2-CH RS485 HAT

Introduction

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.

Interfaces

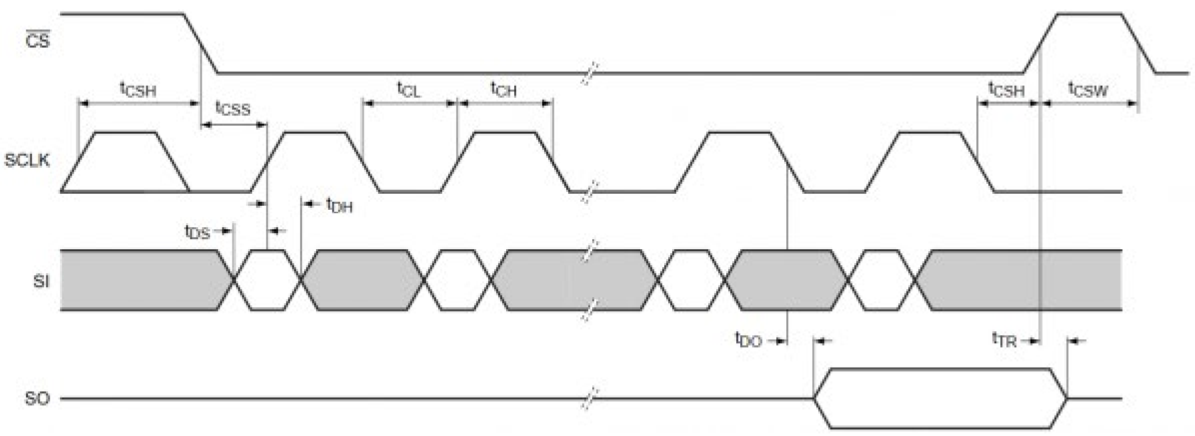
VCC	3.3V
PIN	Description
VCC	3.3V/5V Power
GND	Ground
SCLK	SPI Clock input
MOSI	SPI Data input
MISO	SPI Data output
CS	SPI Chip Selection
IRQ	Interrupt output (Interrupt Request)
EN1	Channel 1 output enable
EN2	Channel 2 output enable

Working principle

Introduction

This product adopts SC16IS752 as controller. SC16IS752 is a dual-channel high-performance UART expansion chip that supports SPI and I2C interfaces communication. This module uses SPI interface. Onboard power isolation, ADI magnetic coupler isolation, onboard TVS (transient voltage suppression tube), self-recovery fuses, protection diodes, and 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, can conduct signal isolation, with high dependence, strong anti-interference, 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）

How to use

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

Hardware Connection

To run examples, you should prepare an external RS485 to UART module, connect it to Channel 1 of 2-CH RS485 HAT.

If you test the 2-CH RS485 HAT with the text.py example, you need to wire Channel 1 and Channel 2 of 2-CH RS485 HAT.

485 PIN	Raspberry Pi(BCM)
VCC	5V
GND	GND
SCK	P21 (SPI1 SCLK)
MOSI	P20 (SPI1 MOSI)
MISO	P19 (SPI1 MISO)
CS	P18 (SPI1 CS)
IRQ	P24
EN1	P27
EN2	P22

