2-CH RS485 HAT

This is a dual-channel isolated RS485 extension board specially designed for Raspberry PI, which adopts SC16IS752+SP3485 solution, embed with protection circuits such as power supply isolation, ADI magnetical isolation, and TVS diode, etc. It is easy to control the 2-channel RS485 for auto transceiving via SPI interface. Due to its fast communication, stability, reliability, and safety, it is an ideal choice for fields like industrial automation.

Feature

- Standard Raspberry Pi 40PIN GPIO extension header, supports Raspberry Pi series boards.
- Adopts SC16IS752 + SP3485 dual-chip combination, converts SPI to RS232, data rate up to 921600bps.
- Supports manual or automatic data sending and receiving, which can be set by a DIP switch.
- Onboard TVS (Transient Voltage Suppressor), effectively suppresses surge voltage and transient spike voltage in the circuit, lightning proof & anti-electrostatic.
- Onboard resettable fuses and protection diodes ensure a stable output of current and voltage, prevent overcurrent and overvoltage, and improve shock resistance.

- Onboard power indicator and serial port transceiver indicator for checking the module power and communication status.
- With SPI control pins, for connecting with host control boards like Arduino.
- Comes with development resources and a manual (examples in C and Python).

Parameter

- UART expansion chip: SC16IS752
- RS485 transceiver: SP3485
- Communication interface: SPI
- Data rate: 300 ~ 921600 bps
- Operating voltage: 3.3V / 5V
- Dimensions: 65mm × 56.5mm
- Mounting hole size: 3.0mm

Interfaces

PIN	SYMBOL	Description
1	VCC	3.3V/5V Power
2	GND	Ground

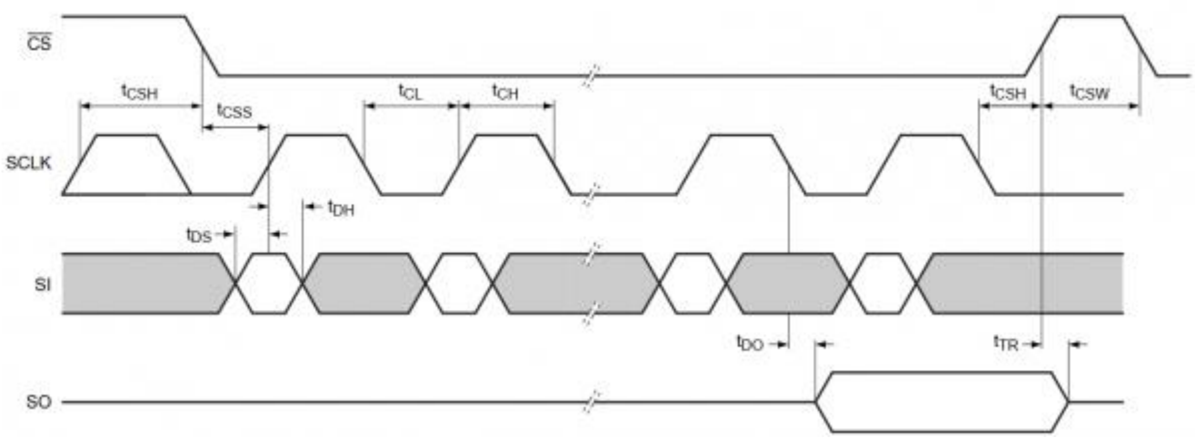
3	SCLK	SPI Clock input
4	MOSI	SPI Data input
5	MISO	SPI Data output
6	CS	SPI Chip Selection
7	IRQ	Interrupt output (Interrupt Request)
8	EN1	Enable Channel 1
9	EN2	Enable Channel 2

Working principle

- ### Introduction

This product adopts SC16IS752 as a controller. SC16IS752 is a dual-channel high-performance UART expansion chip that supports SPI and I2C. This module adopts the SPI interface, and onboard power isolation, ADI magnetic coupler isolation. It also onboards TVS (transient voltage suppression tube), self-recovery fuses, protection diodes, and an automatic transceiver switching circuit. It can effectively suppress the surge voltage and transient peak voltage in the circuit, prevent lightning and static electricity, prevent over-voltage, improve the anti-impact ability, and can conduct signal isolation with high dependence, strong anti-interference, and low power consumption advantages, etc.

