


# Serial Expansion HAT

目录 ▾

说明	资料	FAQ	售后
----	----	-----	----

## Serial Expansion HAT



(<https://www.waveshare.net/shop/Serial-Expansion-HAT.htm>)

基本信息	
分类:	树莓派扩展板
功能:	UART
品牌:	Waveshare
板载接口	
RPi ( <a href="/wiki/Category:RPi%E6%8E%A5%E5%8F%A3">/wiki/Category:RPi%E6%8E%A5%E5%8F%A3</a> )	UART ( <a href="/wiki/Category:UART%E6%8E%A5%E5%8F%A3">/wiki/Category:UART%E6%8E%A5%E5%8F%A3</a> )
相关产品	

## 产品介绍

Serial Expansion HAT是一款专为树莓派设计的串口扩展板，可通过I2C扩展两路UART通道和8个可编程IO

### 特点

- 基于Raspberry Pi 40pin GPIO接口，适用于Raspberry Pi系列主板
- 板载SC16IS752，可通过I2C扩展两路UART通道和8个可编程IO，不需要占用额外管脚
- 通过改变地址跳线可同时接入多达16个模块，相当于扩展出32个串口
- 板载多个指示灯，方便查看UART扩展接口的工作状态

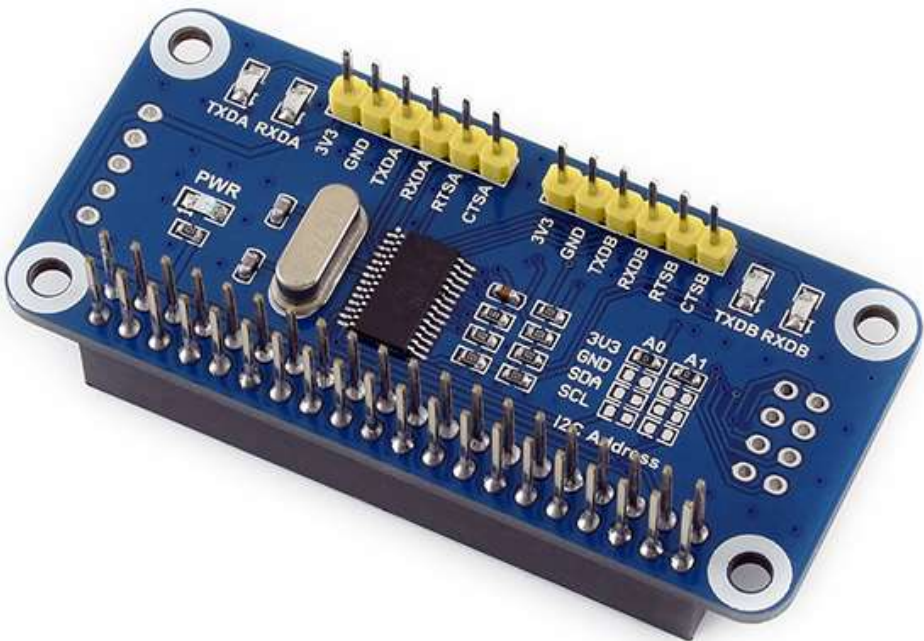
- 预留I2C控制接口，方便接入其他主控板
- 提供完善的配套资料手册(提供C与python例程)

产品参数

- 工作电压：3.3V
- 扩展芯片：SC16IS752
- 控制接口：I2C
- 产品尺寸：65mm × 30mm
- 过孔直径：3.0mm

资源介绍

SC16IS752是双通道高性能的UART扩展芯片，还能额外提供8个可编程IO口，支持SPI和I2C两种接口通信，本模块默认使用I2C接口，并可更改A0与A1的连线方式来改变I2C设备地址。



(/wiki/File:Serial-Expansion-HAT-intro.jpg)

