

# 3.5inch Arduino 16BIT Module MAR3515 User Manual

## Product Description

The Arduino Mega2560 module is a 3.5-inch TFT LCD module with 480x320 resolution and 65K color display. It uses 16-bit line parallel port communication, and the driver IC is ST7796S. The module includes an LCD display, 5V ~ 3.3V level to the circuit, can be directly plugged into the Arduino mega2560 development board, also supports SD card and SPI Flash function expansion.

## Product Features

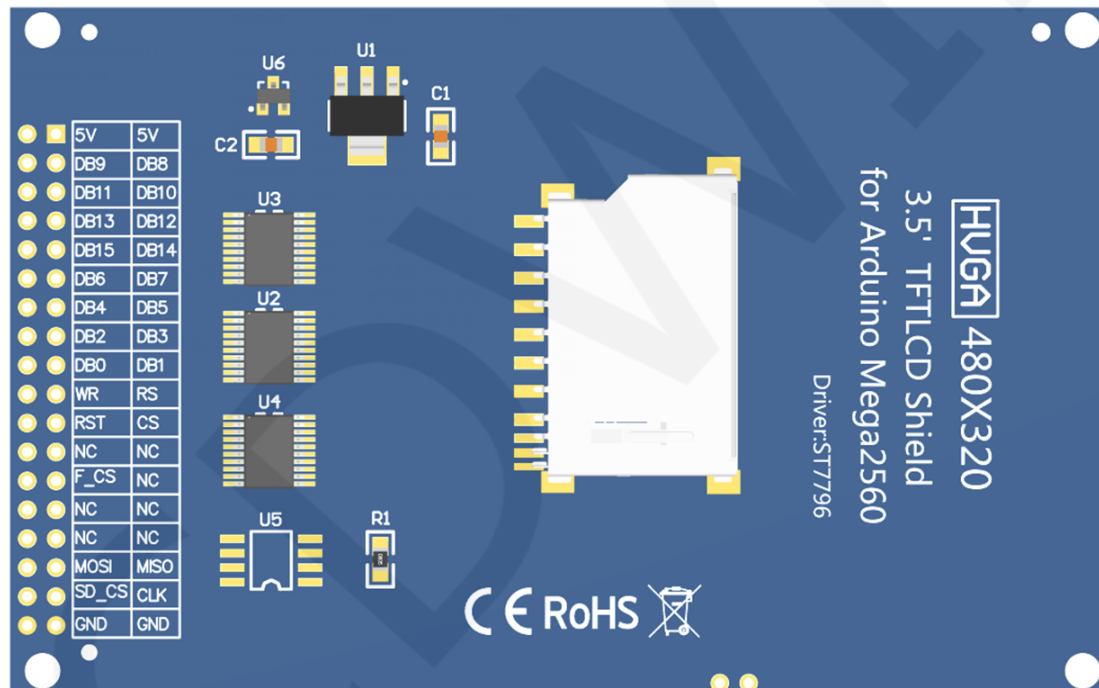
- 3.5-inch color screen, support 16BIT RGB 65K color display, display rich colors
- 320x480 HD resolution for clear display
- 16-bit parallel bus transmission for fast transfer speed
- On-board 5V/3.3V level-shifting IC compatible with 5V/3.3V operating voltage
- Support Arduino Mage2560 for direct plug-in use
- Provide Arduino libraries and rich sample programs
- Support SD card and SPI Flash function extension
- Military-grade process standards, long-term stable work
- Provide underlying driver technical support

## Product Parameters

Name	Description
Display Color	RGB 65K color
SKU	MAR3515
Screen Size	3.5(inch)
Type	TFT
Driver IC	ST7796S
Resolution	480*320 (Pixel)
Module Interface	16Bit parallel interface

Active Area	48.96x73.44 (mm)
Module PCB Size	60.30x96.60 (mm)
Back Light	6 chip HighLight white LEDs
Operating Temperature	-10°C~60°C
Storage Temperature	-20°C~70°C
Operating Voltage	3.3V / 5V
Power Consumption	TDB
Product Weight	About 45(g)

## Interface Description



Module pin Silkscreen picture

### Note:

1. The pins labeled NC in figure 1 are not used;

### Important Note:

1. The following pin numbers 1~30 are the pin number of Module pin with PCB backplane of our company. If you purchase a bare screen,

please refer to the pin definition of the bare screen specification, refer to the wiring according to the signal type instead of directly Wire according to the following module pin numbers. For example: LCD\_CS is 20 pin on our module. It may be x pin on different size bare screen. The following wiring program instructions tell you to connect LCD\_CS signal to the 40 pin of Arduino microcontroller.

