

## Contents

### 1 RS485 CAN HAT

#### 1.1 Features

#### 1.2 Specifications

### 2 Pinout

### 3 Installation

#### 3.1 Python

##### 3.1.1 Library Installation

##### 3.1.2 **CAN TEST**

###### 3.1.2.1 Required Hardware

###### 3.1.2.2 Setup

###### 3.1.2.3 Code Test

##### 3.1.3 **RS485 TEST**

###### 3.1.3.1 Required Hardware

###### 3.1.3.2 Code Test

### 4 Resources

## RS485 CAN HAT

---

RS485 CAN HAT is a multi node communication module, to communicate with devices in a long-range, designed dedicatedly for Raspberry Pi with an operating voltage of 3.3 V, CAN controller MCP2515 via SPI interface, onboard transceiver SN65HVD230 that is controlled via UART. It is built to support automatic TX/RX control without programming. It performs half-duplex communication and it has an onboard transceiver SP3485.

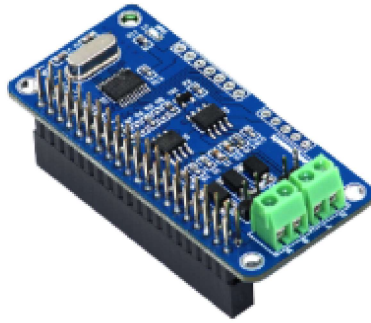
### Features

- Onboard TVS (Transient Voltage Suppressor), lightning-proof & anti-electrostatic
- Standard Raspberry Pi 40 Pins GPIO Extension Headers
- Half-duplex Communication
- Long-Distance Communication via RS485/CAN functions

### Specifications

- Operating voltage: 3.3V
- CAN transceiver: SN65HVD230
- RS485 transceiver: SP3485

- CAN controller: MCP2515
- Communication Protocol - RS485 and CAN



(/File:RS485-CAN-HAT.png)

Buy it From : *Click Here* (<https://shop.sb-components.co.uk/products/rs485-can-hat-for-raspberry-pi>)

## Pinout

---

Board PIN	Raspberry Pi GPIO (BCM)	DESCRIPTION
3V3	3V3	3.3V Power
GND	GND	Ground
SCK	GPIO 11	SPI clock input
MOSI	GPIO 10	SPI data input
MISO	GPIO 9	SPI data output
CS	GPIO 8	SPI chip selection
INT	GPIO 25	Interrupt
RXD	GPIO 15	RS485 UART receive
TXD	GPIO 14	RS485 UART transmit
RSE	GPIO 4	RS485 TX/RX control Line (Currently in auto mode)

# Installation

---

## Python

### Library Installation

Before running this board you have install required libraries for RS485 CAN HAT.

- For Python3, install these two libraries by running below command on the terminal.

```
sudo apt-get install python-pip
sudo pip3 install python-can
```

## CAN TEST

### Required Hardware

- Raspberry Pi x 2
- RS485 CAN HAT x 2

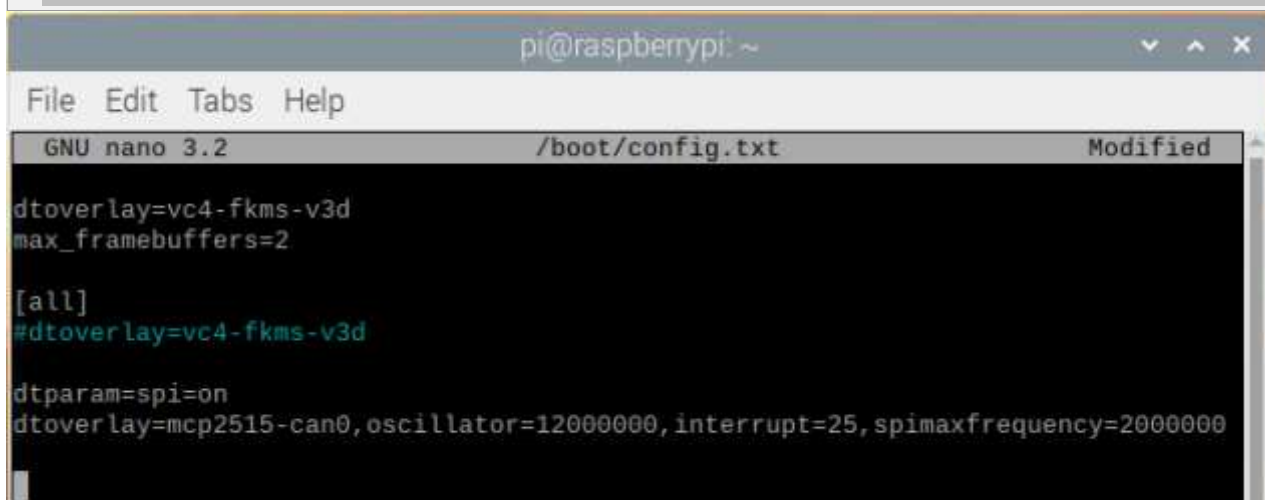
### Setup

- Stack RS485 CAN HAT on GPIO header of both Raspberry Pi, and modify the boot/config.txt file.