Software setup

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

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

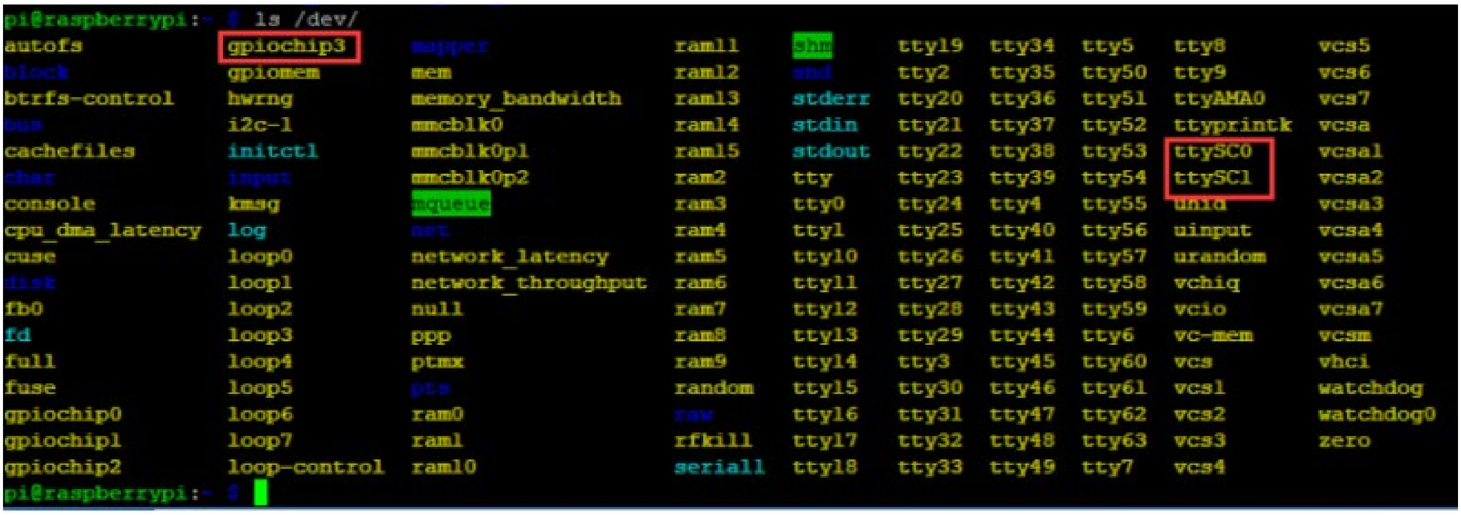
- Add the line below to the file, the int_pin should be set according to the actual welding:

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

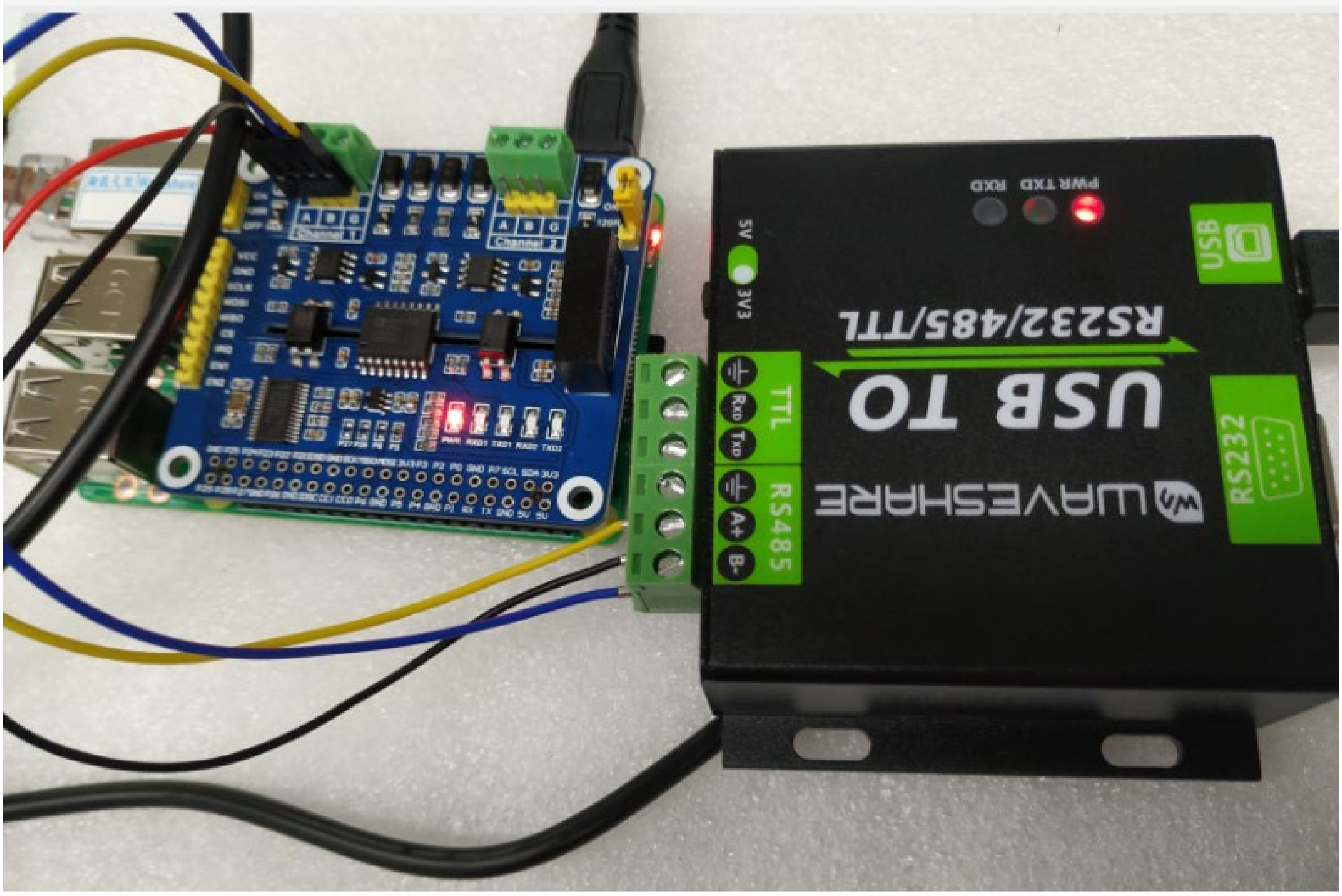
- C program

```
cd c
make clean
make
sudo ./main
```

