

RS485 RS232 HAT

From Waveshare Wiki

Jump to: navigation, search

Instruction

Product Description

RS485 RS232 HAT is an isolated RS485 and RS232 expansion board specially designed by Waveshare for Raspberry Pi. It adopts SC16IS752+SP3485, SP3232 scheme, built-in power isolation, ADI magnetic coupling isolation and TVS and other protection circuits.

Feature

- Based on Raspberry Pi 40pin GPIO interface, suitable for Raspberry Pi series boards.
- Using SC16IS752 and SP3485, SP3232 multi-chip combination scheme, can realize SPI to RS485, RS232, the communication rate is up to 921600bps.
- RS485 supports manual or automatic sending and receiving of data, which can be set by DIP switch.
- RS485 onboard resettable fuse and protection diode can ensure stable output of current and voltage, prevent overcurrent and overvoltage, and improve shock resistance.
- Onboard TVS (transient voltage suppression tube), which can effectively suppress the surge voltage and transient peak voltage in the circuit, and protect against lightning and static electricity.
- Onboard power supply and serial port transceiver indicators, easy to check the module power supply and communication status.
- Lead out the SPI control interface for easy access to main control boards such as Arduino.
- Provide complete supporting information manuals (C and python sample programs and user manuals, etc.).

Product parameters

- Serial port expansion chip: SC16IS752
- RS485 transceiver: SP3485
- RS232 transceiver: SP3232
- Communication Interface: SPI
- Communication rate: 300~921600 bps

RS485 RS232 HAT



(<https://www.waveshare.com/rs485-rs232-hat.htm>)

- Operating Voltage: 3.3/5V
- Dimension: 65mm × 56.5mm
- Via diameter: 3.0mm

Interface Description

- Pin function

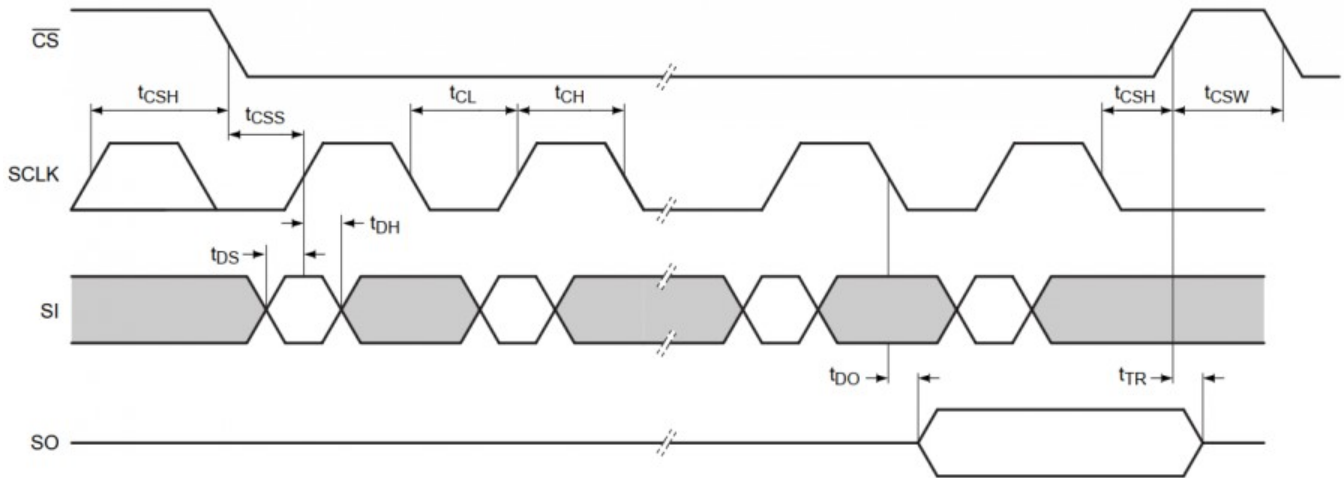
Pin Number	Identification	Pin Description
1	VCC	3.3V/5V Power
2	GND	GND
3	SCLK	SPI clock input
4	MOSI	SPI data input
5	MISO	SPI data output
6	CS	SPI Chip select signal
7	IRQ	interrupt output
8	EN	RS485 output enable