2. About VCC supply voltage: If you purchase a module with PCB backplane, VCC/VDD can be connected to 5V (module has integrated ultra low dropout 5V to 3.3V circuit), if you buy a bare screen LCD, remember only Can connect to 3.3V.
3. About the backlight voltage: the module with the PCB backplane has access to 3.3 V and no more manual access is required. If you are buying a bare screen, the LEDA is connected to 3.0V-3.3V and the LEDKx is grounded.

Number	Module Pin	Pin Description
1	5V	Power pin
2	5V	
3	DB8	Data bus high 8-bit pin
4	DB9	
5	DB10	
6	DB11	
7	DB12	
8	DB13	
9	DB14	
10	DB15	
11	DB7	Data bus low 8-bit pin
12	DB6	
13	DB5	
14	DB4	
15	DB3	

16	<b>DB2</b>	
17	<b>DB1</b>	
18	<b>DB0</b>	
19	<b>RS</b>	LCD register / data selection pin(high level:data, low level:register)
20	<b>WR</b>	LCD write control pin
21	<b>CS</b>	LCD chip select control pin(low level active)
22	<b>RST</b>	LCD reset control pin(low level active)
23	<b>NC</b>	Undefined, reserved
24	<b>NC</b>	
25	<b>NC</b>	
26	<b>F_CS</b>	Extended application: SPI flash Chip Select Pin
27	<b>NC</b>	Undefined, reserved
28	<b>NC</b>	
29	<b>NC</b>	
30	<b>NC</b>	
31	<b>MISO</b>	SPI bus input pin (extended application)
32	<b>MOSI</b>	SPI bus output pin (extended application)
33	<b>CLK</b>	SPI bus clock pin (extended application)
34	<b>SD_CS</b>	Extended reference: SD card select pin
35	<b>GND</b>	Power ground pin
36	<b>GND</b>	

## Hardware Configuration

The LCD module hardware circuit comprises three parts: LCD display control circuit, level conversion circuit, SD card 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.

## working principle

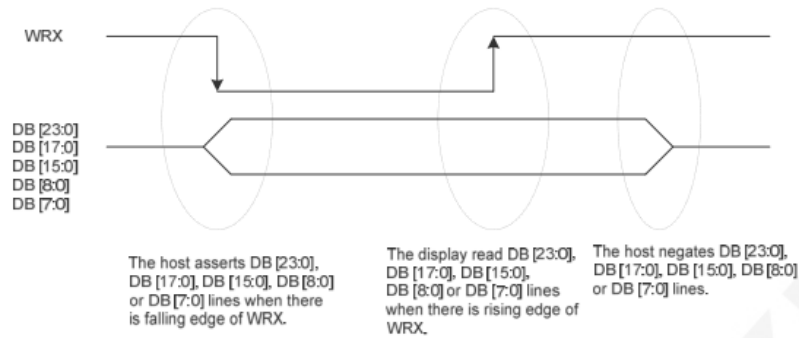
### 1. Introduction to ST7796S Controller

The ST7796S is a single-chip controller for 262 K color TFT-LCDs. It supports a maximum resolution of 320\*480 and has a GRAM of 345600 bytes. 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. ST7796S 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 ST7796S 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 ST7796S display method is performed by setting the address and then setting the color value.

