 C51 和 STM32 测试程序没有触摸屏测试项。

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。