

## 1. 测试平台介绍

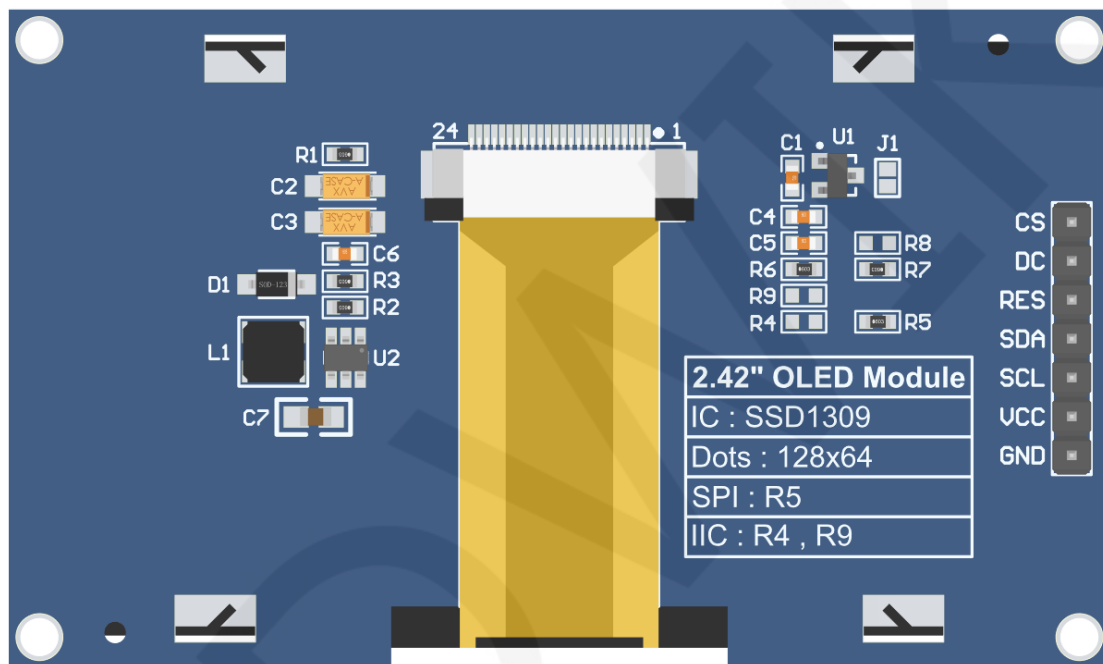
开发板: Raspberry Pi 各个型号的开发板

系统: Raspberry Pi OS

GPIO库: bcm2835、wiringpi

## 2. 引脚接线说明

显示模块使用杜邦线和Raspberry Pi连接, 具体说明如下:



模块背面引脚

### 注意:

- A、接5V单片机, 可短接J1, 使I0电压和I0高电平保持一致;
- B、R8默认不焊接, 如无需控制CS引脚, 则R8焊接0R电阻, 使CS信号保持接地;
- C、选择SPI通信方式, 则R5焊接0R电阻, R4和R9断开;
- D、选择IIC通信方式, 则R4和R9焊接0R电阻, R5断开;

wiringPi 编码	BCM 编码	功能名	物理引脚 BOARD编码		功能名	BCM 编码	wiringPi 编码
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29

Raspberry Pi GPIO map

Raspberry Pi SPI测试程序接线说明			
序号	模块引脚	对应开发板接线引脚	备注
1	GND	GND (物理引脚: 6, 9, 14, 20, 25, 30, 34, 39)	OLED屏电源地
2	VCC	5V/3.3V (物理引脚: 1, 2, 4)	OLED屏电源正
3	SCL	物理引脚: 23 BCM编码: 11 wiringPi编码: 14	SPI总线时钟信号

4	SDA	物理引脚: 19 BCM编码: 10 wiringPi编码: 12	SPI总线写数据信号
5	RES	物理引脚: 5 BCM编码: 3 wiringPi编码: 9	OLED屏复位控制信号, 低电平复位
6	DC	物理引脚: 3 BCM编码: 2 wiringPi编码: 8	OLED屏命令/数据选择控制信号 高电平: 数据, 低电平: 命令
7	CS	物理引脚: 24 BCM编码: 8 wiringPi编码: 10	OLED屏片选控制信号, 低电平有效 (如焊接R8, 则CS引脚可不接)

### Raspberry Pi IIC测试程序接线说明

序号	模块引脚	对应开发板接线引脚	备注
1	GND	GND (物理引脚: 6, 9, 14, 20, 25, 30, 34, 39)	OLED屏电源地
2	VCC	5V/3.3V (物理引脚: 1, 2, 4)	OLED屏电源正
3	SCL	物理引脚: 5 BCM编码: 3 wiringPi编码: 9	IIC总线时钟信号
4	SDA	物理引脚: 3 BCM编码: 2 wiringPi编码: 8	IIC总线数据信号
5	RES	物理引脚: 23 BCM编码: 11 wiringPi编码: 14 /3.3V	OLED屏复位控制信号, 低电平复位 (如无需控制, 可将RES引脚接高电平(3.3V))
6	DC	物理引脚: 19 BCM编码: 10 wiringPi编码: 12 /GND/3.3V	IIC总线从设备地址选择信号 接19脚时, 设为低电平: 0x78, 设为为高电平: 0x7A 低电平(接GND): 0x78, 高电平(接3.3V): 0x7A
7	CS	物理引脚: 24 BCM编码: 8 wiringPi编码: 10 /GND	OLED屏片选控制信号, 低电平有效 使用IIC通信时, 不需要控制。 接24脚时, 则24脚必须设为低电平, 也可接GND (如焊接R8, 则CS引脚可不接)