接口介绍

标识	描述
3V3	3.3V电源
GND	电源地
TXDA	通道A数据输出端
RXDA	通道A数据输入端
RTSA	通道A请求发送

CTSA	通道A允许发送
TXDB	通道B数据输出端
RXDB	通道B数据输入端
RTSB	通道B请求发送
CTSB	通道B允许发送

## LED灯介绍

PWR: 电源指示灯

TXDA: 通道A数据发送指示灯

RXDA: 通道A数据接收指示灯

TXDB: 通道B数据发送指示灯

RXDB: 通道B数据接收指示灯

## I2C设备地址设置说明

A0, A1列为控制I2C硬件地址，通过焊接0R电阻选择地址，它们的对应关系如下表格：

**Table 32. SC16IS752/SC16IS762 address map**

A1	A0	SC16IS752/SC16IS762 I <sup>2</sup> C address (hex) <sup>[1]</sup>
V <sub>DD</sub>	V <sub>DD</sub>	0x90 (1001 000X)
V <sub>DD</sub>	V <sub>SS</sub>	0x92 (1001 001X)
V <sub>DD</sub>	SCL	0x94 (1001 010X)
V <sub>DD</sub>	SDA	0x96 (1001 011X)
V <sub>SS</sub>	V <sub>DD</sub>	0x98 (1001 100X)
V <sub>SS</sub>	V <sub>SS</sub>	0x9A (1001 101X)
V <sub>SS</sub>	SCL	0x9C (1001 110X)
V <sub>SS</sub>	SDA	0x9E (1001 111X)
SCL	V <sub>DD</sub>	0xA0 (1010 000X)
SCL	V <sub>SS</sub>	0xA2 (1010 001X)
SCL	SCL	0xA4 (1010 010X)
SCL	SDA	0xA6 (1010 011X)
SDA	V <sub>DD</sub>	0xA8 (1010 100X)
SDA	V <sub>SS</sub>	0xAA (1010 101X)
SDA	SCL	0xAC (1010 110X)
SDA	SDA	0xAE (1010 111X)

[1] X = logic 0 for write cycle; X = logic 1 for read cycle.

(/wiki/File:Serial\_expansion\_hat\_i2caddr.png)

详见datasheet:Page39 ([https://www.waveshare.net/w/upload/a/ad/SC16IS752\\_datasheet.pdf](https://www.waveshare.net/w/upload/a/ad/SC16IS752_datasheet.pdf))

可以看到表中的列出来的地址均为8位数据，在实际操作中的I2C地址7位，所以需要往最低位移一位。比如说第一个A1和A0接到VDD，表格中给出来的地址是0x90(1001 000X)，但是实际使用的时候，对应的设备地址应该时，100 1000，即0x48。

【注意】本模块默认焊接A0和A1到3.3V，默认I2C地址为：0X48  
通过计算，理论上可以在一个控制器上同时控制32个设备。

购买日期早于2019年12月31日为老版本，老版本由于PCB上中断管脚已经直连了，因此无法在树莓派上堆叠，需要使用杜邦线连接；  
购买日期晚于2019年12月31日为新版本，需要手动焊接更改默认的中断管脚，最大支持5个堆叠，如果需要叠加更多的，需要使用杜邦线连接  
对树莓派而言，使用杜邦线连接理论上不能达到连接32个，因为扩展一块就需要一个中断引脚，树莓派没有那么多的GPIO

