3.5inch Arduino 16BIT Module MAR3513 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 ILI9486. 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

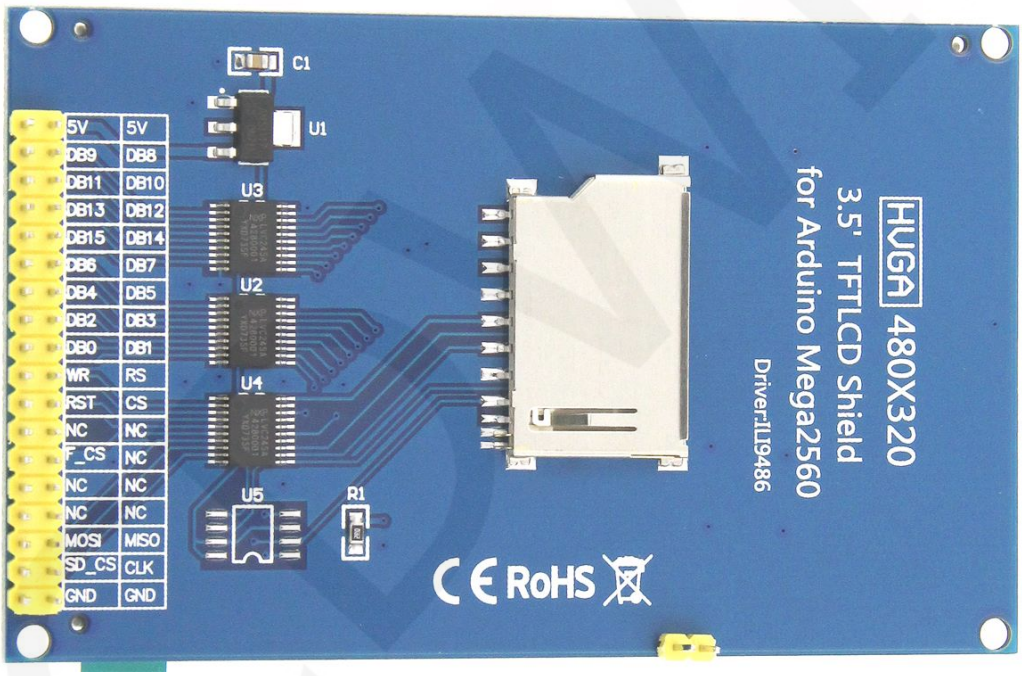
- 3.5-inch color screen, support 16BIT RGB 65K color display, display rich colors
- 320x480 HD resolution for clear display
- 16-bit parallel bus transmission for fast transfer speed
- On-board 5V/3.3V level-shifting IC compatible with 5V/3.3V operating voltage
- Support Arduino Mage2560 for direct plug-in 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	MAR3513
Screen Size	3.5(inch)
Type	TFT
Driver IC	ILI9486
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℃~60℃
Storage Temperature	-20℃~70℃
Operating Voltage	3.3V / 5V
Power Consumption	TDB
Product Weight	About 45(g)

Interface Description



Module pin Silkscreen picture

Number	Module Pin	Pin Description
1	**5V**	Power pin
2	**DB0**	Data bus low 8-bit pin
3	**DB1**	
4	**DB2**	

5	DB3	
6	DB4	
7	DB5	
8	DB6	
9	DB7	
10	DB8	Data bus high 8-bit pin
11	DB9	
12	DB10	
13	DB11	
14	DB12	
15	DB13	
16	DB14	
17	DB15	
18	RS	LCD register / data selection pin
19	WR	LCD write control pin
20	CS	LCD chip select control pin
21	RST	LCD reset control pin
22	NC	Undefined, reserved
23	F_CS	Extended application: SPI flash Chip Select Pin
24	MISO	SPI bus input pin (extended application)
25	MOSI	SPI bus output pin (extended application)
26	CLK	SPI bus clock pin (extended application)
27	SD_CS	Extended reference: SD card select pin
28	GND	Power ground pin