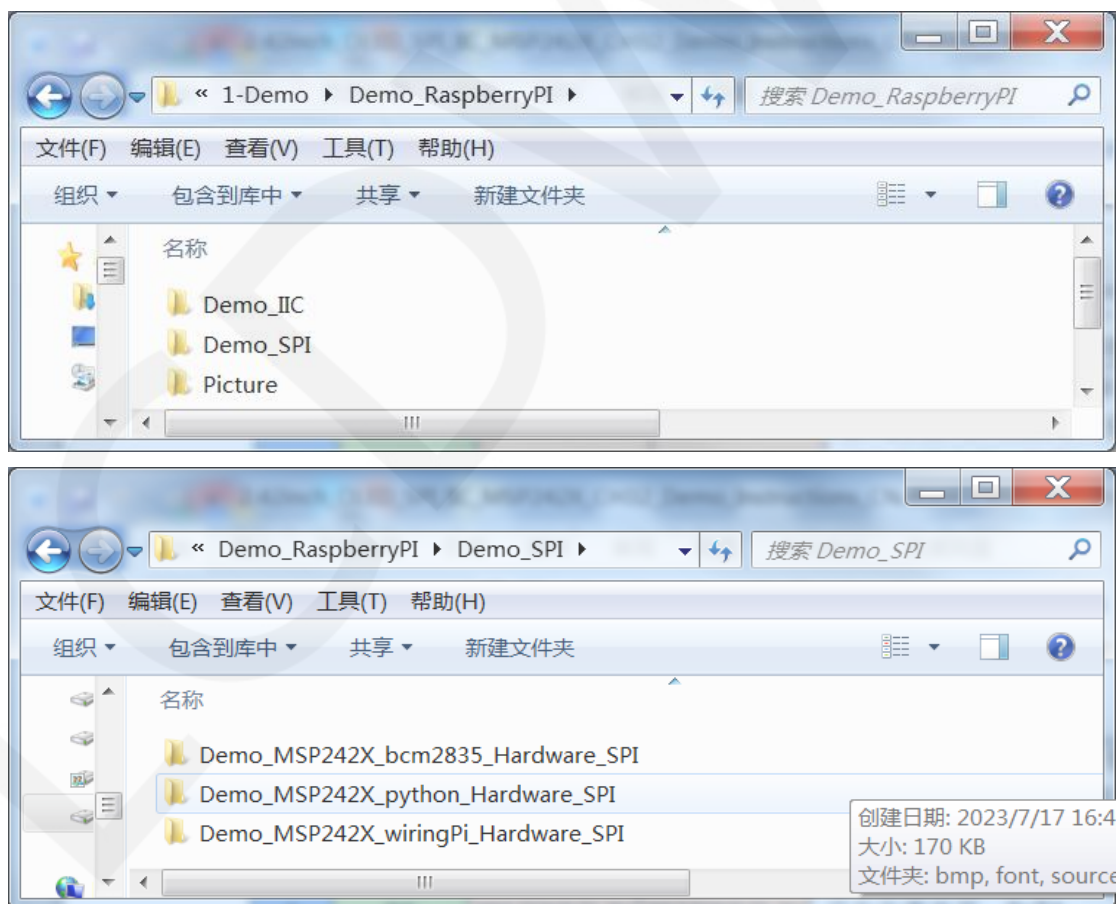
## 注意:

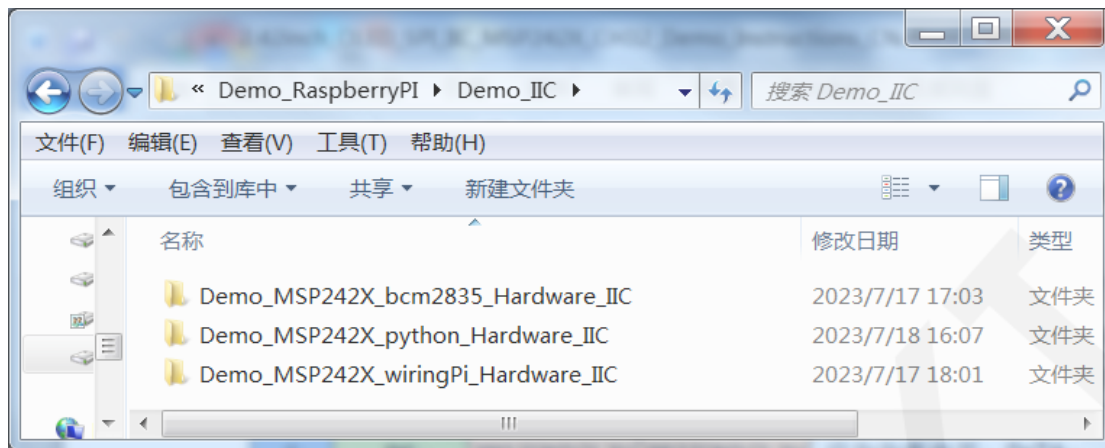
- A、物理引脚是指 RaspBerry Pi 开发板的 GPIO 引脚序号
- B、BCM 编码是指使用 BCM2835 GPIO 库时的 GPIO 引脚编码
- C、wiringPi 编码是指使用 wiringPi GPIO 库时的 GPIO 引脚编码
- D、在代码里面使用哪个 GPIO 库，引脚定义就需要使用相应的 GPIO 库编码，详情见上

图 Raspberry Pi GPIO map 表

## 3. 例程功能说明

本套测试程序程序适用于Raspberry Pi各个型号开发板，其包含bcm2835库、wiringPi GPIO库以及python测试程序，每种测试程序都有硬件SPI和硬件IIC功能测试。测试程序位于Demo\_RaspberryPI目录下，如下图所示：





#### ◇ 示例程序内容说明

- A、主界面显示测试;
- B、简单的黑白刷屏测试;
- C、矩形绘制及填充测试;
- D、圆形绘制及填充测试;
- E、三角形绘制及填充测试;
- F、英文显示测试;
- G、数字和符号显示测试;
- H、中文显示测试;
- I、BMP单色图片显示测试;
- J、菜单显示测试;

#### ◇ 示例程序显示方向和显示模式切换说明

使用bcm2835库或者wiringPi GPIO库测试程序时，在source\include\oled.h文件里找到宏定义**USE\_HORIZONTAL**和**COLOR\_STATE**，如下图所示：

```
#define USE_HORIZONTAL 0 //0-normal, 1-180degree
#define COLOR_STATE 0 //0-normal, 1-inverse
```