#### Introduction to parallel port communication

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

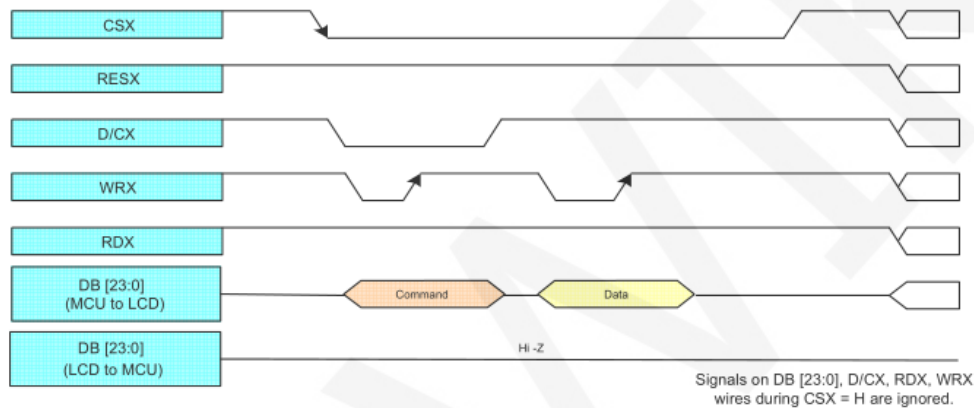


**Figure 1: DBI Type B Write Cycle**

**Note:** WRX is an unsynchronized signal that can be terminated when not being used.

When the D/CX signal is driven to low level, the input data on the interface is interpreted as command information.

The D/CX signal can also be pulled to high level when the data is RAM data or command parameter.



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

#### Arduino MEGA2560 microcontroller test program directly insert instructions

Number	Module Pin	Corresponding to MEGA2560 development board direct plug pins
1	5V	5V

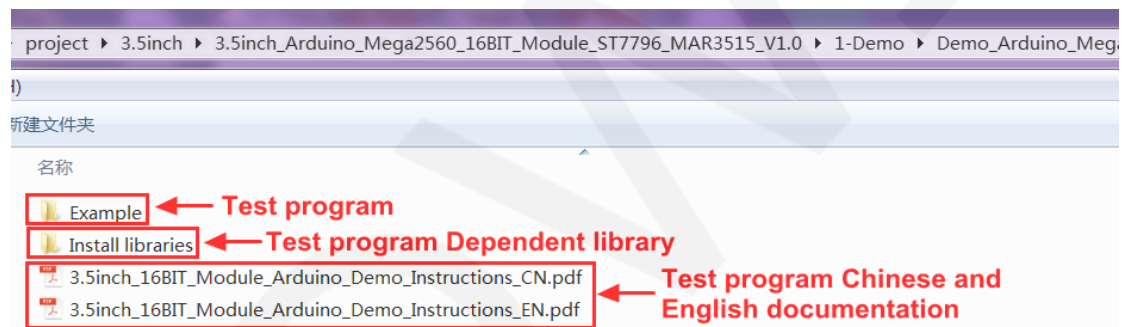


2	5V	
3	DB8	22
4	DB9	23
5	DB10	24
6	DB11	25
7	DB12	26
8	DB13	27
9	DB14	28
10	DB15	29
11	DB7	30
12	DB6	31
13	DB5	32
14	DB4	33
15	DB3	34
16	DB2	35
17	DB1	36
18	DB0	37
19	RS	38
20	WR	39
21	CS	40
22	RST	41
23	NC	not used
24	NC	
25	NC	
26	F_CS	45
27	NC	not used
28	NC	
29	NC	
30	NC	
31	MISO	50
32	MOSI	51
33	CLK	52
34	SD_CS	53

35	GND	GND
36	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](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.

<b>STC89C52RC microcontroller test program wiring instructions</b>		
<b>Number</b>	<b>Module Pin</b>	<b>Corresponding to STC89 development board wiring pin</b>
1	5V	5V
2	5V	
3	DB8	P20
4	DB9	P21
5	DB10	P22
6	DB11	P23
7	DB12	P24
8	DB13	P25
9	DB14	P26
10	DB15	P27
11	DB7	P37
12	DB6	P36
13	DB5	P35
14	DB4	P34
15	DB3	P33
16	DB2	P32
17	DB1	P31
18	DB0	P30
19	RS	P12
20	WR	P11
21	CS	P13
22	RST	P14
23	NC	No need to connect
24	NC	
25	NC	
26	F_CS	No need to connect
27	NC	No need to connect