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

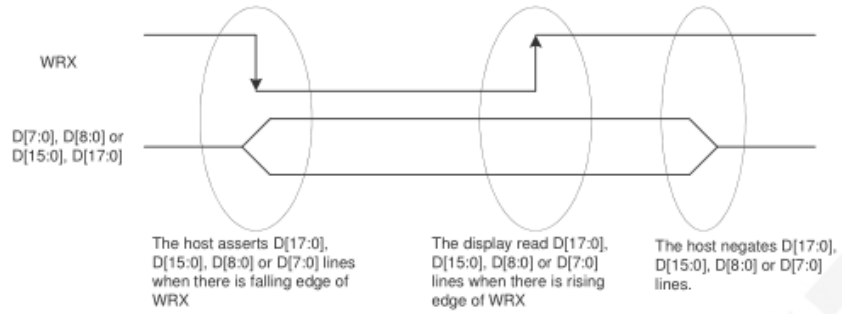
1. Introduction to ITI9486 Controller

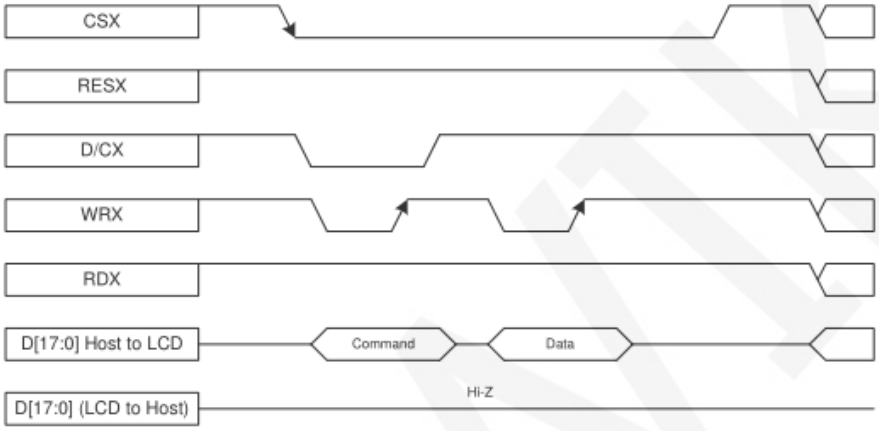
The ITI9486 controller supports a maximum resolution of 320*480 and has a 345600-byte GRAM. It also supports 8-bit, 9-bit, 16-bit, and 18-bit parallel port data buses. It also supports 3-wire and 4-wire SPI serial ports. Since the supported resolution is relatively large and the amount of data transmitted is large, the parallel port transmission is adopted, and the transmission speed is fast. ITI9486 also supports 65K, 262K 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 ITI9486 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 ITI9486 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:





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

Direct insertion instructions for Arduino MEGA2560 microcontroller test program pins		
Number	**Module Pin**	**Corresponding to MEGA2560 development board direct plug pins**
1	**5V**	5V

2	DB0	37
3	DB1	36
4	DB2	35
5	DB3	34
6	DB4	33
7	DB5	32
8	DB6	31
9	DB7	30
10	DB8	22
11	DB9	23
12	DB10	24
13	DB11	25
14	DB12	26
15	DB13	27
16	DB14	28
17	DB15	29
18	RS	38
19	WR	39
20	CS	40
21	RST	41
22	NC	No need to connect
23	F_CS	45
24	MISO	50
25	MOSI	51
26	CLK	52
27	SD_CS	53
28	GND	GND