Communication protocol



- CS：Slave chip selection, when CS is low, the slave chip is enabled.
- SCLK：SPI communication clock
- MOSI/SI：SPI Communication master sends, slave receives
- MOSI/SI：SPI Communication master receives, slave sends
- Timing Sequence：CPHL=0， CPOL=0 （SPI0）

Working with RPI

How to use

We provide C and Python demo codes for Raspberry Pi. A quick testing example is provided in python.

Hardware Connection

232 PIN	Raspberry Pi(BCM)	Description
VCC	5V	3.3/5V Power Input
GND	GND	Ground
SCLK	P21 (SPI1 SCLK)	SPI Clock Signal Input
MOSI	P20 (SPI1 MOSI)	SPI Data Input
MISO	P19 (SPI1 MISO)	SPI Data Output
CS	P18 (SPI1 CS)	SPI Chip Select
IRQ	P24	Interrupt Output
EN1	P27	Channel 1 transceiver enable: high level transmit enable, low level receive enable
EN2	P22	Channel 2 transceiver enable: high level transmit enable, low level receive enable

Software setup

- Open the terminal and modify config.txt file by commands:

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

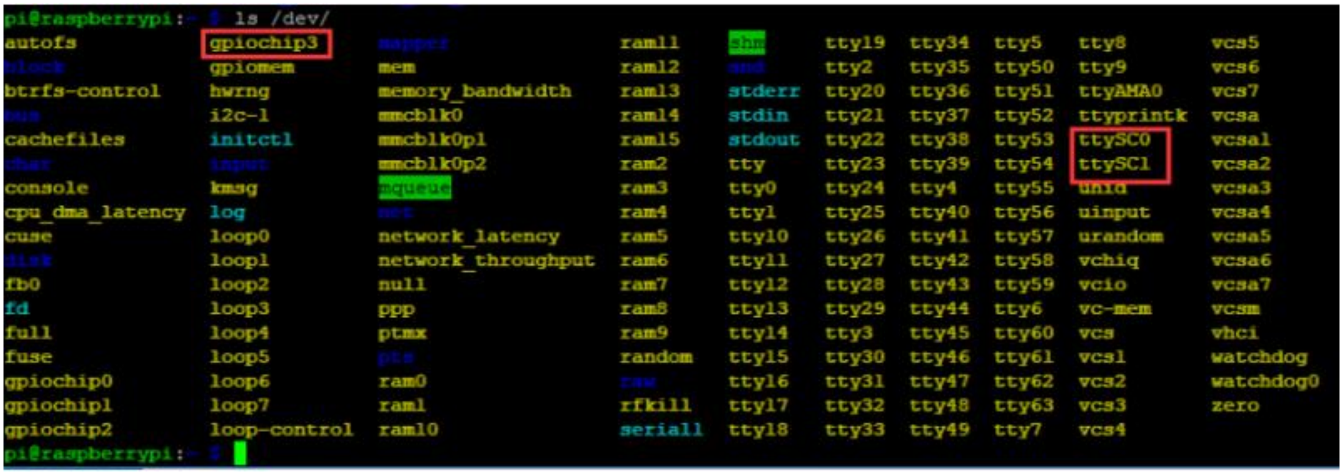
- Add the line below to the file, the int_pin should be set according to the actual welding:

```
dtoverlay=sc16is752-spi1,int_pin=24
```