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

- Add below line to the end of the file.

```
dtparam=spi=on
dtoverlay=mcp2515-can0,oscillator=12000000,interrupt=25,spimaxfrequency=2000000
```



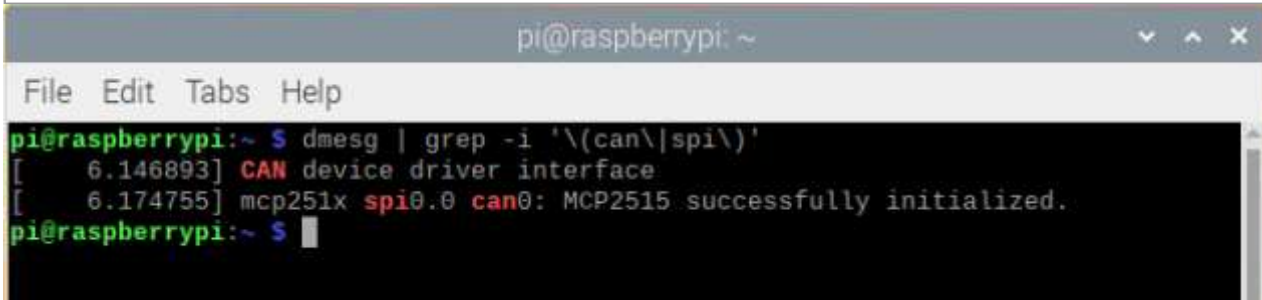
(/File:485-CAN-HAT-config.PNG)

- Save this file on both boards and reboot both raspberry pi one by one.

```
sudo reboot
```

- After restart, run the below command to verify the installation by running the below command (Make sure RS485 CAN HAT is connected while booting).

```
dmesg | grep -i '\(can\|spi\|'\)
```



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ dmesg | grep -i '\(can\|spi\|'  
[ 6.146893] CAN device driver interface  
[ 6.174755] mcp251x spi0.0 can0: MCP2515 successfully initialized.  
pi@raspberrypi:~$
```

(/File:485-CAN-HAT-install-check.PNG)

## Code Test

- Now connect H and L Terminal of the first HAT to H and L terminal of the second HAT.
- Clone or download Github repository by running below command.

```
git clone https://github.com/sbcshop/RS485-CAN-HAT.git
```

- Now enter cloned repository by running below command one by one.

```
cd RS485-CAN-HAT  
cd CAN
```

- Now run code on receiver first.

```
sudo python3 can_receive.py
```

- Now run code on Sender.

```
sudo python3 can_send.py
```

You will get some value with timestamp and id on the receiver terminal.

## RS485 TEST

### Required Hardware

- Raspberry Pi x 2
- RS485 CAN HAT x 2

### Code Test

- Stack RS485 CAN HAT on GPIO header of both Raspberry Pi.
- Now connect A and B Terminal of the first HAT to A and B terminal of the second HAT.
- Clone or download Github repository by running below command.

```
git clone https://github.com/sbcshop/RS485-CAN-HAT.git
```

- Now enter cloned repository by running below command one by one.

```
cd RS485-CAN-HAT  
cd RS485
```

- Now run code on receiver first.

```
sudo python3 485_receive.py
```

- Now run code on Sender.

```
sudo python3 485_send.py
```

You can start getting data on the receiver terminal after the above steps.

## Resources

Retrieved from "<https://learn.sb-components.co.uk/index.php?title=RS485-CAN-HAT&oldid=468>  
(<https://learn.sb-components.co.uk/index.php?title=RS485-CAN-HAT&oldid=468>)"

This page was last edited on 8 June 2021, at 06:19.

[Privacy policy \(/SB-Components\\_Wiki:Privacy\\_policy\)](#)  
(/SB-Components\_Wiki:General\_disclaimer)

[About SB-Components Wiki \(/SB-Components\\_Wiki:About\)](#)

[Disclaimers](#)