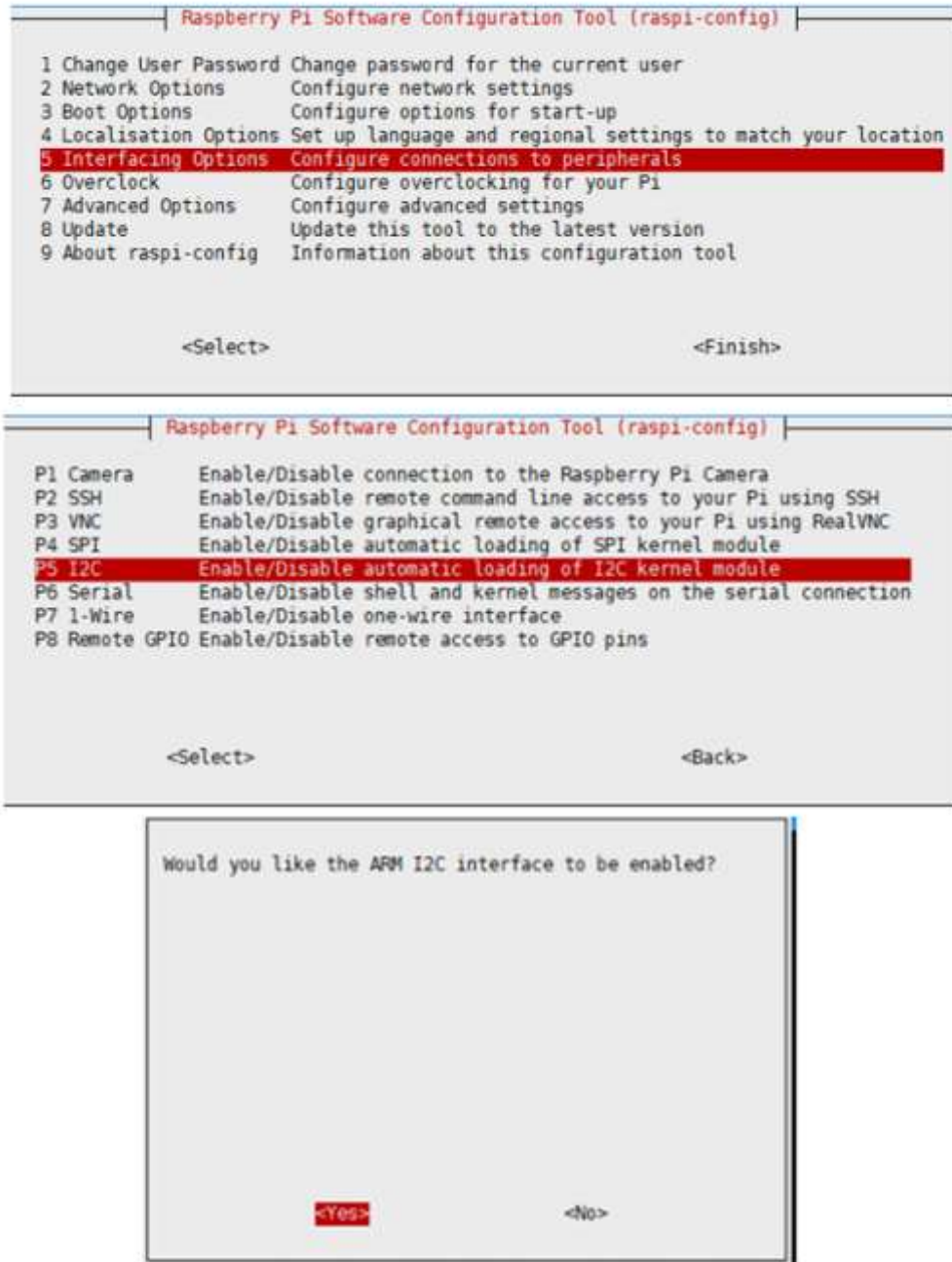
# 树莓派使用

## 打开I2C接口

---

在终端执行：

1. `sudo raspi-config`
2. 选择 Interfacing Options -> I2C ->yes 启动 i2C 内核驱动



(/wiki/File:RPI\_open\_i2c.png)

然后重启树莓派：

1. `sudo reboot`

## 安装库

- 安装BCM2835，打开树莓派终端，并运行以下指令

1. `wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.60.tar.gz`
2. `tar zxvf bcm2835-1.60.tar.gz`
3. `cd bcm2835-1.60/`
4. `sudo ./configure && sudo make && sudo make check && sudo make install`
5. # 更多的可以参考官网：<http://www.airspayce.com/mikem/bcm2835/>



## ■ 安装wiringPi

1. `sudo apt-get install wiringpi`
2. #对于树莓派2019年5月之后的系统（早于之前的可不用执行），可能需要进行升级：
3. `wget https://project-downloads.drogon.net/wiringpi-latest.deb`
4. `sudo dpkg -i wiringpi-latest.deb`
5. `gpio -v`
6. # 运行`gpio -v`会出现2.52版本，如果没有出现说明安装出错

## ■ python库

1. `sudo apt-get install python-dev`
2. `sudo apt-get install python-rpi.gpio`
3. `sudo apt-get install python-smbus`

## 生成设备

### ■ 在/boot/config.txt中加入如下

1. `sudo nano /boot/config.txt`
2. # addr根据A0A1实际焊接方式设置，默认是0x48
3. `dtoverlay=sc16is752-i2c,int_pin=24,addr=0x48`
4. # 重启设备
5. `sudo reboot`