- Then restart Raspberry Pi

```
sudo reboot
```

- After rebooting, the driver of SC16IS752 will be loaded into the system kernel. You can run command **ls /dev** to check the following devices:



```
pi@raspberrypi:~ $ ls /dev/
autofs           gpiochip3     mapper              ram11    shm     tty19  tty34  tty5   tty8       vcs5
block            gpiomem       mem                 ram12    snd     tty2   tty35  tty50  tty9       vcs6
btrfs-control    hwrng         memory_bandwidth    ram13    stderr  tty20  tty36  tty51  ttyAMA0    vcs7
bus              i2c-1         mmcblk0             ram14    stdin   tty21  tty37  tty52  ttyprintk  vcsa
cachefiles       initctl       mmcblk0p1           ram15    stdout  tty22  tty38  tty53  ttySC0     vcsa1
char             input         mmcblk0p2           ram2     tty     tty23  tty39  tty54  ttySC1     vcsa2
console          kmsg          mqueue              ram3     tty0    tty24  tty4   tty55  uhid       vcsa3
cpu_dma_latency  log           net                 ram4     tty1    tty25  tty40  tty56  uinput     vcsa4
cuse             loop0         network_latency     ram5     tty10   tty26  tty41  tty57  urandom    vcsa5
disk             loop1         network_throughput  ram6     tty11   tty27  tty42  tty58  vchiq      vcsa6
fb0              loop2         null                ram7     tty12   tty28  tty43  tty59  vcio       vcsa7
fd               loop3         ppp                 ram8     tty13   tty29  tty44  tty6   vc-mem     vcsm
full             loop4         ptmx                ram9     tty14   tty3   tty45  tty60  vcs        vhci
fuse             loop5         pts                 random   tty15   tty30  tty46  tty61  vcs1       watchdog
gpiochip0        loop6         ram0                raw      tty16   tty31  tty47  tty62  vcs2       watchdog0
gpiochip1        loop7         ram1                rfkill   tty17   tty32  tty48  tty63  vcs3       zero
gpiochip2        loop-control  ram10               serial1  tty18   tty33  tty49  tty7   vcs4
pi@raspberrypi:~ $
```

Install Libraries

- Install wiringpi

```
sudo apt-get install wiringpi
# An upgrade may be required for raspberry PI 4B:
cd /tmp
```

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

sudo dpkg -i wiringpi-latest.deb

gpio -v

# Running gpio-v to check if the version is 2.52, If it is not, you need to check the installation again.
```

- Install the python2 library

```
sudo apt-get update

sudo apt-get install python-pip

sudo pip install RPi.GPIO

sudo apt-get install python-serial
```

- Install the python3 library

```
sudo apt-get update

sudo apt-get install python3-pip

sudo pip3 install RPi.GPIO

sudo apt-get install python3-serial
```

Test

- Download and run the examples:

```
sudo apt-get install p7zip-full

wget https://www.waveshare.com/w/upload/4/44/2-CH_RS485_HAT_code.7z
```

```
7z x 2-CH_RS485_HAT_code.7z

sudo chmod 777 -R  2-CH_RS485_HAT

cd 2-CH_RS485_HAT/
```

- You can also clone the project from our Github:

