



# 1.3inch LCD Module User Manual

## OVERVIEW

This is a general LCD display Module, IPS screen, 1.3inch diagonal, 240\*240 resolution, with embedded controller, communicating via SPI interface.

Examples are provided for testing. Examples are compatible with Raspberry Pi (bcm2835, wiringPi and python), STM32 and Arduino

## SPECIFICATION

<b>Operating Voltage</b>	: 3.3V
<b>Interface</b>	: SPI
<b>Type</b>	: TFT
<b>Control Driver</b>	: ST7789
<b>Resolution</b>	: 240(H)RGB x 240(V)
<b>Viewing Area</b>	: 23.4 (H) x 23.4 (V) mm
<b>Pixel size</b>	: 0.0975 (H) x 0.0975 (V) mm
<b>Dimension</b>	: 45 x 31(mm)

## PINOUT

PIN	Description
VCC	3.3V/5V
GND	Ground
DIN	SPI Data input
CLK	SPI Clock input
CS	Chip selection, Low active
DC	Data/Command selection
RST	Reset
BL	Backlight

# 目录

- Overview.....1
- Specification .....1
- Pinout.....2
- Hardware.....5
- Controller.....5
- Communication protocol.....5
- Demo codes .....7
- Download .....7
- Raspberry Pi.....8
- Copy to Raspberry Pi.....8
- Libraries install.....8
- Hardware connection..... 10
- Running examples..... 11
- Expected result ..... 12
- STm32 ..... 13
- Hardware connection..... 13

---

Expected result.....	13
Arduino.....	14
Hardware connection.....	14
Expected result.....	14
FAQ.....	15

## HARDWARE

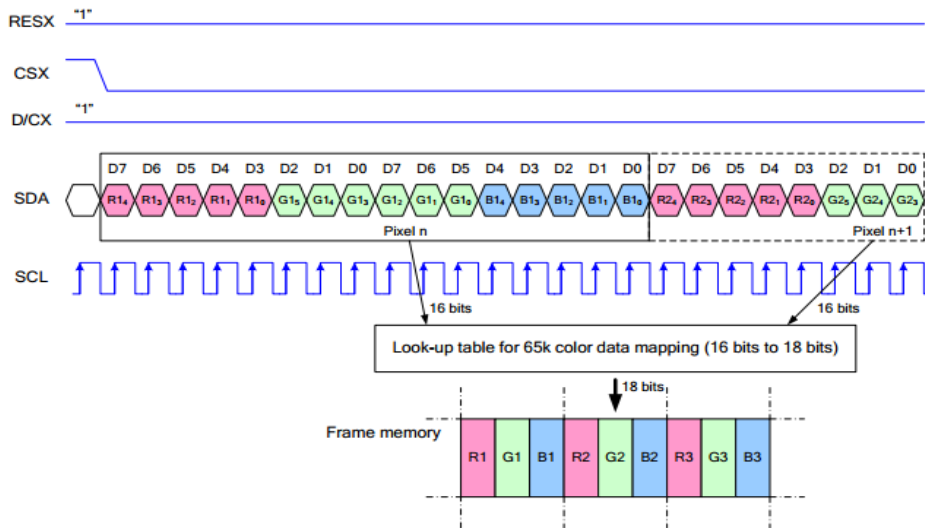
### CONTROLLER

ST7789VM is a controller for 240 x RGB x 320 LCD. Note that the resolution of this LCD module is 240(H)RGB x 240(V) indeed.

ST7789VM supports RGB444, RGB565 and RGB666 three formats. This LCD module we use RGB565.

For most of the LCD controller, there are several interfaces for choosing, this module we use SPI interface which is fast and simple.

### COMMUNICATION PROTOCOL



Note: It is not like the tradition SPI protocol, it only uses MOSI to send data from master to slave for LCD display. For details please refer to Datasheet Page 105.

RESX: Reset, should be pull-down when power on, set to 1 other time.

CSX: Slave chip select. The chip is enabled only CS is set Low

D/CX: Data/Command selection; DC=0, write command; DC=1, write data

SDA: Data transmitted. (RGB data)

SCL: SPI clock

The SPI communication protocol of the data transmission uses control bits: clock

phase (CPHA) and clock polarity (CPOL):

CPOL defines the level while synchronization clock is idle. If CPOL=0, then it is LOW.