Operating Steps：

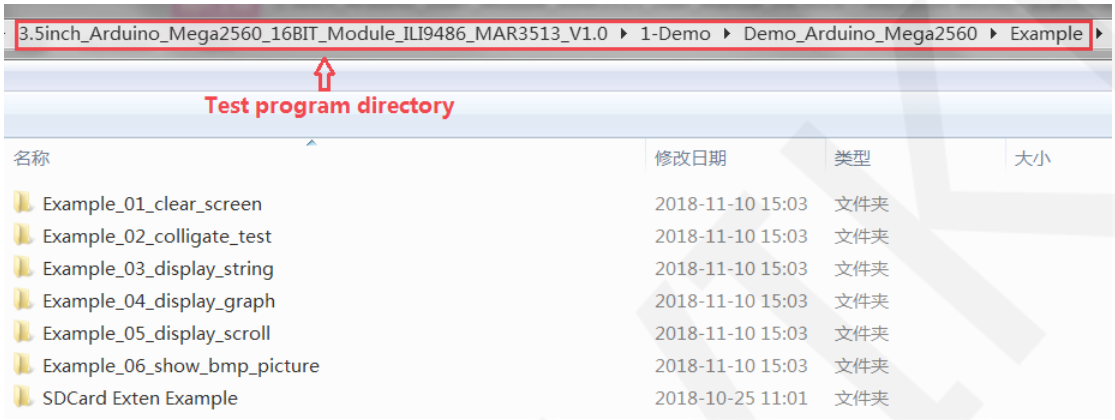
A. Insert the LCD module directly into the Arduino MCU according to the above wiring instructions, and power on;

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

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.

STC89C52RC microcontroller test program wiring instructions		
Number	**Module Pin**	**Corresponding to STC89 development board wiring pin**
1	**5V**	5V
2	**DB0**	P30
3	**DB1**	P31
4	**DB2**	P32
5	**DB3**	P33

6	**DB4**	P34
7	**DB5**	P35
8	**DB6**	P36
9	**DB7**	P37
10	**DB8**	P20
11	**DB9**	P21
12	**DB10**	P22
13	**DB11**	P23
14	**DB12**	P24
15	**DB13**	P25
16	**DB14**	P26
17	**DB15**	P27
18	**RS**	P12
19	**WR**	P11
20	**CS**	P13
21	**RST**	P14
22	**NC**	No need to connect
23	**F_CS**	No need to connect
24	**MISO**	No need to connect
25	**MOSI**	No need to connect
26	**CLK**	No need to connect
27	**SD_CS**	No need to connect
28	**GND**	GND

STC12C5A60S2 microcontroller test program wiring instructions

Number	Module Pin	Corresponding to STC12 development board wiring pin
1	**5V**	5V
2	**DB0**	P00
3	**DB1**	P01
4	**DB2**	P02

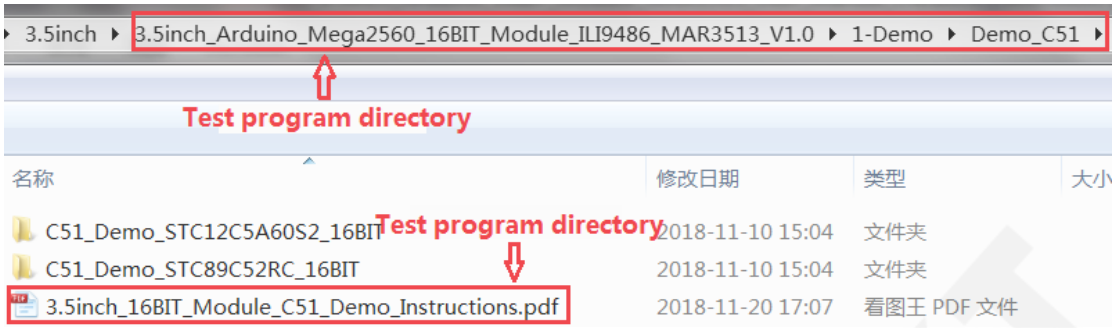
5	**DB3**	P03
6	**DB4**	P04
7	**DB5**	P05
8	**DB6**	P06
9	**DB7**	P07
10	**DB8**	P20
11	**DB9**	P21
12	**DB10**	P22
13	**DB11**	P23
14	**DB12**	P24
15	**DB13**	P25
16	**DB14**	P26
17	**DB15**	P27
18	**RS**	P12
19	**WR**	P11
20	**CS**	P13
21	**RST**	P33
22	**NC**	No need to connect
23	**F_CS**	No need to connect
24	**MISO**	No need to connect
25	**MOSI**	No need to connect
26	**CLK**	No need to connect
27	**SD_CS**	No need to connect
28	**GND**	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