```
sudo git clone https://github.com/waveshare/2-CH-RS485-HAT

cd 2-CH-RS485-HAT/
```

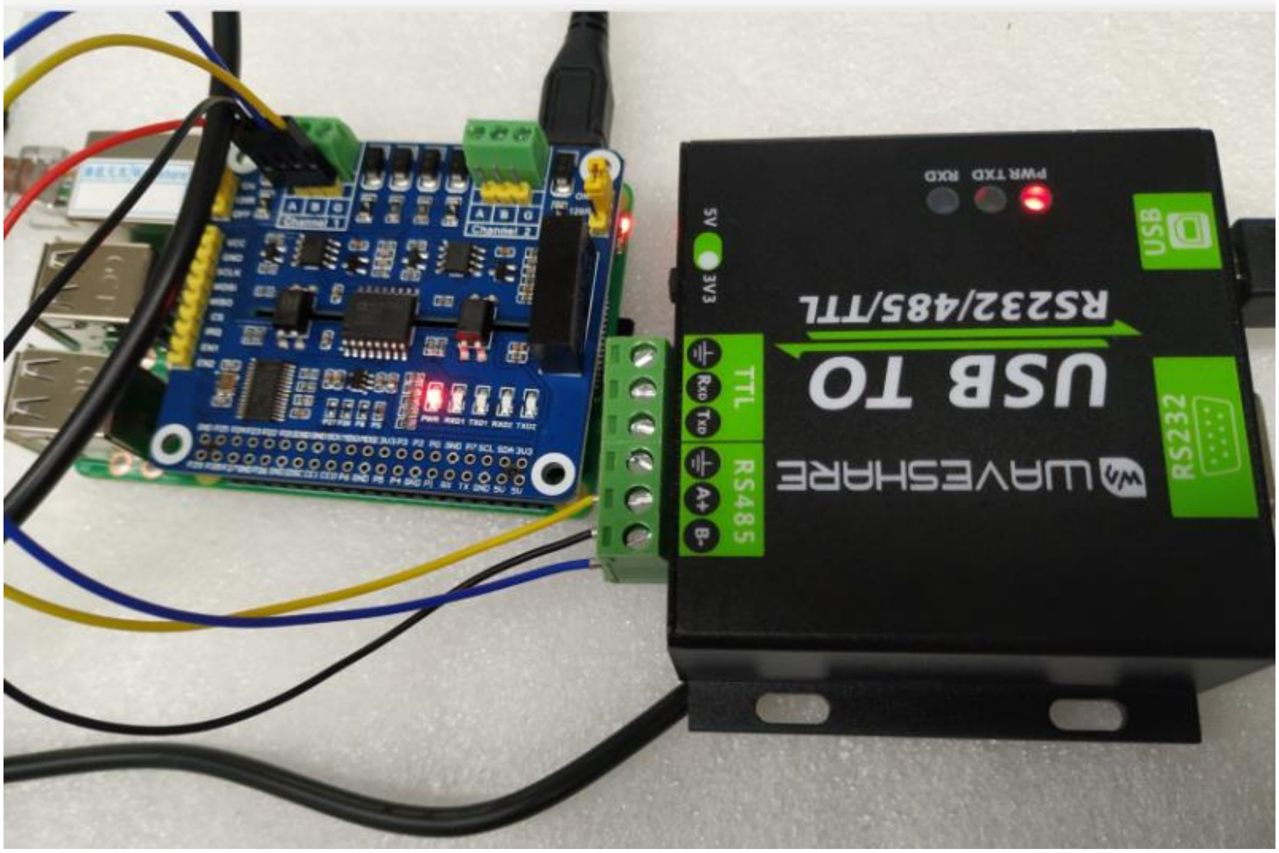
- C program

```
cd c

make clean

make

sudo ./main
```

- Python program

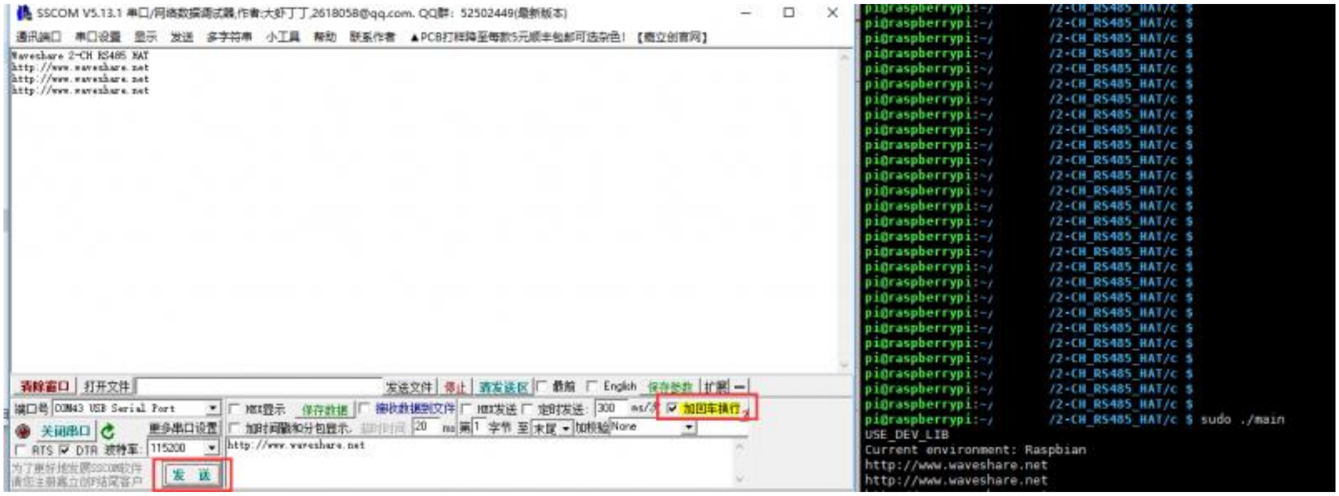
```
cd python