28	NC	
29	NC	
30	NC	
31	MISO	No need to connect
32	MOSI	No need to connect
33	CLK	No need to connect
34	SD_CS	No need to connect
35	GND	GND
36	GND	

### STC12C5A60S2 microcontroller test program wiring instructions

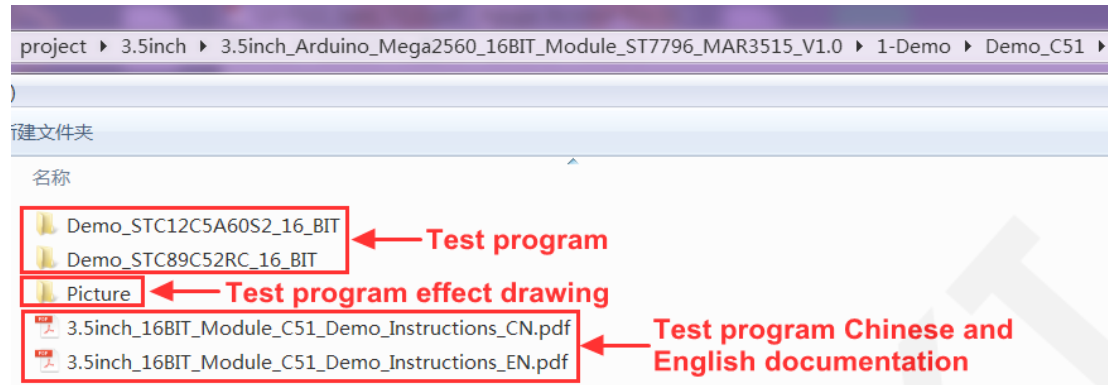
Number	Module Pin	Corresponding to STC12 development board wiring pin
1	5V	5V
2	5V	
3	DB8	P20
4	DB9	P21
5	DB10	P22
6	DB11	P23
7	DB12	P24
8	DB13	P25
9	DB14	P26
10	DB15	P27
11	DB7	P07
12	DB6	P06
13	DB5	P05
14	DB4	P04
15	DB3	P03
16	DB2	P02
17	DB1	P01
18	DB0	P00

19	RS	P12
20	WR	P11
21	CS	P13
22	RST	P33
23	NC	No need to connect
24	NC	
25	NC	
26	F_CS	No need to connect
27	NC	No need to connect
28	NC	
29	NC	
30	NC	
31	MISO	No need to connect
32	MOSI	No need to connect
33	CLK	No need to connect
34	SD_CS	No need to connect
35	GND	GND
36	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](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 Mini STM32 development board wiring pin
1	5V	5V
2	5V	
3	DB8	PB8
4	DB9	PB9
5	DB10	PB10
6	DB11	PB11
7	DB12	PB12
8	DB13	PB13
9	DB14	PB14
10	DB15	PB15

11	DB7	PB7
12	DB6	PB6
13	DB5	PB5
14	DB4	PB4
15	DB3	PB3
16	DB2	PB2
17	DB1	PB1
18	DB0	PB0
19	RS	PC8
20	WR	PC7
21	CS	PC9
22	RST	PC10
23	NC	No need to connect
24	NC	
25	NC	
26	F_CS	No need to connect
27	NC	No need to connect
28	NC	
29	NC	
30	NC	
31	MISO	No need to connect
32	MOSI	No need to connect
33	CLK	No need to connect
34	SD_CS	No need to connect
35	GND	GND
36	GND	

<b>STM32F103ZET6 microcontroller test program wiring instructions</b>		
<b>Number</b>	<b>Module Pin</b>	<b>Corresponding to Elite STM32 development board wiring pin</b>
1	5V	5V
2	5V	
3	DB8	PF8
4	DB9	PF9
5	DB10	PF10
6	DB11	PF11
7	DB12	PF12
8	DB13	PF13
9	DB14	PF14
10	DB15	PF15
11	DB7	PF7
12	DB6	PF6
13	DB5	PF5
14	DB4	PF4
15	DB3	PF3
16	DB2	PF2
17	DB1	PF1
18	DB0	PF0
19	RS	PC8
20	WR	PC7
21	CS	PC9
22	RST	PC10
23	NC	No need to connect
24	NC	
25	NC	
26	F_CS	No need to connect
27	NC	No need to connect