- Python program

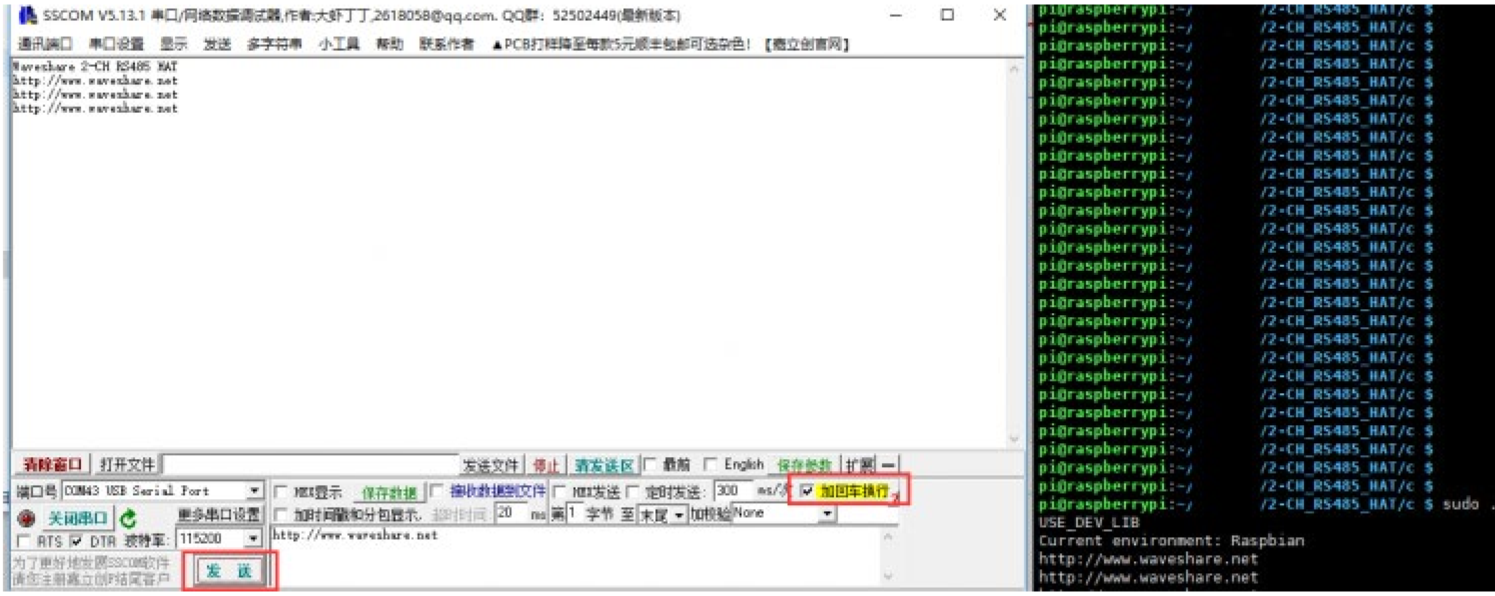
```
cd python
cd examples
sudo python main.py
```