cd examples

sudo python main.py
```

Hardware connection: Channel 1 of the 2-CH RS485 HAT is connected to USB TO RS232/485/TTL:



Connect USB TO RS232/485/TTL to the computer, open the serial port assistant software, select the corresponding serial port, and set the baud rate to 115200.

- Run the C program, the data sent by computer will all be received by Raspberry Pi, as below:


SSCOM V5.13.1 串口/网络数据调试器,作者:大虾丁丁,2618058@qq.com. QQ群: 52502449(最新版本)
通讯端口 串口设置 显示 发送 多字符串 小工具 帮助 联系作者 ▲PCB打样降至每款5元顺丰包邮可选杂色! 【嘉立创官网】
清除窗口 打开文件 发送文件 停止 清发送区 □ 最前 □ English 保存参数 扩展 —
□ HEX显示 保存数据 □ 接收数据到文件 □ HEX发送 □ 定时发送:
□ 加时间戳和分包显示. 超时时间 第 字节 至 末尾 加校验
关闭串口 更多串口设置
为了更好地发展SSCOM软件
请您注册嘉立创P结尾客户
发 送

```
Waveshare 2-CH RS485 HAT
http://www.waveshare.net
http://www.waveshare.net
http://www.waveshare.net
```

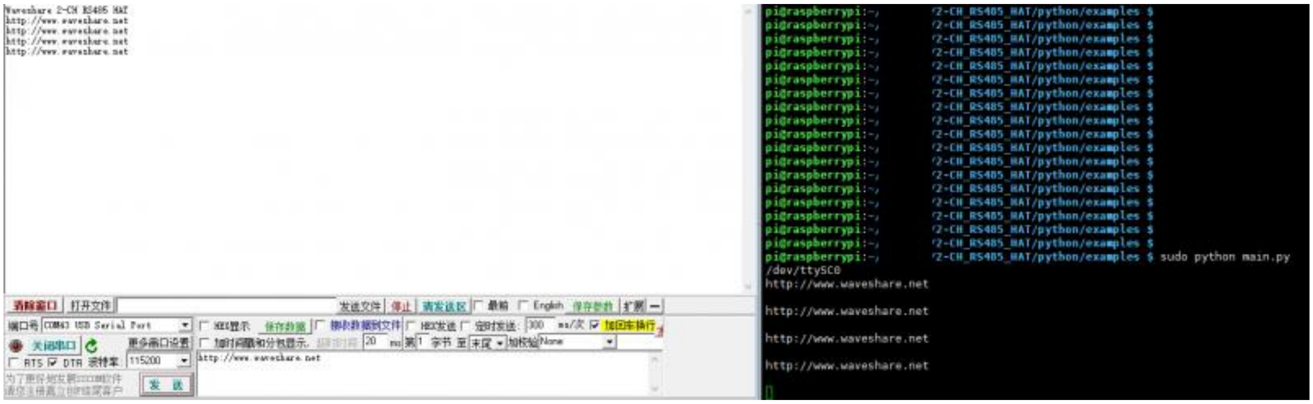
端口号 COM43 USB Serial Port
□ RTS ☑ DTR 波特率: 115200
定时发送: 300 ms/次 ☑ 加回车换行
20 ms 第 1 字节 至 末尾 加校验 None