28	NC	
29	NC	
30	NC	
31	MISO	No need to connect
32	MOSI	No need to connect
33	CLK	No need to connect
34	SD_CS	No need to connect
35	GND	GND
36	GND	

### STM32F407ZGT6 microcontroller test program wiring instructions

Number	Module Pin	Corresponding to Explorer STM32 development board wiring pin
1	5V	5V
2	5V	
3	DB8	PG8
4	DB9	PG9
5	DB10	PG10
6	DB11	PG11
7	DB12	PG12
8	DB13	PG13
9	DB14	PG14
10	DB15	PG15
11	DB7	PG7
12	DB6	PG6
13	DB5	PG5
14	DB4	PG4
15	DB3	PG3
16	DB2	PG2
17	DB1	PG1

18	DB0	PG0
19	RS	PC8
20	WR	PC7
21	CS	PC9
22	RST	PC10
23	NC	No need to connect
24	NC	
25	NC	
26	F_CS	No need to connect
27	NC	No need to connect
28	NC	
29	NC	
30	NC	
31	MISO	No need to connect
32	MOSI	No need to connect
33	CLK	No need to connect
34	SD_CS	No need to connect
35	GND	GND
36	GND	

**STM32F429IGT6, STM32F767IGT6, STM32H743IIT6  
microcontroller test program wiring instructions**

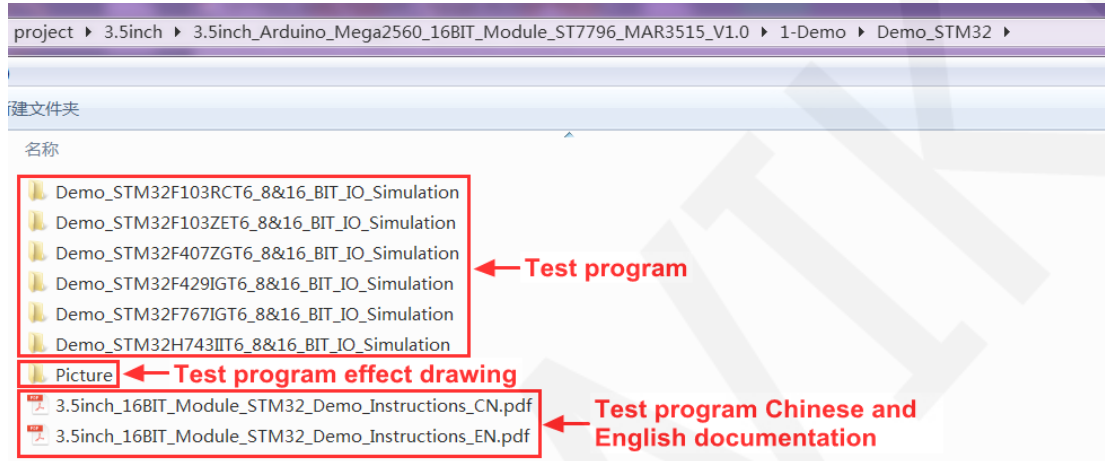
Number	Module Pin	Corresponding to Apollo STM32F4/F7 development board wiring pin(using FSMC bus )
1	5V	5V
2	5V	
3	DB8	PE8
4	DB9	PE9
5	DB10	PE10
6	DB11	PE11

7	DB12	PE12
8	DB13	PE13
9	DB14	PE14
10	DB15	PE15
11	DB7	PE7
12	DB6	PE6
13	DB5	PE5
14	DB4	PE4
15	DB3	PE3
16	DB2	PE2
17	DB1	PE1
18	DB0	PE0
19	RS	PC8
20	WR	PC7
21	CS	PC9
22	RST	PC10
23	NC	No need to connect
24	NC	
25	NC	
26	F_CS	No need to connect
27	NC	No need to connect
28	NC	
29	NC	
30	NC	
31	MISO	No need to connect
32	MOSI	No need to connect
33	CLK	No need to connect
34	SD_CS	No need to connect
35	GND	GND
36	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)