按如下定义修改 **USE\_HORIZONTAL** 和 **COLOR\_STATE** 宏定义即可：

```
#define USE_HORIZONTAL 0 //0° 旋转（默认值）
#define USE_HORIZONTAL 1 //180° 旋转
#define COLOR_STATE 0 //黑底，单色显示内容（默认值）
#define COLOR_STATE 1 //单色底，黑色显示内容
```

使用python测试程序时，在每个source\show\_xxx.py文件里找到**COLOR**和**DIR**定义，如下图所示：

```
# color and direction setting
COLOR = 0 #0-Normal display 1-Inverse display
DIR = 0 #0-Normal direction 1-Rotate 180 degrees
```

按如下定义修改**DIR**和**COLOR**值即可：

```
DIR = 0 #0° 旋转（默认值）
DIR = 1 #180° 旋转
COLOR = 0 #黑底，单色显示内容（默认值）
COLOR = 1 #单色底，黑色显示内容
```

#### ✧ 示例程序IIC从设备地址修改说明（只针对IIC测试程序）

使用 bcm2835 库或者 wiringPi GPIO 库测试程序时，首先在 source\include\iic.h 文件里找到宏定义 **IIC\_SLAVE\_ADDR**，如下图所示：

```
#define IIC_SLAVE_ADDR 0x3C //0x3D
```

按如下定义修改**IIC\_SLAVE\_ADDR**宏定义即可：

```
#define IIC_SLAVE_ADDR 0x3C //从设备地址为0x78（默认值）
#define IIC_SLAVE_ADDR 0x3D //从设备地址为0x7A
```

以上从设备地址在代码里都做了左移1位处理。

接下来如果使用 bcm2835 库测试程序，需在 source\src\oled.c 文件里找到

**OLED\_Init\_GPIO** 函数。如果使用 0x7A 从设备地址，则无需将

**bcm2835\_gpio\_write(10,HIGH)**这行代码注释（使其生效），如果使用 0x78 从设备地址，

则将 **bcm2835\_gpio\_write(10,HIGH)**这行代码注释起来（使其不生效），如下图所示：

```
void OLED_Init_GPIO(void)
{
    bcm2835_gpio_fsel(10, BCM2835_GPIO_FSEL_OUTP);
    bcm2835_gpio_fsel(8, BCM2835_GPIO_FSEL_OUTP);
    bcm2835_gpio_fsel(OLED_RST, BCM2835_GPIO_FSEL_OUTP);
    bcm2835_gpio_write(8,LOW);
    bcm2835_gpio_write(10,LOW);
    //slave address is 0x7A, select the follow define:
    //bcm2835_gpio_write(10,HIGH);
}
```

如果使用wiringPi GPIO库测试程序时，需在source\src\oled.c文件里找到**OLED\_Init\_GPIO**函数。如果使用0x7A从设备地址，则无需将**digitalWrite(12,HIGH)**这行代码注释（使其生效），如果使用 0x78 从设备地址，则需将 **digitalWrite(12,HIGH)**这行代码注释起来（使其不生效），如下图所示：

```
void OLED_Init_GPIO(void)
{
    pinMode(10, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(OLED_RST, OUTPUT);
    digitalWrite(10,LOW);
    digitalWrite(12,LOW);

    //slave address is 0x7A, select the folllow define:
    //digitalWrite(12,HIGH);
}
```

使用python测试程序时，首先在source\oled.py文件里找到宏定义**IIC\_SLAVE\_ADDR**，如下图所示：

```
IIC_SLAVE_ADDR = 0x3C #0x3D
```

按如下定义修改**IIC\_SLAVE\_ADDR**值即可：

```
IIC_SLAVE_ADDR = 0x3C #从设备地址为0x78（默认值）
```

```
IIC_SLAVE_ADDR = 0x3D #从设备地址为0x7A
```

以上从设备地址在代码里都做了左移1位处理。

接下来在source\oled.py文件里找到**\_\_init\_\_**函数，如果使用0x7A从设备地址，则无需将**GPIO.output(10,GPIO.HIGH)**这行代码注释（使其生效），如果使用0x78从设备地址，则需将**GPIO.output(10,GPIO.HIGH)**这行代码注释起来（使其不生效），如下图所示：

```
def __init__(self, res, smbus):
    # set oled display parameter
    self.width = WIDTH
    self.height = HEIGHT
    self.pagesize = 8
    self.ylevel = 0xB0
    self.xlevel1 = 0x00
    self.xlevelh = 0x10
    self.oledbuffer = [0]*(self.width*self.pagesize)
    self.oledres = res
    GPIO.setmode(GPIO.BCM)
    GPIO.setwarnings(False)
    GPIO.setup(8,GPIO.OUT)
    GPIO.setup(10,GPIO.OUT)
    GPIO.setup(self.oledres,GPIO.OUT)
    GPIO.output(8,GPIO.LOW)
    GPIO.output(10,GPIO.LOW)
    #slave address is 0x7A, select the follow define:
    #GPIO.output(10,GPIO.HIGH)
    # Initialize iic
    self.oledsmbus = smbus
```

## 4. 例程使用说明

### ✧ 搭建开发环境

首先得去官网下载Raspberry Pi系统镜像文件，然后使用microSD卡烧录镜像文件，具体的烧录方法，请自行网上查阅。

官网网址：<https://www.raspberrypi.org/downloads/raspbian/>

### ✧ 开启Raspberry Pi OS内核SPI和IIC内核驱动

系统镜像文件烧录好后，将microSD卡插入树莓派，然后将显示模块用杜邦线按照引脚定义接到Raspberry Pi上，接下来给Raspberry Pi插上网线，最后给Raspberry Pi上电。打开终端软件（例如putty），使用SSH方式登录Raspberry Pi（确保Raspberry Pi和PC在同一网段）。在终端软件输入如下命令：

```
sudo raspi-config
```

如果需要开启SPI内核驱动，在弹出的界面里选择 **Interfacing Options->SPI->YES**

如果需要开启IIC内核驱动，在弹出的界面里选择 **Interfacing Options->I2C->YES**

选择完成后，连续按**Esc**键保存并退出图形界面。



## ✧ 安装GPIO软件库

### A、安装bcm2835库

软件库下载方法有如下3种：

- 1) 如果Raspberry Pi无法连接互联网，则可通过PC机下载，然后使用SD卡或者FTP工具（如FileZilla）将其拷贝到Raspberry Pi OS里。

下载网址：<http://www.airspayce.com/mikem/bcm2835/>

进入网址，点击下图所示链接就可以下载了。

#### C library for Broadcom BCM 2835 as used in Raspberry Pi

This is a C library for Raspberry Pi (RPI). It provides access to GPIO and other IO functions on the Broadcom BCM 2835 chip, as used in the RaspberryPi, allowing access to devices.