### ■ 重启后，SC16IS752的驱动会加载到系统内核中，此时可以运行`ls /dev`查看一下 将会多出如下设备：

```

pi@raspberrypi:~$ ls /dev/
autofs          gpiochip3       loop            ram11           tty19           tty34           tty5            tty8            vcs5
btrfs-control   gpiochip0       mem            ram12           tty2            tty35           tty50           tty9            vcs6
cachefiles      i2c-1           memory_bandwidth ram13           stderr          tty20           tty36           ttyAMA0         vcs7
console         initctl         mmcblk0        ram14           stdin           tty21           tty37           tty52           ttyprintk      vcsa
cpu_dma_latency loop            mmcblk0p1      ram15           stdout          tty22           tty38           tty53           ttySC0         vcsa1
cuse            kmsg           mmcblk0p2      ram2            tty             tty23           tty39           tty54           ttySC1         vcsa2
fd              loop0          network_latency ram3            tty0            tty24           tty4            tty55           uinput         vcsa3
full            loop1          network_throughput ram4            tty1            tty25           tty40           tty56           urandom        vcsa4
fuse            loop2          null           ram5            tty10           tty26           tty41           tty57           vchiq          vcsa5
gpiochip0       loop3          ppp            ram6            tty11           tty27           tty42           tty58           vcio           vcsa6
gpiochip1       loop4          ptmx           ram7            tty12           tty28           tty43           tty59           vcsa7          vchi
gpiochip2       loop5          ram0           ram8            tty13           tty29           tty44           tty6            vc-mem         vcsa8
loop            loop6          ram1           random          tty14           tty3            tty45           tty60           vcs            vchi
loop-control   loop7          ram10          rftkill         tty15           tty30           tty46           tty61           vcs1           watchdog
loop0          loop8          ram11          serial1         tty16           tty31           tty47           tty62           vcs2           watchdog0
loop1          loop9          ram12          tty1            tty17           tty32           tty48           tty63           vcs3           zero
loop2          mem           ram13          tty18           tty19           tty33           tty49           tty7            vcs4

```

(/wiki/File:Serial\_expansion\_hat\_finddev.png)

其中GPIOchip对应的是8个GPIO驱动（2020-02-13之前的生成的是gpiochip3,之后的系统生成的是gpiochip2），ttySC0和ttySC1为两个串口驱动

如果需要堆叠的话，第二个模块不能直接插在树莓派，由于PCB上使用的中断引脚是一样，会导致无法识别出多个设备。

此时应该使用杜邦线连接，并保证中断引脚不一致。同时config.txt中的命令应该修改为  
`dtoverlay=sc16is752-i2c,int_pin=xx,addr=0x48`  
 xx为你实际插的BCM编码引脚



1. `cd c/gpio/`
2. `make clean`
3. `make`
4. `sudo ./main`

运行程序后，程序会翻转GPIO0的电平，如果在GPIO0上接上LED灯，会发现LED灯闪烁

#### ■ python

1. `cd python/GPIO`
2. `sudo python gpio.py`

运行程序后，程序翻转GPIO0的电平几次，如果GPIO0上接上LED灯，会发现LED灯闪烁五次之后常亮，另外GPIO1被设置为输入模式，你可以将GPIO1分别接到3.3V和GND查看一下状态