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



Connect USB TO 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)

```
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

□ HEX显示 □ 加时间戳和分包显示 □ HEX发送 □ 定时发送: 300 ms/次 ☑ 加回车换行

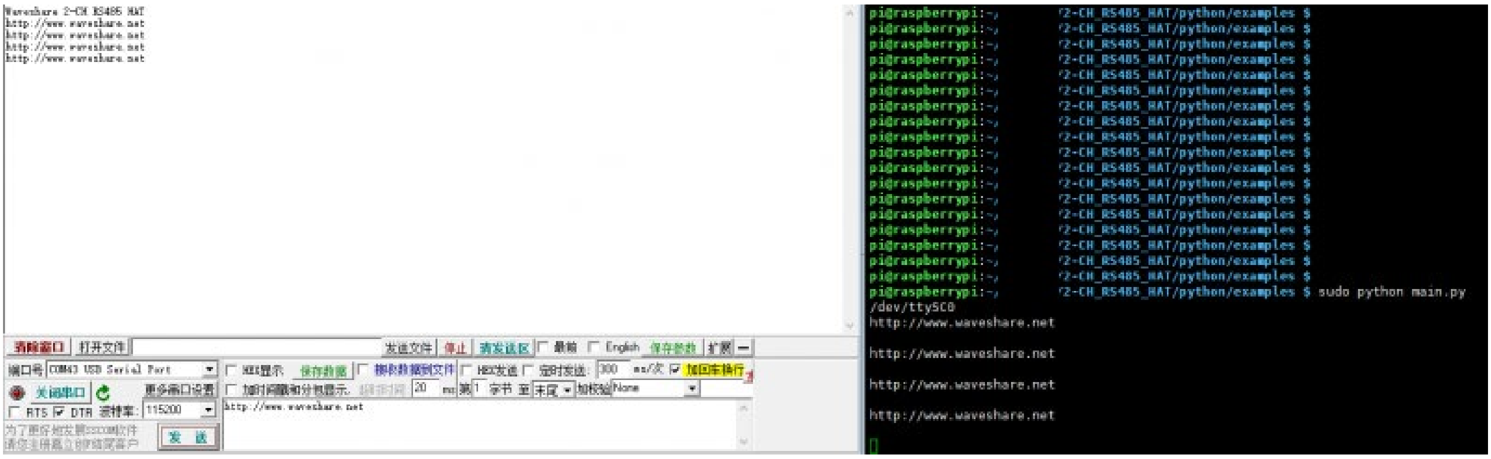
超时时间 20 ms 第1 字节 至 末尾 加校验 None

```
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 .
USE_DEV_LIB
Current environment: Raspbian
http://www.waveshare.net
http://www.waveshare.net
```

Note: The path of the samples is based on the actual directory;

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