It provides functions for reading digital inputs and setting digital outputs, using SPI and I2C, and for accessing the system timers. Pin event detection is supported by poll.

Works on all versions up to and including RPI 4. Works with all versions of Debian up to and including Debian Buster 10. Reported to be working on Bullseye (Raspbian v

It is C++ compatible, and installs as a header file and non-shared library on any Linux-based distro (but clearly is no use except on Raspberry Pi or another board with BC

The version of the package that this documentation refers to can be downloaded from <http://www.airspayce.com/mikem/bcm2835/bcm2835-1.73.tar.gz> You can find the

Several example programs are provided.

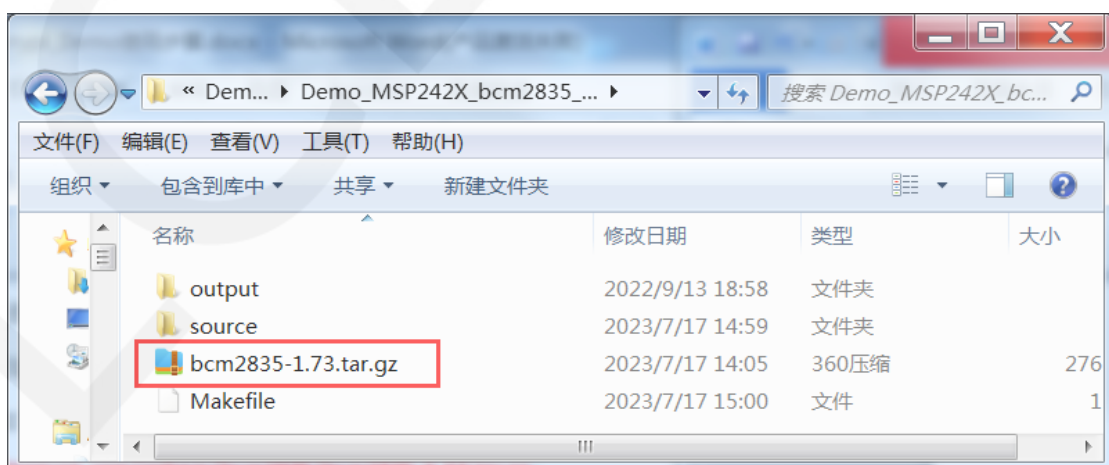
- 2) 如果Raspberry Pi可以连接互联网，则可通过终端软件在Raspberry Pi OS里输入如下命令下载：

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-X.XX.tar.gz
```

其中X.XX为软件库版本号，如上图所示，版本号为1.73，那么可以通过如下命令下载1.73版本：

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.73.tar.gz
```

- 3) 直接使用示例程序目录下的bcm2835库（如下图所示），通过SD卡或者FTP工具（如FileZilla）将其拷贝到Raspberry Pi OS里。



bcm2835库下载成功或者拷贝成功后，在终端软件输入如下命令进行解压、编译和安装：

```
tar -zxvf bcm2835-X.XX.tar.gz  
cd bcm2835-X.XX  
./configure  
sudo make  
sudo make check  
sudo make install
```

其中 **X.XX** 为 bcm2835 库版本号，需要根据实际情况填写，例如 1.73

## B、安装wiringPi GPIO库

软件库下载方法有如下4种：

- 1) 如果Raspberry Pi无法连接互联网，则可通过PC机下载，然后使用SD卡或者FTP工具（如FileZilla）将其拷贝到Raspberry Pi OS里。

下载网址：<https://project-downloads.drogon.net/wiringpi-latest.deb>

在浏览器里输入网址或者点击网址就可以下载了。

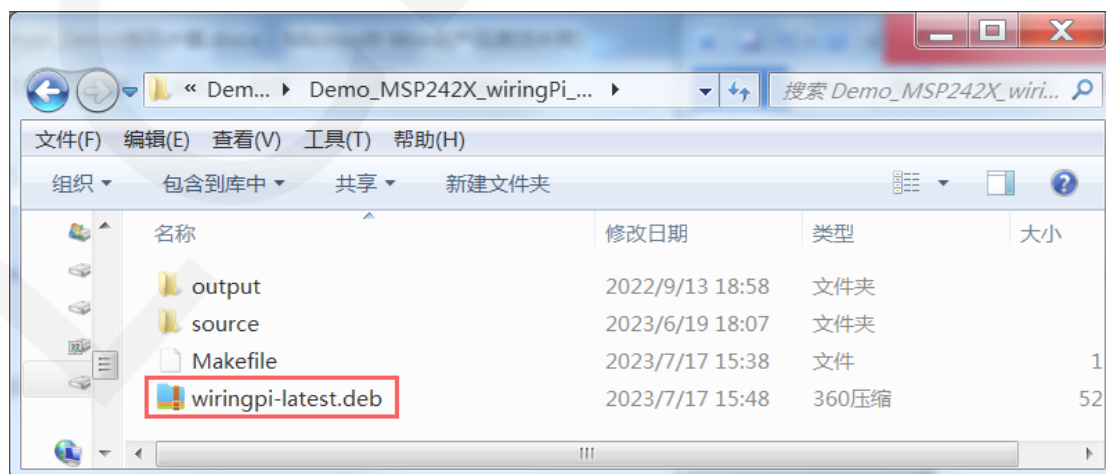
- 2) 如果Raspberry Pi可以连接互联网，则可通过终端软件在Raspberry Pi OS里输入如下命令下载：

```
wget https://project-downloads.drogon.net/wiringpi-latest.deb
```