```
http://www.waveshare.net
```

```
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $
pi@raspberrypi:~/        /2-CH_RS485_HAT/c $ sudo ./main
USE_DEV_LIB
Current environment: Raspbian
http://www.waveshare.net
http://www.waveshare.net
```

Note: You need to press Enter for a line feed, otherwise the data will not be returned; the program directory of the example is based on the actual.

- Run the main.py, the data sent by computer will all be received by Raspberry Pi, as below:


清除窗口 打开文件 发送文件 停止 清发送区 □ 最前 □ English 保存参数 扩展 —
□ HEX显示 保存数据 □ 接收数据到文件 □ HEX发送 □ 定时发送:
□ 加时间戳和分包显示. 超时时间 第 字节 至 末尾 加校验
关闭串口 更多串口设置
为了更好地发展SSCOM软件
请您注册嘉立创P结尾客户
发 送

```
Waveshare 2-CH RS485 HAT
http://www.waveshare.net
http://www.waveshare.net
http://www.waveshare.net
http://www.waveshare.net
```

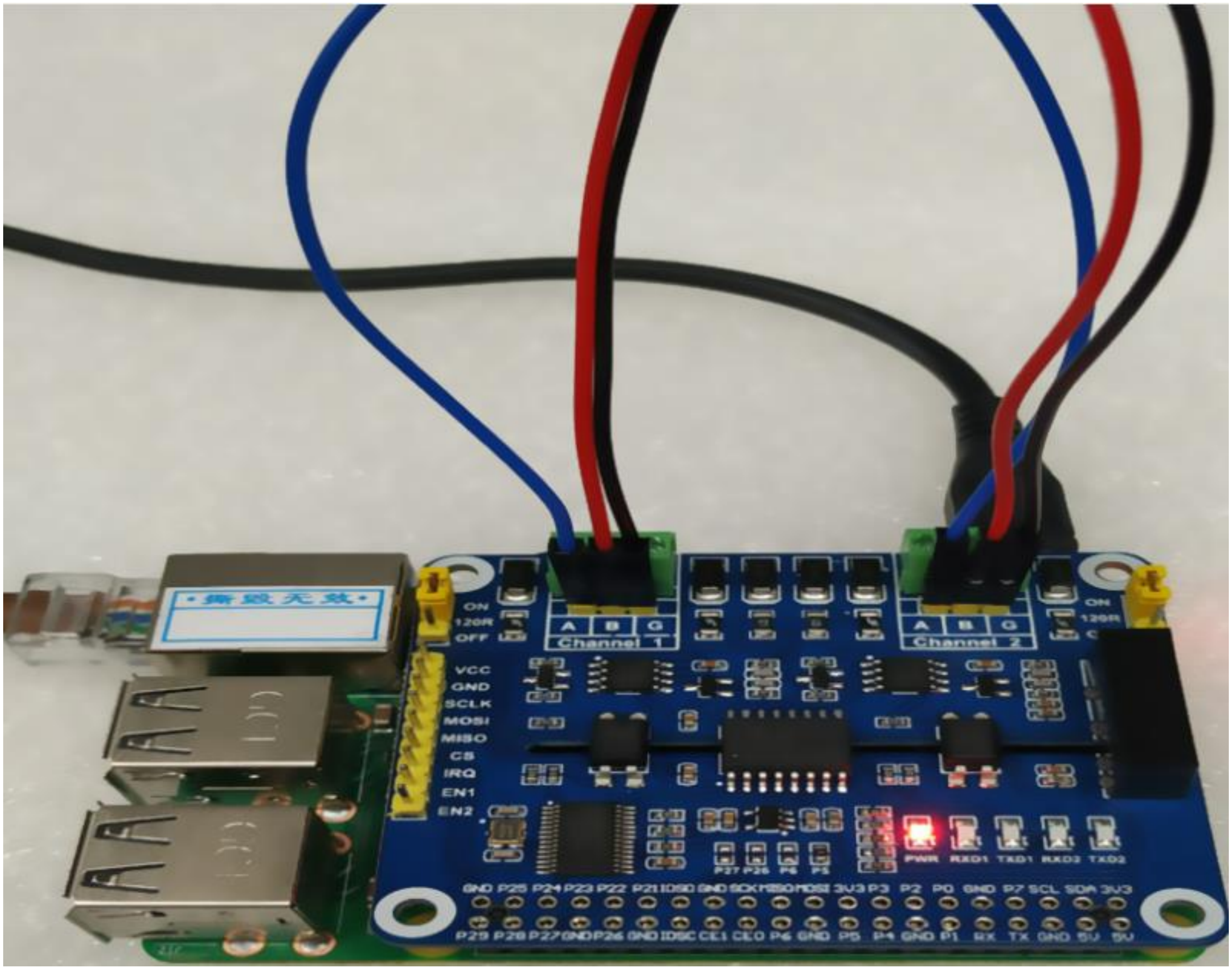
端口号 COM43 USB Serial Port
□ RTS ☑ DTR 波特率: 115200
定时发送: 300 ms/次 ☑ 加回车换行
20 ms 第 1 字节 至 末尾 加校验 None