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 MiniSTM32 development board wiring pin**
1	**5V**	5V
2	**DB0**	PB0
3	**DB1**	PB1
4	**DB2**	PB2
5	**DB3**	PB3
6	**DB4**	PB4
7	**DB5**	PB5
8	**DB6**	PB6
9	**DB7**	PB7
10	**DB8**	PB8
11	**DB9**	PB9
12	**DB10**	PB10
13	**DB11**	PB11

14	**DB12**	PB12
15	**DB13**	PB13
16	**DB14**	PB14
17	**DB15**	PB15
18	**RS**	PC8
19	**WR**	PC7
20	**CS**	PC9
21	**RST**	PC10
22	**NC**	No need to connect
23	**F_CS**	No need to connect
24	**MISO**	No need to connect
25	**MOSI**	No need to connect
26	**CLK**	No need to connect
27	**SD_CS**	No need to connect
28	**GND**	GND

STM32F103ZET6 microcontroller test program wiring instructions		
Number	**Module Pin**	**Corresponding to Elite STM32 development board wiring pin**
1	**5V**	5V
2	**DB0**	PF0
3	**DB1**	PF1
4	**DB2**	PF2
5	**DB3**	PF3
6	**DB4**	PF4
7	**DB5**	PF5
8	**DB6**	PF6
9	**DB7**	PF7
10	**DB8**	PF8
11	**DB9**	PF9
12	**DB10**	PF10
13	**DB11**	PF11
14	**DB12**	PF12

15	**DB13**	PF13
16	**DB14**	PF14
17	**DB15**	PF15
18	**RS**	PC8
19	**WR**	PC7
20	**CS**	PC9
21	**RST**	PC10
22	**NC**	No need to connect
23	**F_CS**	No need to connect
24	**MISO**	No need to connect
25	**MOSI**	No need to connect
26	**CLK**	No need to connect
27	**SD_CS**	No need to connect
28	**GND**	GND

STM32F407ZGT6 microcontroller test program wiring instructions		
Number	**Module Pin**	**Corresponding to Explorer STM32F4 development board wiring pin**
1	**5V**	5V
2	**DB0**	PG0
3	**DB1**	PG1
4	**DB2**	PG2
5	**DB3**	PG3
6	**DB4**	PG4
7	**DB5**	PG5
8	**DB6**	PG6
9	**DB7**	PG7
10	**DB8**	PG8
11	**DB9**	PG9
12	**DB10**	PG10
13	**DB11**	PG11

14	**DB12**	PG12
15	**DB13**	PG13
16	**DB14**	PG14
17	**DB15**	PG15
18	**RS**	PC8
19	**WR**	PC7
20	**CS**	PC9
21	**RST**	PC10
22	**NC**	No need to connect
23	**F_CS**	No need to connect
24	**MISO**	No need to connect
25	**MOSI**	No need to connect
26	**CLK**	No need to connect
27	**SD_CS**	No need to connect
28	**GND**	GND

STM32F429IGT6 microcontroller test program wiring instructions		
Number	**Module Pin**	**Corresponding to Apollo STM32F4/F7 development board wiring pin**
1	**5V**	5V
2	**DB0**	PE0
3	**DB1**	PE1
4	**DB2**	PE2
5	**DB3**	PE3
6	**DB4**	PE4
7	**DB5**	PE5
8	**DB6**	PE6
9	**DB7**	PE7
10	**DB8**	PE8
11	**DB9**	PE9
12	**DB10**	PE10
13	**DB11**	PE11