```
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

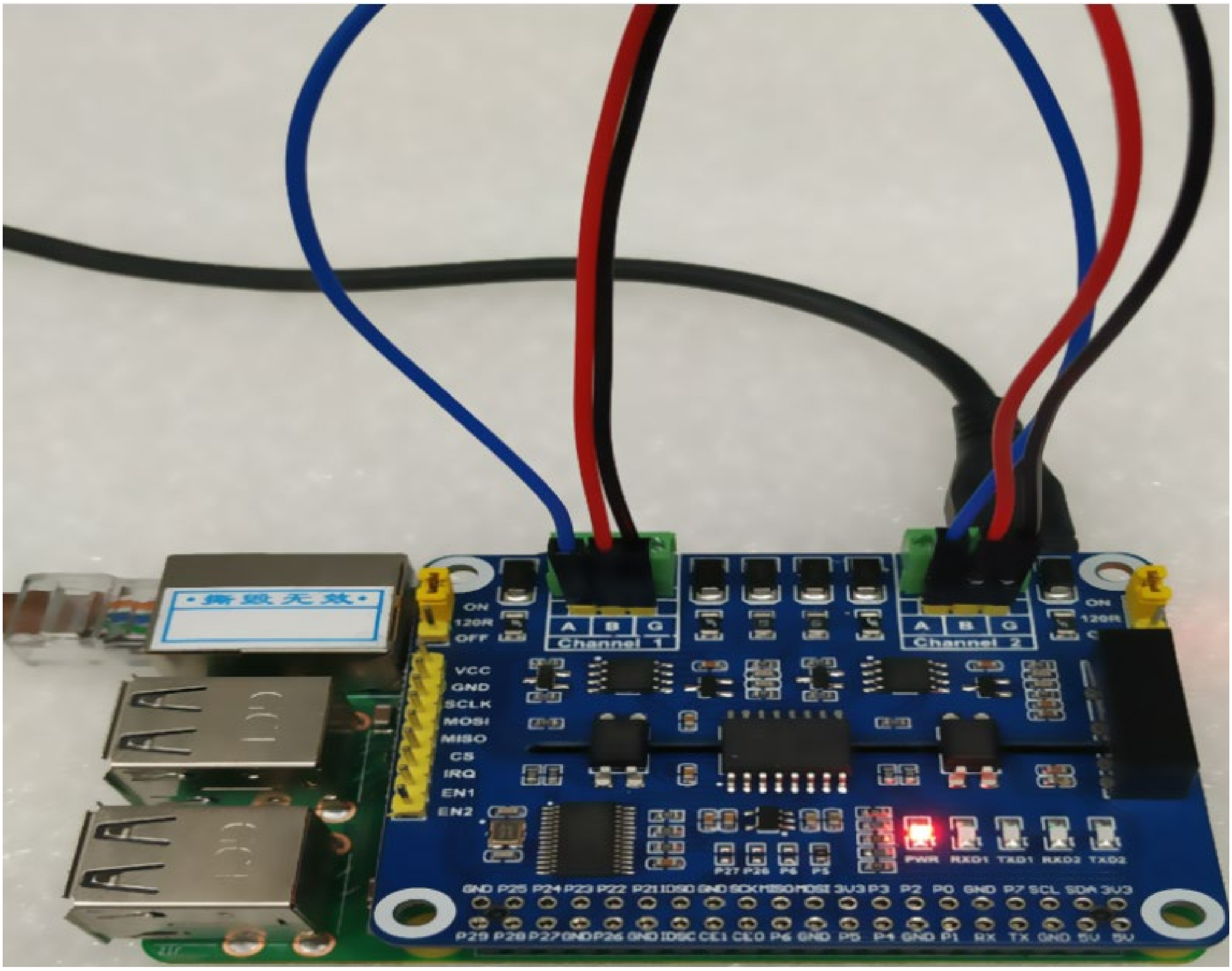
☐ HEX显示 ☐ HEX发送 ☐ 定时发送: 300 ms/次 ☑ 加回车换行