- 3) 如果Raspberry Pi可以连接互联网，也可通过终端软件在Raspberry Pi OS里输入如下命令从github下载：

```
git clone https://github.com/WiringPi/WiringPi
```

- 4) 直接使用示例程序目录下的wiringPi GPIO库（如下图所示），通过SD卡或者FTP工具（如FileZilla）将其拷贝到Raspberry Pi OS里。



如果软件库不是从GitHub下载或拷贝，则在终端软件输入如下命令进行编译和安装：

```
sudo dpkg -i -B wiringpi-latest.deb
```

如果软件库是从Github下载，则在终端软件输入如下命令进行编译和安装：

```
cd WiringPi
```

```
./build
```

wiringPi GPIO 库安装完成后，可以在终端软件输入如下命令进行检查是否安装成功

```
gpio -v
```

```
gpio readall
```

如下图所示，红色框内显示 wiringPi 库的版本号和 GPIO 编码号，如果没有出现这些内容，那说明安装没有成功。

```
pi@raspberrypi:~ $ gpio -v
gpio version: 2.52
Copyright (c) 2012-2018 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
Type: Pi 4B, Revision: 04, Memory: 8192MB, Maker: Sony
* Device tree is enabled.
*--> Raspberry Pi 4 Model B Rev 1.4
* This Raspberry Pi supports user-level GPIO access.
pi@raspberrypi:~ $ gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|     |     | 3.3v    |      |   | 1  || 2  |     |   5v    |     |     | |
|  2  |  8  | SDA.1   | OUT  | 1 | 3  || 4  |     |   5v    |     |     |
|  3  |  9  | SCL.1   | OUT  | 1 | 5  || 6  |     |   0v    |     |     |
|  4  |  7  | GPIO. 7 | IN   | 1 | 7  || 8  | 1 | IN  | TxD    | 15  | 14  |
|     |     | 0v      |      |   | 9  || 10 | 1 | IN  | RxD    | 16  | 15  |
| 17  |  0  | GPIO. 0 | IN   | 0 | 11 || 12 | 1 | OUT | GPIO. 1 | 1   | 18  |
| 27  |  2  | GPIO. 2 | IN   | 0 | 13 || 14 |   |   0v    |     |     |
| 22  |  3  | GPIO. 3 | IN   | 0 | 15 || 16 | 0 | IN  | GPIO. 4 | 4   | 23  |
|     |     | 3.3v    |      |   | 17 || 18 | 0 | IN  | GPIO. 5 | 5   | 24  |
| 10  | 12  | MOSI    | ALT0 | 0 | 19 || 20 |   |   0v    |     |     |
|  9  | 13  | MISO    | ALT0 | 0 | 21 || 22 | 0 | IN  | GPIO. 6 | 6   | 25  |
| 11  | 14  | SCLK    | ALT0 | 0 | 23 || 24 | 1 | OUT | CE0    | 10  | 8   |
|     |     | 0v      |      |   | 25 || 26 | 1 | ALT0| CE1    | 11  | 7   |
|  0  | 30  | SDA.0   | IN   | 1 | 27 || 28 | 1 | IN  | SCL.0  | 31  | 1   |
|  5  | 21  | GPIO.21 | IN   | 1 | 29 || 30 |   |   0v    |     |     |
|  6  | 22  | GPIO.22 | IN   | 1 | 31 || 32 | 0 | IN  | GPIO.26 | 26  | 12  |
| 13  | 23  | GPIO.23 | IN   | 0 | 33 || 34 |   |   0v    |     |     |
| 19  | 24  | GPIO.24 | IN   | 0 | 35 || 36 | 0 | IN  | GPIO.27 | 27  | 16  |
| 26  | 25  | GPIO.25 | IN   | 0 | 37 || 38 | 0 | IN  | GPIO.28 | 28  | 20  |
|     |     | 0v      |      |   | 39 || 40 | 0 | IN  | GPIO.29 | 29  | 21  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

## C、安装python库

目前最新的Raspberry Pi OS已经用python3的库兼容python2了，所以只需要安装python3的库。在安装之前先在终端软件执行如下命令检查系统是否已经安装所需要的python3的库，如果已经全部安装，那就不需再安装。

```
dpkg -l | grep -e python3-pip -e python3-pil -e python3-numpy -e spidev
```

如下图所示，表示已经全部安装，不需再安装。

```
pi@raspberrypi:~$ dpkg -l | grep -e python3-pip -e python3-pil -e python3-numpy -e spidev
ii python3-numpy          1:1.19.5-1          armhf      Fast
ii python3-pil:armhf     8.1.2+dfsg-0.3+deb11u1  armhf      Pytho
ii python3-pip           20.3.4-4+rpt1+deb11u1  all        Pytho
ii python3-spidev       20200602~200721-1    armhf      Bind
```

如果没有安装，则在终端软件执行如下命令安装：

```
sudo apt-get update
```

```
sudo apt-get install python3-pip
```

```
sudo apt-get install python3-pil
```

```
sudo apt-get install python3-numpy
```

```
sudo pip3 install RPi.GPIO
```

```
sudo pip3 install spidev
```

如果运行的 Raspberry Pi OS 使用的是 python2，则在终端软件执行如下命令安装：

```
sudo apt-get update
```

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

```
sudo apt-get install python-pil
```

```
sudo apt-get install python-numpy
```

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