```
http://www.waveshare.net
```

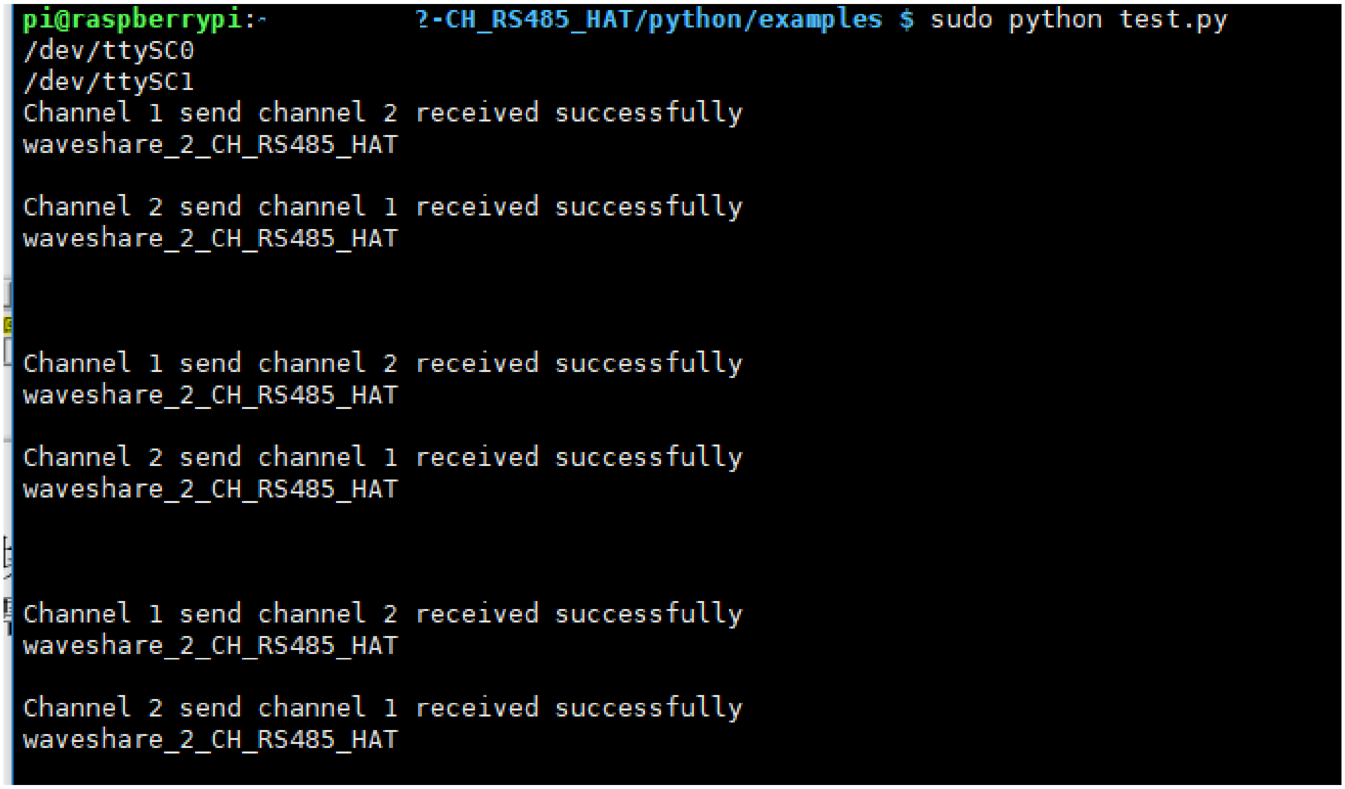
```
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/        /2-CH_RS485_HAT/python/examples $ sudo python main.py
/dev/ttySC0
http://www.waveshare.net