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

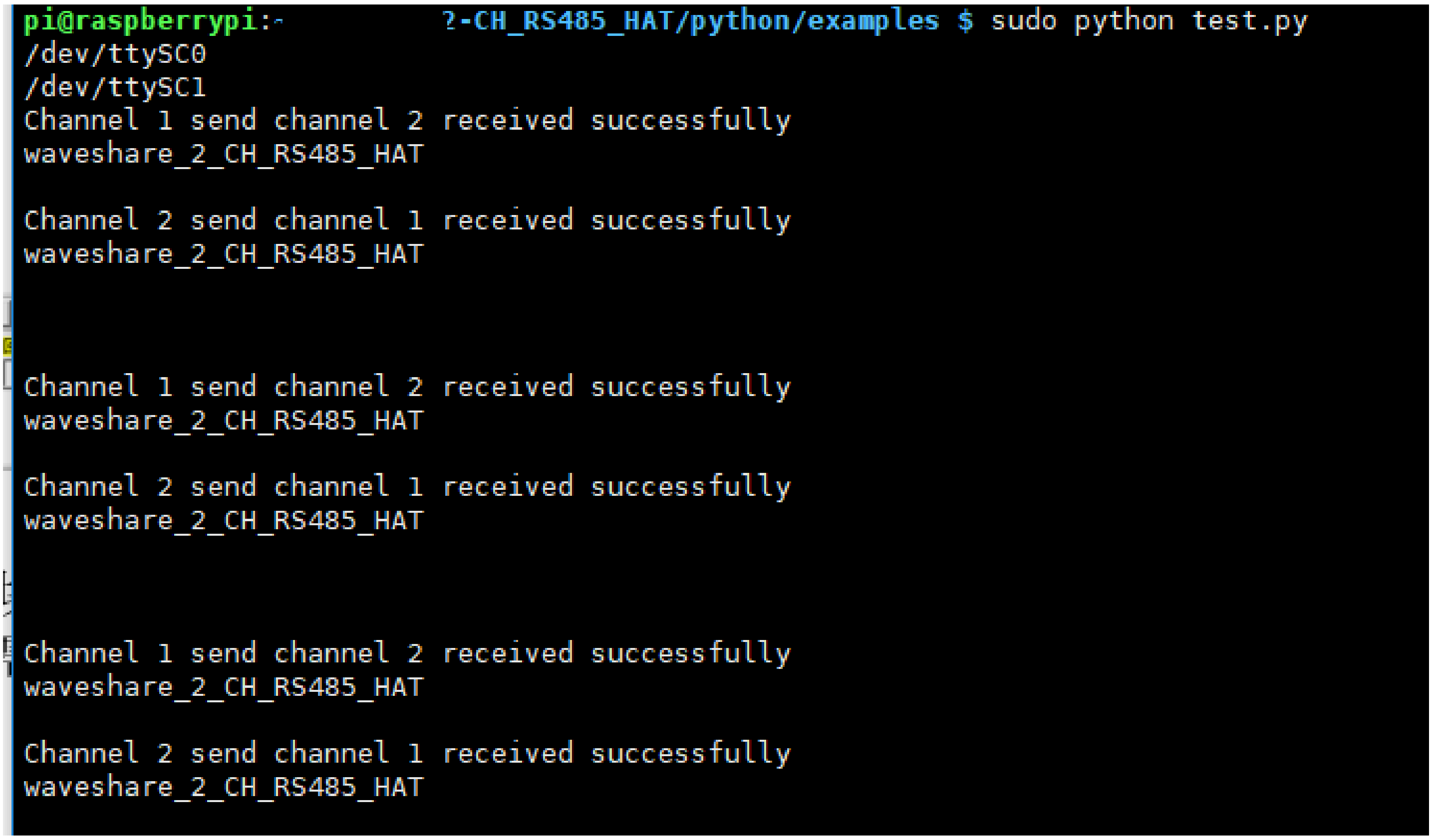
```
pi@raspberrypi:~/      /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/      /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/      /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/      /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/      /2-CH_RS485_HAT/python/examples $
pi@raspberrypi:~/      /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: The path of the samples is based on the actual directory;

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

Demo code

- Demo code
- Github

Datasheets

- SP3481_SP3485.pdf
- SC16IS752_datasheet.pdf