```
sudo pip install spidev
```

## ✧ 编译并运行程序

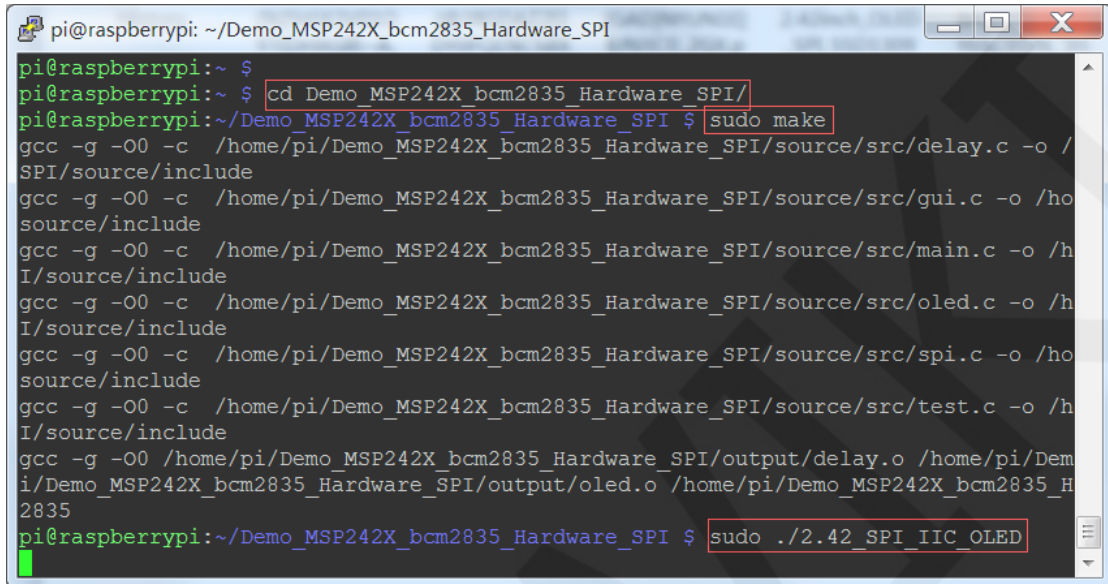
### A、编译并运行 bcm2835 库测试程序

如果使用 SPI 测试程序，则通过 SD 卡或者 FTP 工具（如 FileZilla）将资料包里 **Demo\_RaspberryPI/Demo\_SPI** 目录下的 **Demo\_MSP242X\_bcm2835\_Hardware\_SPI** 文件夹拷贝到 Raspberry Pi OS 里，然后在终端软件执行如下命令编译并运行程序：

```
cd Demo_MSP242X_bcm2835_Hardware_SPI
```

```
sudo make
```

```
sudo ./2.42_SPI_IIC_OLED
```



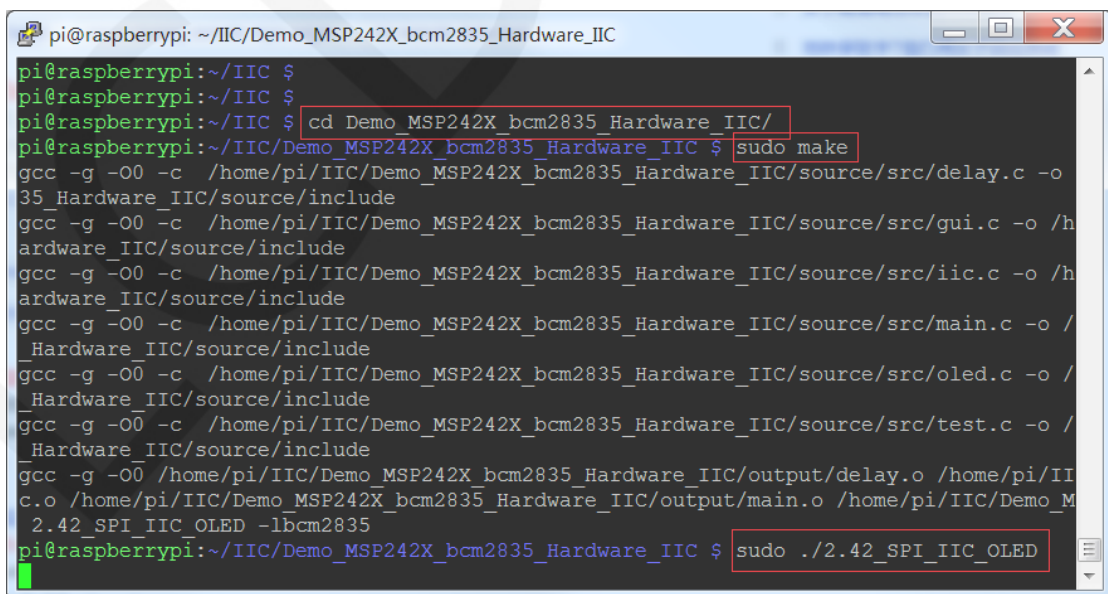
```
pi@raspberrypi: ~/Demo_MSP242X_bcm2835_Hardware_SPI
pi@raspberrypi:~ $
pi@raspberrypi:~ $ cd Demo_MSP242X_bcm2835_Hardware_SPI/
pi@raspberrypi:~/Demo_MSP242X_bcm2835_Hardware_SPI $ sudo make
gcc -g -O0 -c /home/pi/Demo_MSP242X_bcm2835_Hardware_SPI/source/src/delay.c -o /
SPI/source/include
gcc -g -O0 -c /home/pi/Demo_MSP242X_bcm2835_Hardware_SPI/source/src/gui.c -o /h
source/include
gcc -g -O0 -c /home/pi/Demo_MSP242X_bcm2835_Hardware_SPI/source/src/main.c -o /h
I/source/include
gcc -g -O0 -c /home/pi/Demo_MSP242X_bcm2835_Hardware_SPI/source/src/oled.c -o /h
I/source/include
gcc -g -O0 -c /home/pi/Demo_MSP242X_bcm2835_Hardware_SPI/source/src/spi.c -o /h
source/include
gcc -g -O0 -c /home/pi/Demo_MSP242X_bcm2835_Hardware_SPI/source/src/test.c -o /h
I/source/include
gcc -g -O0 /home/pi/Demo_MSP242X_bcm2835_Hardware_SPI/output/delay.o /home/pi/Dem
i/Demo_MSP242X_bcm2835_Hardware_SPI/output/oled.o /home/pi/Demo_MSP242X_bcm2835_H
2835
pi@raspberrypi:~/Demo_MSP242X_bcm2835_Hardware_SPI $ sudo ./2.42_SPI_IIC_OLED
```

如果使用 IIC 测试程序，则通过 SD 卡或者 FTP 工具（如 FileZilla）将资料包里 **Demo\_RaspberryPi\Demo\_IIC** 目录下的 **Demo\_MSP242X\_bcm2835\_Hardware\_IIC** 文件夹拷贝到 Raspberry Pi OS 里，然后在终端软件执行如下命令编译并运行程序：