http://www.waveshare.net

http://www.waveshare.net

http://www.waveshare.net

```

Note: You need to press Enter for a line feed, otherwise the data will not be returned; the program directory of the example is based on the actual.

If you don't have other RS485 devices, you can choose the test method as follow by connecting channel 1 with channel 2:



- Running result of test.py :



```
pi@raspberrypi:~          ?-CH_RS485_HAT/python/examples $ sudo python test.py
/dev/ttySC0
/dev/ttySC1
Channel 1 send channel 2 received successfully
waveshare_2_CH_RS485_HAT

Channel 2 send channel 1 received successfully
waveshare_2_CH_RS485_HAT



Channel 1 send channel 2 received successfully
waveshare_2_CH_RS485_HAT

Channel 2 send channel 1 received successfully
waveshare_2_CH_RS485_HAT



Channel 1 send channel 2 received successfully
waveshare_2_CH_RS485_HAT

Channel 2 send channel 1 received successfully
waveshare_2_CH_RS485_HAT
```

Resources

- ### Documentation
  - Schematic
  - 3D Drawings
- ### Demo code
  - Demo code
  - Github
- ### Datasheets
  - SP3481_SP3485.pdf
  - SC16IS752_datasheet.pdf

FAQ

Question:485 speed is too high to communicate?

Answer:

If the automatic transceiver is used, you will find that it can only reach 921600 bps at most, and the manual transceiver can reach 2M when testing.

Question: In the semi-automatic mode (manual mode), the green RXD light or the green TXD light is always on?

Answer:

In the semi-automatic mode, the voltage on the EN pin determines whether RS485 is transmitted or received. In the transmit mode, the RXD green light may be always on. In the receive mode, the TXD green light may be always on, which is normal.

Question:485 communication is abnormal, what should I do?

Answer:

1. Check whether A and B of 485 correspond to the controlled devices.

2. You can use the USB to 485 device to communicate with the module first to ensure that there is no problem with the settings of the Raspberry Pi;

3. Check the setting of odd and even bit parity of serial communication parameters.

Question: Can the Ubuntu system be installed on the Raspberry Pi? Why can't I find the config.txt file in the boot folder?

Answer:

1. The Raspberry Pi is installed with the mainstream ubuntu system and can be used.

2. Ubuntu's config.txt file is usually in the /boot/firmware folder.

3. You also can use the SD card of the Raspberry Pi to read and change the config.txt file under the computer (or other hosts that can recognize the SD card) through a card reader.