## 叠加教程

如果你不仅需要一个，你需要扩展多个串口需要按照如下操作（适用于新版本）。安装i2c-tools，方便查看I2C地址，在控制台输入：

1. `sudo apt-get install i2c-tools`

等待安装成功，执行如下命令查看I2C地址。

1. `sudo i2cdetect -y 1`

如果默认已经插入了一个Serial Expansion HAT,那么执行如下命令会得到如下的效果：



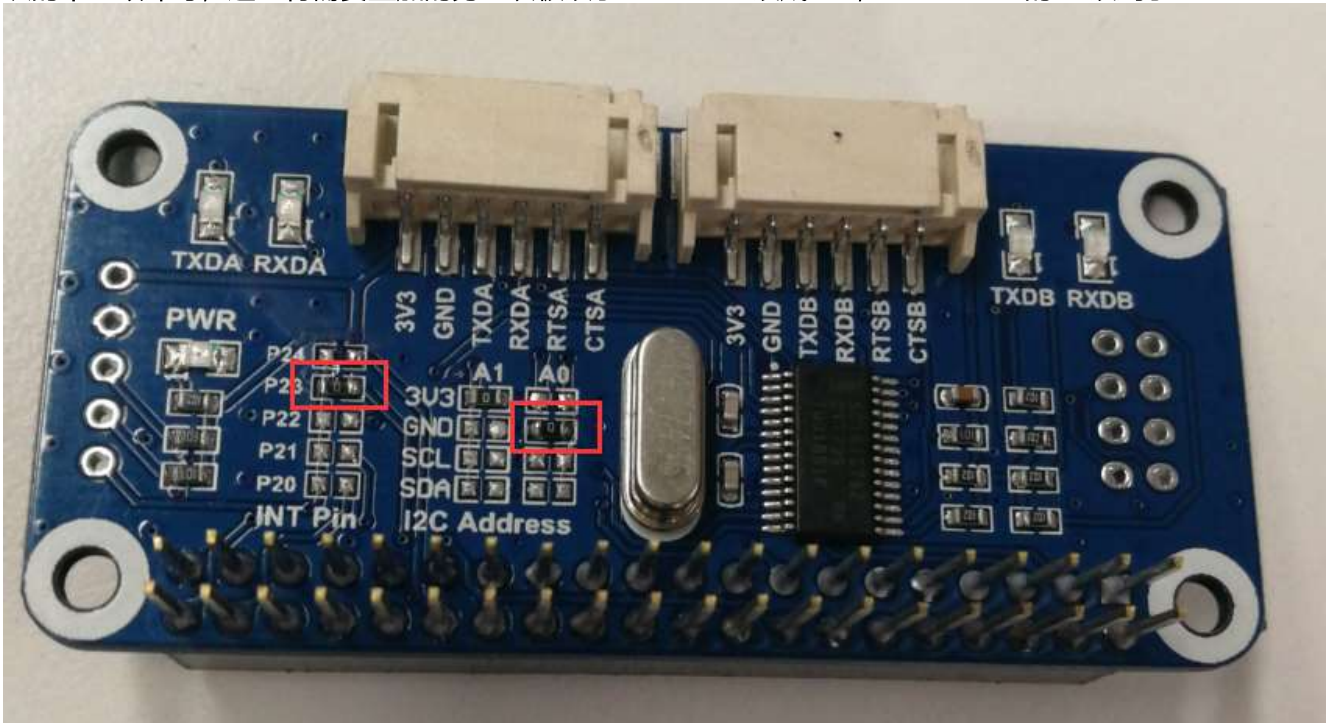
```
pi@raspberrypi:~$ sudo i2cdetect -y 1
 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- UU -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- --
pi@raspberrypi:~$ ls /dev/ttySC*
/dev/ttySC0 /dev/ttySC1
```

(/wiki/File:Serial expansion hat add0.png)

其中0X48的设备显示成了UU，说明系统已经加载成功了。运行ls /dev/ttySC\*可以查看到已经生成了两个串口设备了。



在叠加第二块模块的之前，需要修改相应的硬件设置，将板子上的默认INT与I2C address的配置改成与默认的不一致即可，这里将需要叠加的另一块板改了：INT PIN改成P23,I2C Address的A0改到了GND:



(/wiki/File:Serial\_expansion\_hat\_add.png)

修改完成之后，将模块叠加在之前的模块上，运行命令：

```
pi@raspberrypi:~$ sudo i2cdetect -y 1
0 1 2 3 4 5 6 7 8 9 a b c d e f
00: -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- UU 49 -- -- --
50: -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- --
```

(/wiki/File:Serial\_expansion\_hat\_add1.png)

新的模块的I2C地址为0X49,前面已经修改了中断管脚为23，那么我们就可以打开/boot/config.txt中加入如下配置：

```
1. sudo nano /boot/config.txt
```

在之前配置下方加入如下，保存并退出(ctrl +x,然后执行Y，并回车)：

```
1. dtoverlay=sc16is752-i2c,int_pin=23,addr=0x49
```

```
dtoverlay=sc16is752-i2c,int_pin=24,addr=0x48
dtoverlay=sc16is752-i2c,int_pin=23,addr=0x49
```

(/wiki/File:Serial\_expansion\_hat\_add2.png)

重启树莓派

```
1. sudo reboot
```

执行如下命令，会出现4个设备号，叠加即成功了。

```
pi@raspberrypi:~ $ ls /dev/ttySC*  
/dev/ttySC0 /dev/ttySC1 /dev/ttySC2 /dev/ttySC3
```

(/wiki/File:Serial\_expansion\_hat\_add3.png)

由于板载只可设置5个不同的电阻，因此最大叠加数为5个，如需要添加更多的，需要使用杜邦线连接

---

3个分类 (/wiki/Special:Categories): 模块 (/wiki/Category:%E6%A8%A1%E5%9D%97)

树莓派 (/wiki/Category:%E6%A0%91%E8%8E%93%E6%B4%BE)

树莓派扩展板

(/wiki/Category:%E6%A0%91%E8%8E%93%E6%B4%BE%E6%89%A9%E5%B1%95%E6%9D%BF)



([https://www.waveshare.net/help\\_center/join\\_us.htm](https://www.waveshare.net/help_center/join_us.htm))

Copyright © 2020 深圳市微雪电子有限公司 (<http://waveshare.net>) 版权所有