



1.5inch RGB OLED Module

User Manual

OVERVIEW

This is a general RGB OLED display Module, 1.5inch diagonal, 128x128 pixels, 16-bit high color (65K colors), with embedded controller, communicating via SPI interface.

SPECIFICATIONS

- Driver: SSD1351
- Interface: 4-wire SPI, 3-wire SPI
- Display color: RGB, 65K colors
- Resolution: 128x128
- Operating voltage: 3.3V/5V

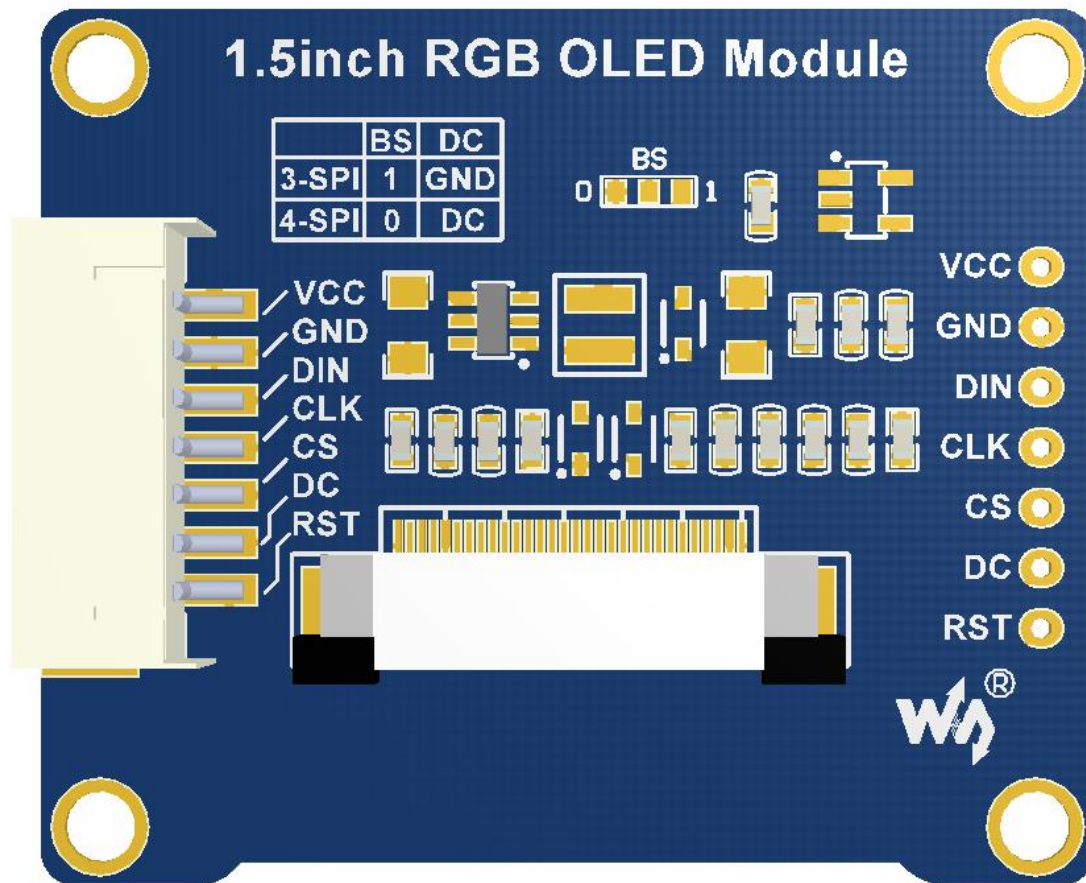
INTERFACE

Symbol	Description
VCC	Power (3.3~5V input)
GND	Ground
DIN	Data input

CLK	Clock data input
CS	Chip selection, low active
DC	4-wire SPI: Data/Command selection (high for data, low for command) 3-wire SPI: Connects to GND, keeps low
RST	Reset, low active

HARDWARE SETTING

This OLED supports two communication types: 4-wire SPI and 3-wire SPI. There is solderable resistor on the backside, you can change it for related SPI.



Factory setting is 4-wire SPI, that is BS0 set to 0.

Note: The table show the connection of pins

	BS	CS	D/C	DIN	CLK
4-wire SPI	0	Chip selection	D/C	MOSI	SCK
3-wire SPI	1	Chip selection	GND	MOSI	SCK

4-wire SPI: (Factory setting), BS set to 0 connect to GND. DIN should be connected to MOSI, and CLK should be connected to SCK.

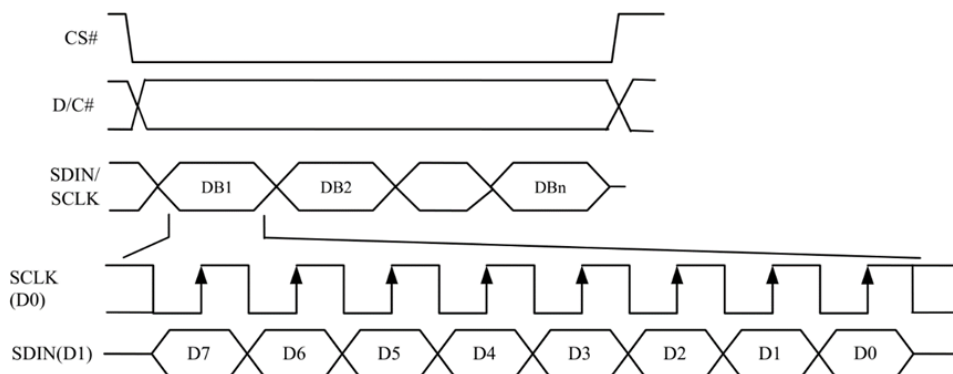
3-wire SPI: BS set to 1 connect to VCC, D/C should be connected to GND. DIN connect to MOSI and CLK connect to SCK.