CPHA defines at which clock's tick the data transmission starts. CPHA=0 – at the first one, otherwise at the second one

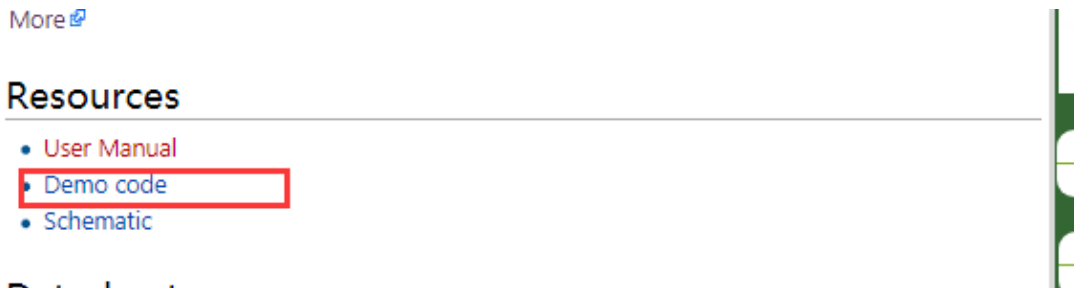
This combination of two bits provides 4 modes of SPI data transmission. The commonly used is SPI0 mode, i.e. CPHA=0 and CPOL=0.

According to the figure above, data transmitting begins at the first falling edge, 8bit data are transmitted at one clock cycle. It is SPI0. MSB.


## DEMO CODES

## DOWNLOAD

Visit Waveshare wiki and search for [1.3inch LCD Module](#). Download the demo code:



Extract and get the folders as below:

 Arduino	2018/11/26 19:18	文件夹
 RaspberryPi	2018/11/24 17:27	文件夹
 STM32	2018/11/26 19:18	文件夹

Arduino: For Arduino UNO

Raspberry Pi: Includes three examples, BCM2835, WiringPi and Python

STM32: For XNUCLEO-F103RB, which integrate STM32F103RBT6

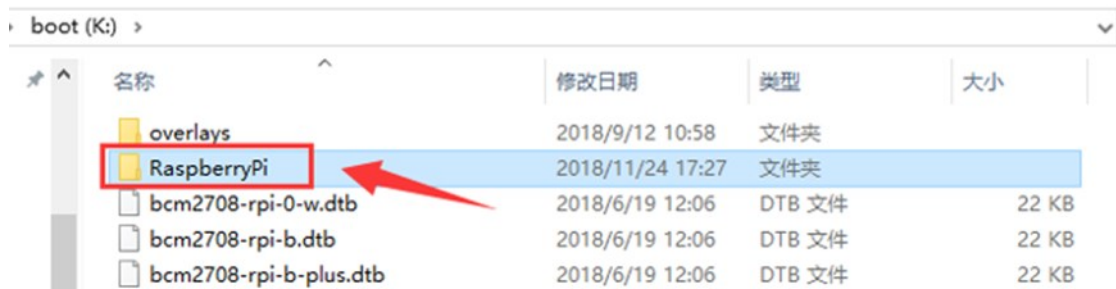
## RASPBERRY PI

### COPY TO RASPBERRY PI

1. Insert SD card which has Raspbian installed to your PC



2. Copy RaspberryPi extracted to root directory (BOOT) of SD card



3. Power on your Raspberry Pi and open Terminal, you can find that the examples is listed in boot directory

```
pi@raspberrypi:~$ ls /boot/
bcm2708-rpi-0-w.dtb  bcm2710-rpi-3-b.dtb  config.txt  fixup_x.dat  kernel.img  start_cd.elf
bcm2708-rpi-b.dtb  bcm2710-rpi-3-b-plus.dtb  COPYING.linux  FSCK0000.REC  LICENCE.broadcom  start_db.elf
bcm2708-rpi-b-plus.dtb  bcm2710-rpi-cm3.dtb  fixup_cd.dat  FSCK0001.REC  LICENSE.oracle  start_elf
bcm2708-rpi-cm.dtb  bootcode.bin  fixup.dat  issue.txt  overlays  start_x.elf
bcm2709-rpi-2-b.dtb  cmdline.txt  fixup_db.dat  kernel7.img  RaspberryPi  System Volume Information
```

4. Copy the RaspberryPi folder to /home/pi and change its execute permission.

```
sudo cp -r /boot/RaspberryPi/ ./
```

```
sudo chmod 777 -R RaspberryPi/
```

```
pi@raspberrypi:~$ sudo cp -r /boot/RaspberryPi/ ./
pi@raspberrypi:~$ ls
code  libcode  RaspberryPi  RPiLib  ubuntu  usbdisk
pi@raspberrypi:~$ sudo chmod 777 -R RaspberryPi/
pi@raspberrypi:~$ ls
code  libcode  RaspberryPi  RPiLib  ubuntu  usbdisk
```