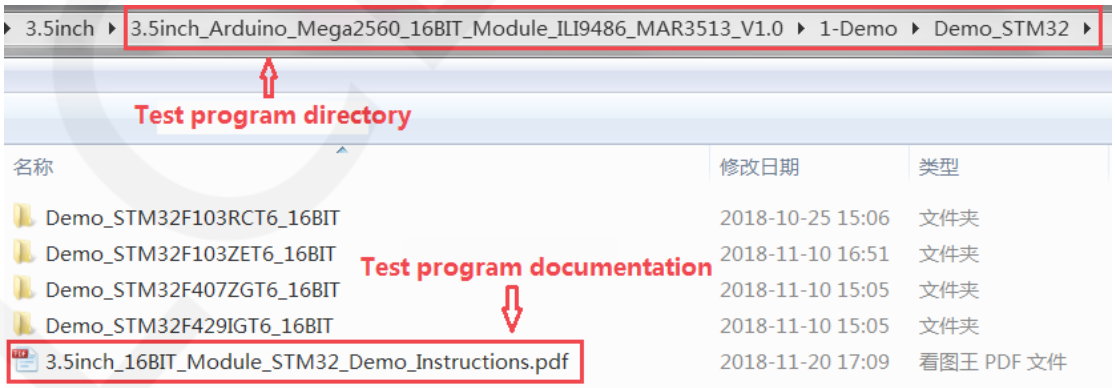
14	**DB12**	PE12
15	**DB13**	PE13
16	**DB14**	PE14
17	**DB15**	PE15
18	**RS**	PC8
19	**WR**	PC7
20	**CS**	PC9
21	**RST**	PC10
22	**NC**	No need to connect
23	**F_CS**	No need to connect
24	**MISO**	No need to connect
25	**MOSI**	No need to connect
26	**CLK**	No need to connect
27	**SD_CS**	No need to connect
28	**GND**	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

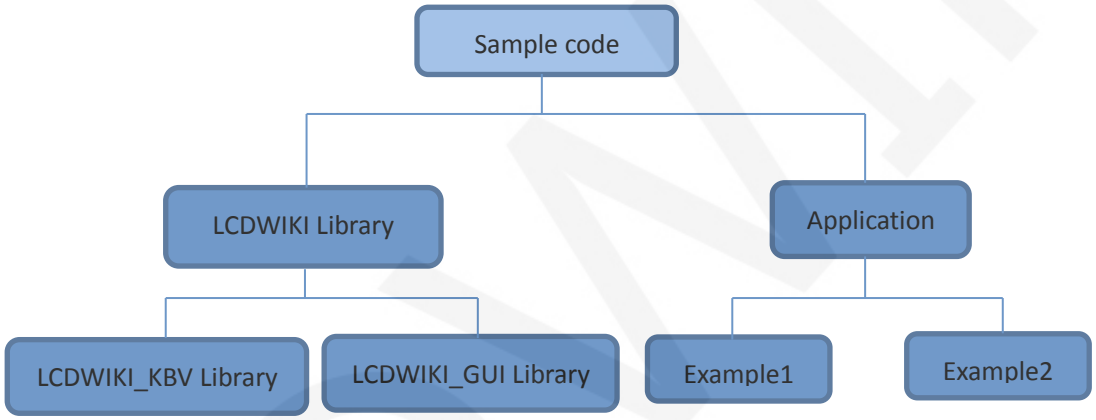
D. If the LCD module displays characters and graphics normally, the program runs successfully；

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;

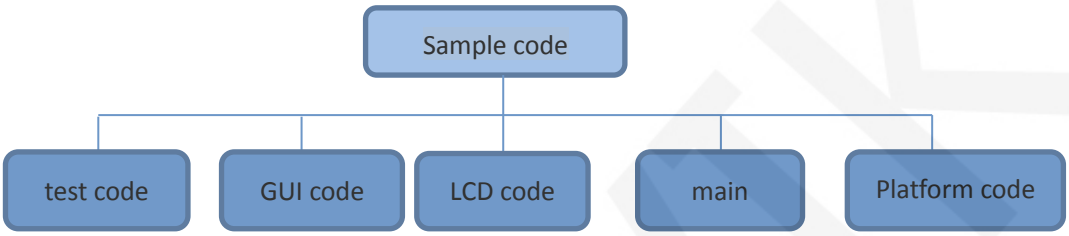
LCDWIKI_KBV is the underlying library, which is associated with hardware. It is mainly responsible for operating registers, including hardware module initialization, data and command transmission, pixel coordinates and color settings, display mode configuration, etc;

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

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.