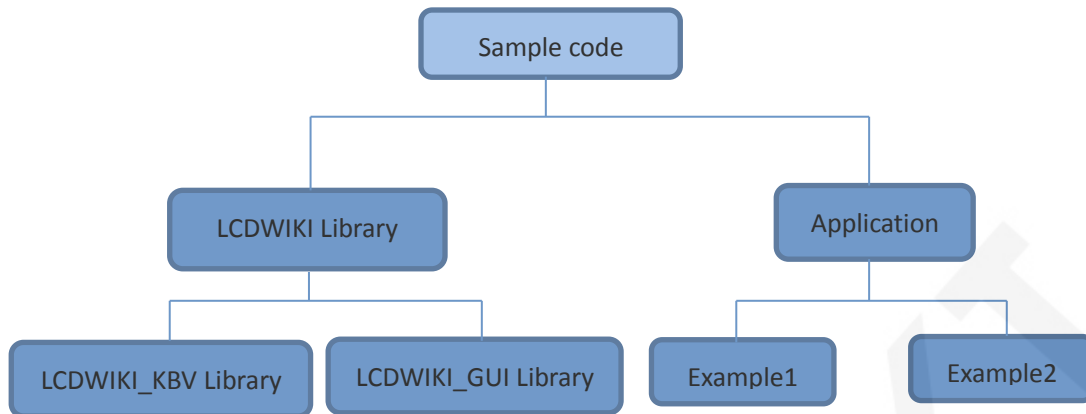
- C. Open the selected test program project, compile and download; detailed description of the STM32 test program compilation and download can be found in the following document:  
[http://www.lcdwiki.com/res/PublicFile/STM32\\_Keil\\_Use\\_Illustration\\_EN.pdf](http://www.lcdwiki.com/res/PublicFile/STM32_Keil_Use_Illustration_EN.pdf)
- 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 two parts: LCDWIKI\_KBV library and LCDWIKI\_GUI library;

The application contains several test examples, each with different test content;

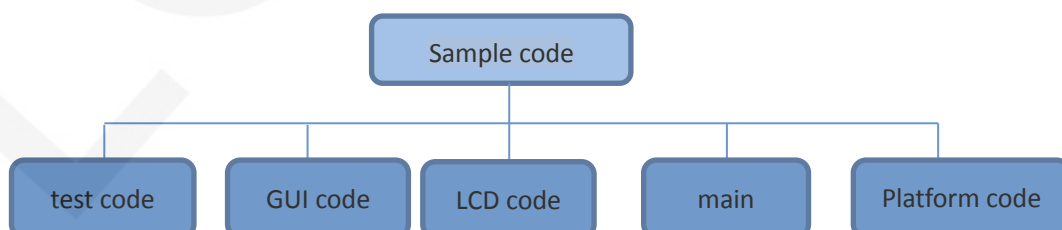
LCDWIKI\_KBV is the underlying library, which is associated with hardware. It is mainly responsible for operating registers, including hardware module initialization, data and command transmission, pixel coordinates and color settings, display mode configuration, etc;

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

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;

## 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.

### C. STM32 test program GPIO definition description

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

```
////////////////////////////////////
//-----LCD port definition-----
#define GPIO_TYPE GPIOC //GPIO group type
//#define LED 4 //Backlight control pin PC4
#define LCD_CS 9 //Chip Select pin PC9
#define LCD_RS 8 //Data/command switching pin PC8
#define LCD_RST 5 //reset pin PC10
#define LCD_WR 7 //write pin PC7
```

```
//PB0~15,As the data line
//note:If using an 8-bit mode data bus,Then the LCD
//Example:If connected to 8-bit mode, this example i
//Example:If it is 16-bit mode:DB0-DB7 are connected
#define DATAOUT(x) GPIOB->ODR=x; //data output
```

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.

### 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.

## 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](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.