```
cd Demo_MSP242X_bcm2835_Hardware_IIC
```

```
sudo make
```

```
sudo ./2.42_SPI_IIC_OLED
```



```
pi@raspberrypi: ~/IIC/Demo_MSP242X_bcm2835_Hardware_IIC
pi@raspberrypi:~/IIC $
pi@raspberrypi:~/IIC $ cd Demo_MSP242X_bcm2835_Hardware_IIC/
pi@raspberrypi:~/IIC/Demo_MSP242X_bcm2835_Hardware_IIC $ sudo make
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_bcm2835_Hardware_IIC/source/src/delay.c -o
35_Hardware_IIC/source/include
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_bcm2835_Hardware_IIC/source/src/gui.c -o /h
ardware_IIC/source/include
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_bcm2835_Hardware_IIC/source/src/iic.c -o /h
ardware_IIC/source/include
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_bcm2835_Hardware_IIC/source/src/main.c -o /
_Hardware_IIC/source/include
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_bcm2835_Hardware_IIC/source/src/oled.c -o /
_Hardware_IIC/source/include
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_bcm2835_Hardware_IIC/source/src/test.c -o /
_Hardware_IIC/source/include
gcc -g -O0 /home/pi/IIC/Demo_MSP242X_bcm2835_Hardware_IIC/output/delay.o /home/pi/II
c.o /home/pi/IIC/Demo_MSP242X_bcm2835_Hardware_IIC/output/main.o /home/pi/IIC/Demo_M
2.42_SPI_IIC_OLED -lbcm2835
pi@raspberrypi:~/IIC/Demo_MSP242X_bcm2835_Hardware_IIC $ sudo ./2.42_SPI_IIC_OLED
```

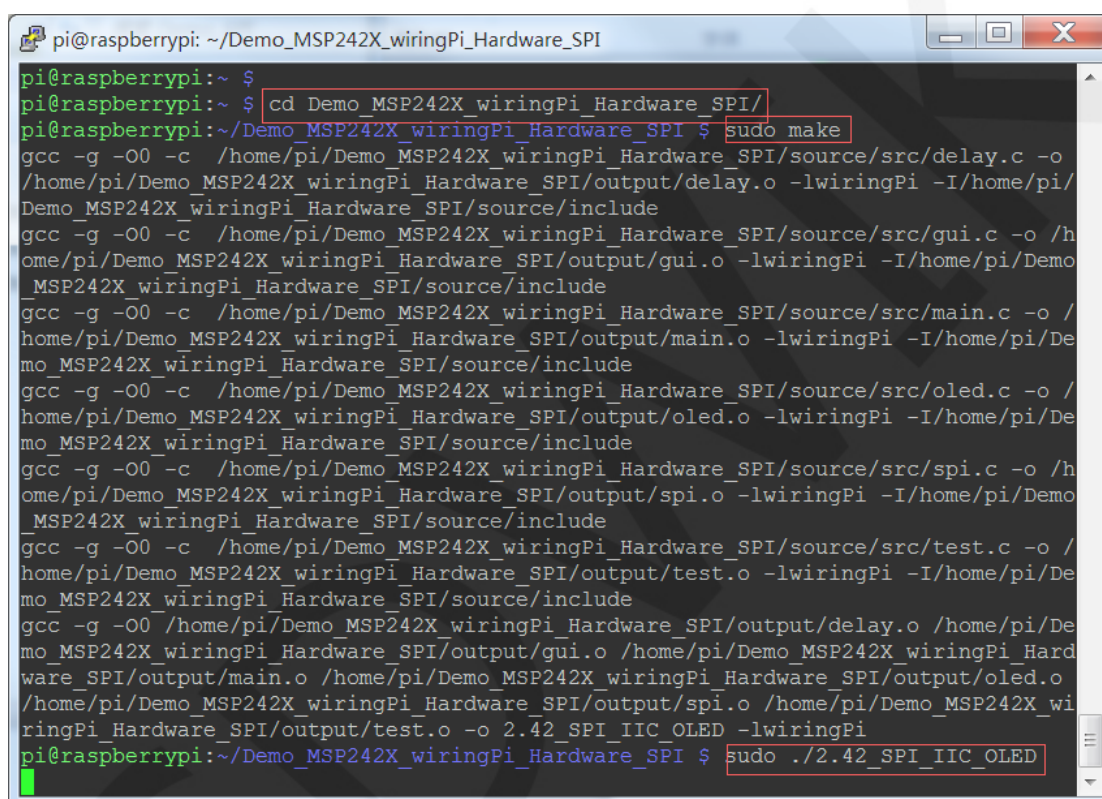
## B、编译并运行 wiringPi GPIO 库测试程序

如果使用 SPI 测试程序，则通过 SD 卡或者 FTP 工具（如 FileZilla）将资料包里 **Demo\_RaspberryPi\Demo\_SPI** 目录下的 **Demo\_MSP242X\_wiringPi\_Hardware\_SPI** 文件夹拷贝到 Raspberry Pi OS 里，然后在终端软件执行如下命令编译并运行程序：

```
cd Demo_MSP242X_wiringPi_Hardware_SPI
```

```
sudo make
```

```
sudo ./2.42_SPI_IIC_OLED
```



```
pi@raspberrypi: ~/Demo_MSP242X_wiringPi_Hardware_SPI
pi@raspberrypi:~$ cd Demo_MSP242X_wiringPi_Hardware_SPI/
pi@raspberrypi:~/Demo_MSP242X_wiringPi_Hardware_SPI$ sudo make
gcc -g -O0 -c /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/src/delay.c -o /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/delay.o -lwiringPi -I/home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/include
gcc -g -O0 -c /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/src/gui.c -o /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/gui.o -lwiringPi -I/home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/include
gcc -g -O0 -c /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/src/main.c -o /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/main.o -lwiringPi -I/home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/include
gcc -g -O0 -c /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/src/oled.c -o /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/oled.o -lwiringPi -I/home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/include
gcc -g -O0 -c /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/src/spi.c -o /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/spi.o -lwiringPi -I/home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/include
gcc -g -O0 -c /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/src/test.c -o /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/test.o -lwiringPi -I/home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/source/include
gcc -g -O0 /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/delay.o /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/gui.o /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/main.o /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/oled.o /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/spi.o /home/pi/Demo_MSP242X_wiringPi_Hardware_SPI/output/test.o -o 2.42_SPI_IIC_OLED -lwiringPi
pi@raspberrypi:~/Demo_MSP242X_wiringPi_Hardware_SPI$ sudo ./2.42_SPI_IIC_OLED
```

如果使用 IIC 测试程序，则通过 SD 卡或者 FTP 工具（如 FileZilla）将资料包里 **Demo\_RaspberryPi\Demo\_IIC** 目录下的 **Demo\_MSP242X\_wiringPi\_Hardware\_IIC** 文件夹拷贝到 Raspberry Pi OS 里，然后在终端软件执行如下命令编译并运行程序：