WORKING PROTOCOL

SSD1351 is a Dot Matrix OLED/PLED controller for 128RGB*128 screen, embed 128*128*128 bits SRAM as display buffer. It supports 265k and 65k gray scale. Interface supports 8080, 8bit 6800, 3-wire SPI, 4-wire SPI and so on.

For reducing the size and save IO resources, 1.5inch RGB OLED uses 4-wire SPI and 3-wire SPI communication.

4WIRE-SPI



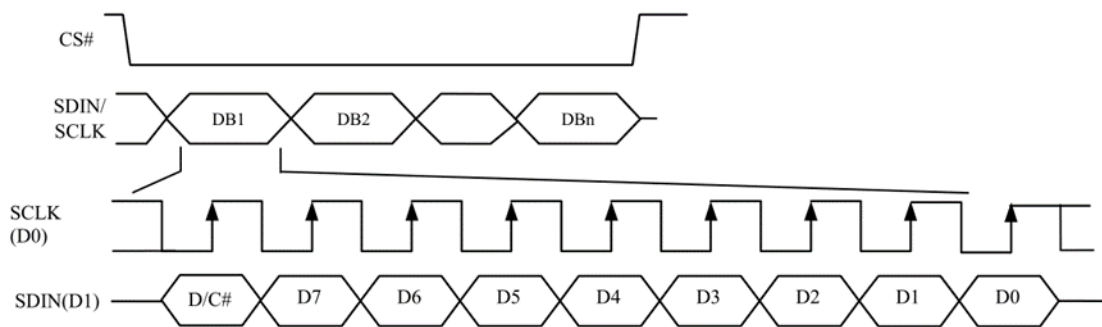
Using 4-wire SPI, you should first set the DC to 1 or 0 and then send data.

If DC=1, the data sent is stored to SRAM of SSD1351 as display data. In this mode, the length of data should be more than 1 byte.

If DC=0, the data sent is used as control command, the length of command is usually 1 byte.

For more details, please refer to SSD1352 Datasheet Figure 8-5

3-WIRE SPI



The difference between 3-wire SPI and 4-wire SPI is that 3-wire adds one bit before the transmitting byte for stand of command/data instead of DC pin.

If you use 3-wire SPI, make sure the DC pin is connected to GND. And the data transmitted are 9bit instead of 8bit.

HOW TO USE

STM32, Arduino and Raspberry Pi sample code is provided for this OLED. The sample code is used to draw shapes and display string.

STM32

CONNECTION:

The demo code is based on XNUCLEO-F103RB

1.5inch RGB OLED	XNUCLEO-F103RB
VCC	3V3/5V
GND	GND
DIN	D11(PA7)
CLK	D13(PA5)
CS	D10(PB6)
DC	D7(PA8)
RST	D8(PA9)

FILES:

Project is built for MDK-ARM v5, generated by sTM32CubeMx.

../Src:

OLED_Driver.cpp: Bottom interface of OLED, provide the function of initialization and basic display and configuration.

OLED_GFX.cpp: Display functions for OLED

ASCII_Font.h: Font library, provide two English fonts (5*8 and 8*16)for display.

ARDUINO

CONNECTION

Demo code is based on UNO PLUS

1.5inch RGB OLED	UNO PLUS
VCC	3V3/5V
GND	GND
DIN	D11
CLK	D13
CS	D10
DC	D7
RST	D8

FILES

OLED_Driver.cpp: Bottom interface of OLED , provide the function of initialization and basic display and configuration.

OLED_GFX.cpp: Display functions for OLED

ASCII_Font.h: Font library, and a 5*7 array for OLED display data.

RASPERRY PI

CONNECTION

Demo code is based on Raspberry Pi 3B

功能引脚	开发板
VCC	3V3/5V
GND	GND
DIN	MOSI
CLK	SCK
CS	CE0
DC	24(BCM)
RST	25(BCM)

SETTING

1. Enable SPI interface of Raspberry Pi

```
sudo raspi-config
```

Choose Interfaces Options->SPI->yes

2. Install libraries

For libraries installing, please refer to:

https://www.waveshare.com/wiki/Libraries_Installation_for_RPi

FILES

You need to copy the demo code (Raspberry Pi one) to your Pi. We copy to **/home/pi** of Raspberry Pi.

Enter the folder (demo code) and execute **ls**:

```
pi@raspberrypi:~/oled_test $ ls  
cambriab.ttf main.py OLED_Driver.py picture1.jpg picture2.jpg picture3.jpg
```

cambriab.ttf: Font file for string display.

mian.py: Main function file.

You can use command **sudo python main.py** to run the demo code.