



0.96inch LCD Module User Manual

OVERVIEW

This is a general LCD display Module, IPS screen, 0.96inch diagonal, 160x80 HD 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

Operating Voltage	: 3.3V
Interface	: SPI
Type	: TFT
Control Driver	: ST7735S
Resolution	: 160 (V) RGB x 80 (H) mm
Viewing Area	: 21.7 (V) x 10.8 (H) mm
Pixel size	: 0.1356 (V) x 0.135 (H) mm
Dimension	: 32.5 x 26.00 (mm)

PINOUT

PIN	Description
VCC	3.3V/5V
GND	Ground
DIN	SPI Data input
CLK	SPI Clock input
CS	Chip select, 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..... 7
 - Copy to Raspberry Pi..... 7
 - Libraries install..... 8
 - Hardware connection..... 10
 - Running examples..... 10
 - Expected result..... 11
- STM32..... 12
 - Hardware connection..... 12

Expected result.....	12
Arduino.....	13
Hardware connection.....	13
Expected result.....	13
FAQ.....	14

HARDWARE

CONTROLLER

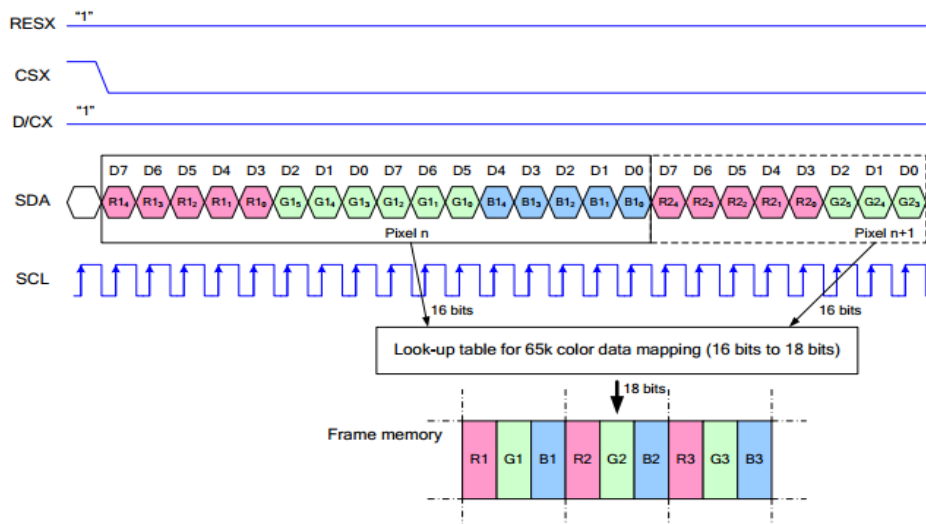
ST7735S is a controller for 162 x RGB x132 LCD. Note that the resolution of this LCD module is 160(H)RGBx80(V) indeed.

ST7735S supports RGB444, RGB565 and RGB666 three formats. This LCD module we use RGB565.

Because that the first pixel of the LCD is different with the origin point of controller, therefore, we should offset the position when initialize the module:
 Horizontal: begin from the second pixel; Vertical: begin from the 27th pixel. Make sure that the display position of LCD is same as RAM.

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. CPHL=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. CPHL=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 [0.96inch 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

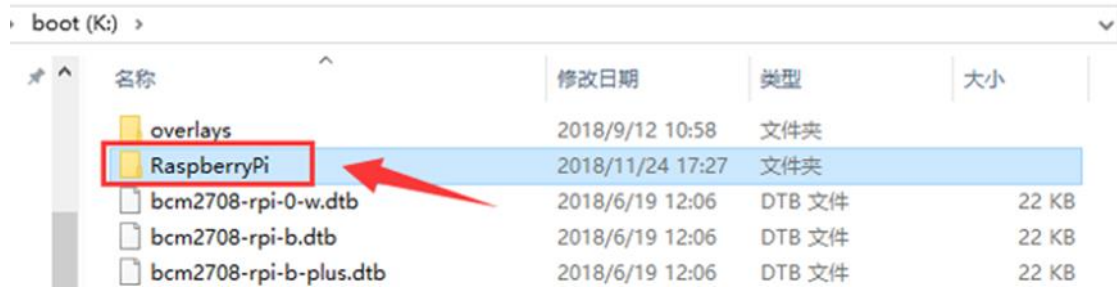
RASPBERRY PI

COPY TO RASPBERRY PI

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



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



- 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
```

- 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:

```
cd

sudo tar zxvf bcm2835-1.xx.tar.gz

cd bcm2835-1.xx

sudo ./configure
```



```
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:

```
cd

sudo apt-get install git

sudo git clone git://git.drogon.net/wiringPi

cd wiringPi

sudo ./build

cd
```

Install Python libraries:

```
cd

sudo apt-get install python-pip

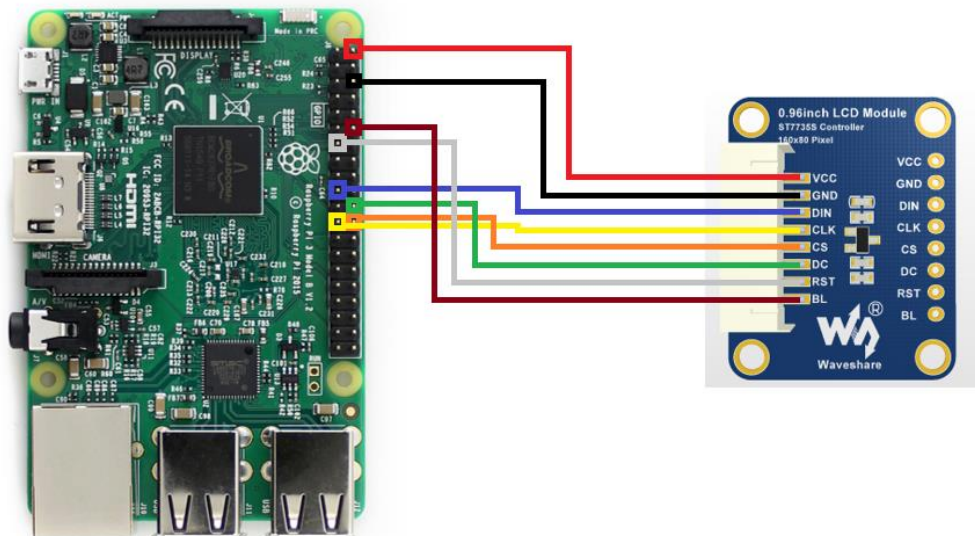
sudo pip install RPi.GPIO

sudo pip install spidev

sudo apt-get install python-imaging

cd
```

HARDWARE CONNECTION



0.96inch 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 ./motor
```

Press Ctrl and C to stop running

wiringpi example:

```
cd wiringpi  
sudo ./motor
```

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 40 x 40 image
5. Display 160x80 image

STM32

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

HARDWARE CONNECTION

0.96inch 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 40x40 image
5. Display 160x80 image

ARDUINO

This example is compatible with Arduino UNO

HARDWARE CONNECTION

0.96inch 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 40x40 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`