```
cd Demo_MSP242X_wiringPi_Hardware_IIC
```

```
sudo make
```

```
sudo ./2.42_SPI_IIC_OLED
```

```
pi@raspberrypi: ~/IIC/Demo_MSP242X_wiringPi_Hardware_IIC
pi@raspberrypi:~/IIC $
pi@raspberrypi:~/IIC $
pi@raspberrypi:~/IIC $
pi@raspberrypi:~/IIC $ cd Demo_MSP242X_wiringPi_Hardware_IIC/
pi@raspberrypi:~/IIC/Demo_MSP242X_wiringPi_Hardware_IIC $ sudo make
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/src/delay.c -o /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/delay.o -lwiringPi -I/home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/include
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/src/gui.c -o /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/gui.o -lwiringPi -I/home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/include
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/src/iic.c -o /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/iic.o -lwiringPi -I/home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/include
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/src/main.c -o /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/main.o -lwiringPi -I/home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/include
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/src/oled.c -o /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/oled.o -lwiringPi -I/home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/include
gcc -g -O0 -c /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/src/test.c -o /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/test.o -lwiringPi -I/home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/source/include
gcc -g -O0 /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/delay.o /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/gui.o /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/iic.o /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/main.o /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/oled.o /home/pi/IIC/Demo_MSP242X_wiringPi_Hardware_IIC/output/test.o -o 2.42_SPI_IIC_OLED -lwiringPi
pi@raspberrypi:~/IIC/Demo_MSP242X_wiringPi_Hardware_IIC $ ./2.42_SPI_IIC_OLED
```

### C、编译并运行 python 测试程序

如果使用 SPI 测试程序，则通过 SD 卡或者 FTP 工具（如 FileZilla）将资料包里 **Demo\_RaspberryPi/Demo\_SPI** 目录下的 **Demo\_MSP242X\_python\_Hardware\_SPI** 文件夹拷贝到 Raspberry Pi OS 里，然后在终端软件执行如下命令编译并运行程序。

使用 python3，执行如下命令：

```
cd Demo_MSP242X_python_Hardware_SPI/source/
```

```
sudo python3 show_char.py
```

```
sudo python3 show_graph.py
```

```
sudo python3 show_bmp.py
```

```
pi@raspberrypi:~ $ cd Demo_MSP242X_python_Hardware_SPI/source/
pi@raspberrypi:~/Demo_MSP242X_python_Hardware_SPI/source $ sudo python3 show_char.py
```

```
pi@raspberrypi:~/Demo_MSP242X_python_Hardware_SPI/source $ sudo python3 show_graph.py
```

```
pi@raspberrypi:~/Demo_MSP242X_python_Hardware_SPI/source $ sudo python3 show_bmp.py
```

使用 python2，执行如下命令：

```
cd Demo_MSP242X_python_Hardware_SPI/source/
```

```
sudo python show_char.py
```

```
sudo python show_graph.py
```

```
sudo python show_bmp.py
```

```
pi@raspberrypi:~ $ cd Demo_MSP242X_python_Hardware_SPI/source/
pi@raspberrypi:~/Demo_MSP242X_python_Hardware_SPI/source $ sudo python show_char.py

pi@raspberrypi:~/Demo_MSP242X_python_Hardware_SPI/source $ sudo python show_graph.py

pi@raspberrypi:~/Demo_MSP242X_python_Hardware_SPI/source $ sudo python show_bmp.py
```

如果使用 IIC 测试程序，则通过 SD 卡或者 FTP 工具（如 FileZilla）将资料包里 **Demo\_RaspberryPi/Demo\_IIC** 目录下的 **Demo\_MSP242X\_python\_Hardware\_IIC** 文件夹拷贝到 Raspberry Pi OS 里，然后在终端软件执行如下命令编译并运行程序。

使用 python3，执行如下命令：

```
cd Demo_MSP242X_python_Hardware_IIC/source/
```

```
sudo python3 show_char.py
```

```
sudo python3 show_graph.py
```

```
sudo python3 show_bmp.py
```

```
pi@raspberrypi:~/IIC $ cd Demo_MSP242X_python_Hardware_IIC/source/
pi@raspberrypi:~/IIC/Demo_MSP242X_python_Hardware_IIC/source $ sudo python3 show_char.py

pi@raspberrypi:~/IIC/Demo_MSP242X_python_Hardware_IIC/source $ sudo python3 show_graph.py

pi@raspberrypi:~/IIC/Demo_MSP242X_python_Hardware_IIC/source $ sudo python3 show_bmp.py
```

使用 python2，执行如下命令：

```
cd Demo_MSP242X_python_Hardware_IIC/source/
```

```
sudo python show_char.py
```

```
sudo python show_graph.py
```

```
sudo python show_bmp.py
```

```
pi@raspberrypi:~/IIC $ cd Demo_MSP242X_python_Hardware_IIC/source/
pi@raspberrypi:~/IIC/Demo_MSP242X_python_Hardware_IIC/source $ sudo python show_char.py

pi@raspberrypi:~/IIC/Demo_MSP242X_python_Hardware_IIC/source $ sudo python show_graph.py

pi@raspberrypi:~/IIC/Demo_MSP242X_python_Hardware_IIC/source $ sudo python show_bmp.py
```