Hardware Description

Controller

This product uses SC16IS752 as the control chip. SC16IS752 is a dual-channel high-performance UART expansion chip that supports SPI and I2C interface communication. This module uses the SPI interface. Onboard power isolation, ADI magnetic coupling isolation, onboard TVS (transient voltage suppression tube), self-recovery fuse and protection diode and automatic transceiver conversion circuit. It can effectively suppress the surge voltage and transient peak voltage in the circuit, prevent lightning and static electricity, prevent overcurrent and overvoltage, improve the anti-shock ability, and can carry out signal isolation. It has the advantages of high reliability, strong anti-interference, and low power consumption.

Letter of Agreement



(/wiki/File:RS485_HAT.png)

CS: the slave chip selection, when CS is low level, the chip is enabled.

SCLK: SPI communication clock.

MOSI/SI: SPI communication host sends, the slave receives.

MISO/SO: SPI communication host receives, the slave sends.

Timing: CPHL=0, CPOL=0 (SPI0)

[Remarks] For specific information about SPI communication, you can search for information online.

Hardware connection

Detailed connection:

The C and python demos need an additional RS485 to serial port module connected to channel 1 to see the effect. test.py in Python requires channel 1 and channel 2 to be connected.

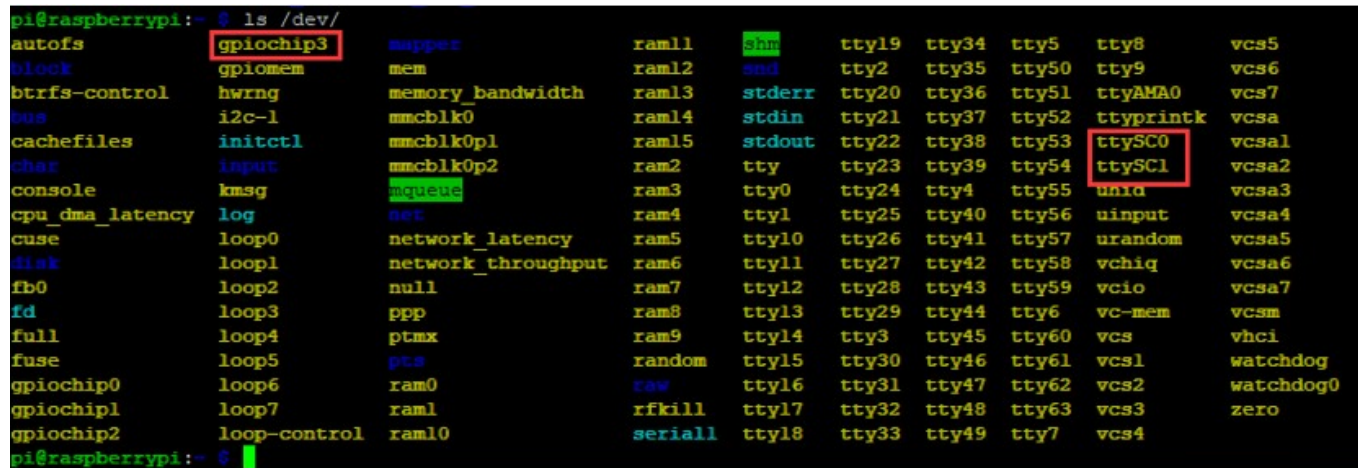
Function Pins	Raspberry Pi Interface (BCM)	Description
VCC	5V	3.3V/5V power supply positive
GND	GND	Power Ground
SCLK	D21(SPI1 SCLK)	SPI clock input
MOSI	D20(SPI1 MOSI)	SPI data input
MISO	D19(SPI1 MISO)	SPI data output
CS	D18(SPI1 CS)	SPI Chip Select Signal
IRQ	D24	Interrupt output, can be switched to D16, D12, D25 pins
EN	D22	RS485 TX/RX enable: high level for TX, low level for RX

Add driver

- Execute in terminal:

```
sudo nano /boot/config.txt
#Add the following, int_pin is set according to the actual welding method:
dtoverlay=sc16is752-spi1,int_pin=24
#reboot device
sudo reboot
```