### LIBRARIES INSTALL



To use the demo codes, you need to first install libraries

### **Install BCM2835:**

Download bcm2835 libraries from <http://www.airspayce.com/mikem/bcm2835/> , Copy it to Raspberry Pi and install it.

```
cd  
  
sudo tar zxvf bcm2835-1.xx.tar.gz  
  
cd bcm2835-1.xx  
  
sudo ./configure  
  
sudo make  
  
sudo make check  
  
sudo make install  
  
cd
```

xx is the version of library. For example, if the library you download is bcm2835-1.52, the command should be : sudo tar zxvf bcm2835-1.52.tar.gz

### **Install wiringPi:**

Open Terminal

```
cd  
  
sudo apt-get install git  
  
sudo git clone git://git.drogon.net/wiringPi  
  
cd wiringPi
```

```
sudo ./build
```

```
cd
```

### Install Python libraries:

Open Terminal

```
cd
```

```
sudo apt-get install python-pip
```

```
sudo pip install RPi.GPIO
```

```
sudo pip install spidev
```

```
sudo apt-get install python-imaging
```

```
sudo pip install numpy
```

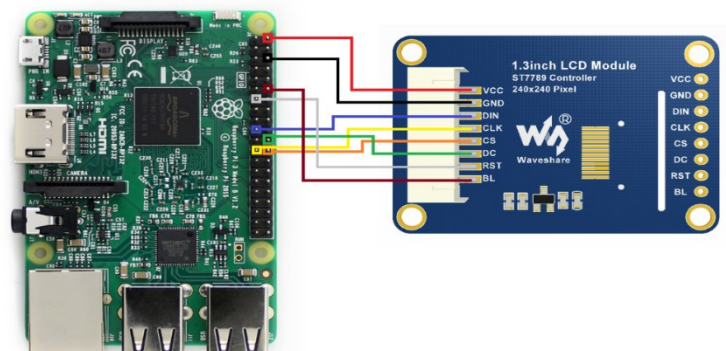
```
sudo apt-get install ttf-wqy-zenhei
```

```
cd
```

---

## HARDWARE CONNECTION

The color of cable provided may be different, please connect it according to the silk screen printing.



1.3inch LCD	Raspberry Pi (Board)	Raspberry Pi (BCM)
VCC	5V	5V
GND	GND	GND
DIN	19	MOSI
CLK	23	SCLK
CS	24	CE0
DC	22	P25
RST	13	P27
BL	12	P18

---

## RUNNING EXAMPLES

Enter the folder: `cd RaspberryPi/`

```
pi@raspberrypi:~ $ cd RaspberryPi/  
pi@raspberrypi:~/RaspberryPi $ ls  
bcm2835  python  wiringpi
```

bcm2835 example:

```
cd bcm2835  
sudo ./main
```

If you get error information that cannot find the file, please execute **sudo make** to compile codes and try again. Press Ctrl and C to stop running

wiringpi example:

```
cd wiringpi  
sudo ./main
```

If you get error information that cannot find the file, please execute **sudo make** to compile codes and try again. Press Ctrl and C to stop running

python example:

```
cd python  
sudo python main.py
```

Press Ctrl and C to stop running

---

#### EXPECTED RESULT

1. Clear screen
2. Display number and strings
3. Draw figures
4. Display 100x100 image
5. Display 240x240 image

## STM32

The development board used is XNUCLEO-F103RB, based on HAL library

---

### HARDWARE CONNECTION

1.3inch LCD	XNUCLEO-F103RB
VCC	5V
GND	GND
DIN	PA7
CLK	PA5
CS	PB6
DC	PA8
RST	PA9
BL	PB0

---

### EXPECTED RESULT

1. Clear screen
2. Display number and strings
3. Draw figures
4. Display 70x70 image

## ARDUINO

This example is compatible with Arduino UNO

---

### HARDWARE CONNECTION

1.3inch LCD	UNO
VCC	5V
GND	GND
CLK	D13
DIN	D11
CS	D10
DC	D7
RST	D8
BL	D9

---

### EXPECTED RESULT

1. Clear screen
2. Display number and strings
3. Display figures
4. Display 70x70 image

## FAQ

**1. How to control backlight?**

- You can use the function `LCD_SetBacklight()` to control the backlight

**2. Why the LCD is black when working with Raspberry Pi**

a) Check if SPI interface was enabled

b) Check if the BL pin work normally, if the pin has no output, please try to disconnect the BL control pin

**3. What does it happen if using Raspberry Pi improperly?**

If you run python or bcm2835 examples after wiringPi, the LCD may cannot work normally, please try to restart Raspberry Pi can try again.

**4. How to rotate display?**

-You can use the function `Paint_SetRotate(Rotate)` to rotate display. Rotate should be 0, 90, 180 or 270.

-Python can call `rotate(Rotate)` function for any angle.

**5. Python Image library**

- For some of the OS, you should execute command to install python-imaging

library: `sudo apt-get install python-imaging`