After restarting, the driver of SC16IS752 will be loaded into the system kernel. At this time, you can run `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            queuefs         ram3            tty0            tty24           tty4           tty55           uinput          vcsa3
cpu_dma_latency log              net             ram4            tty1            tty25           tty40           tty56           urandom         vcsa4
cuse            loop0           network_latency ram5            tty10           tty26           tty41           tty57           vchiq           vcsa5
disk            loop1           network_throughput ram6            tty11           tty27           tty42           tty58           vcio            vcsa6
fb0             loop2           null            ram7            tty12           tty28           tty43           tty59           vc-mem          vcsa7
fd              loop3           ppp             ram8            tty13           tty29           tty44           tty6            vcsa            vcsa8
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           seriall         tty18           tty33           tty49           tty7            vcs4
```

(/wiki/File:Rs485.png)

In the Raspberry Pi system on 2020-05-27, gpiochip3 has not become gpiochip2

Install the library

PS: If you are using the system of the Bullseye branch, you need to change "apt-get" to "apt", the system of the Bullseye branch only supports Python3, and there is no wiringPi.

- Install BCM:

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.71.tar.gz
tar zxvf bcm2835-1.71.tar.gz
cd bcm2835-1.71/
sudo ./configure && sudo make && sudo make check && sudo make install
# For more information, please refer to the official website: http://www.airspayce.com/mikem/bcm2835/
```

- Install wiringPi:

```
sudo apt-get install wiringpi
#For Raspberry Pi 4B may need to be upgraded;
cd /tmp
wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
gpio -v
# Run gpio -v and version 2.52 will appear. If it does not appear, the installation is wrong
```

- Install python2 library:

```
sudo apt-get update
sudo apt-get install python-pip
sudo pip install RPi.GPIO
sudo apt-get install python-serial
```

- Install python3 library:

```
sudo apt-get update
sudo apt-get install python3-pip
sudo pip3 install RPi.GPIO
sudo apt-get install python3-serial
```

Testing

- Download and run the test demo:

```
sudo apt-get install p7zip-full
wget https://files.waveshare.com/upload/4/44/RS485_RS232_HAT.zip
7z x RS485_RS232_HAT.zip
sudo chmod 777 -R RS485_RS232_HAT
cd RS485_RS232_HAT/
```

- C demo:

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

- PYTHON demo:

```
cd python
cd examples
#Use RS485 for communication
sudo python3 RS485.py
#Use RS232 for communication
sudo python3 RS232.py
#Use RS232 TO RS485 to make the modules communicate with each other
sudo python3 test.py
```

Resource

Document

- Schematic (https://files.waveshare.com/upload/3/39/RS485_RS232_HAT_SchDoc.pdf)

Demo

- Sample Demo (https://files.waveshare.com/upload/4/44/RS485_RS232_HAT.zip)

Datasheet

- SP3485 (https://files.waveshare.com/upload/3/36/SP3481_SP3485.pdf)
- SC16IS752 (https://files.waveshare.com/upload/a/ad/SC16IS752_datasheet.pdf)

FAQ

Question: The 485 Communication is abnormal, what should I do?

Answer:

1. Determine whether A and B of 485 correspond to the controlled 485 devices one by one;
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? How to configure the config.txt file?

Answer:

1. The Raspberry Pi is installed with the mainstream Ubuntu system and can be used;
2. You cannot directly change the config.txt file in the Raspberry Pi. When you open the

boot folder on the Raspberry Pi, the config.txt is an empty file, which generally cannot be changed;

3. Instead, you need to 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 the card reader.

Support

Technical Support

If you need technical support or have any feedback/review, please click the **Submit Now** button to submit a ticket, Our support team will check and reply to you within 1 to 2 working days. Please be patient as we make every effort to help you to resolve the issue.

Working Time: 9 AM - 6 PM GMT+8
(Monday to Friday)

Submit Now (<https://service.waveshare.com/>)

Retrieved from "https://www.waveshare.com/w/index.php?title=RS485_RS232_HAT&oldid=84536 (https://www.waveshare.com/w/index.php?title=RS485_RS232_HAT&oldid=84536)"
