RPi Zero Relay

Industrial 6-Channel Relay Module For Raspberry Pi Zero, RS485/CAN Bus, Power Supply Isolation, Photocoupler Isolation

Features

- RS485 half-duplex communication: using SP3485, UART control, auto RX/TX switch
- CAN half-duplex communication: using MCP2515 + SN65HVD230 solution, SPI control
- Onboard unibody power supply isolation, provides stable isolated voltage, needs no extra power supply for the isolated terminal
- Onboard photocoupler isolation, prevent interference from external high-voltage circuit connected to the relay
- Onboard TVS (Transient Voltage Suppressor), effectively suppress surge voltage and transient spike voltage in the circuit, lightningproof & anti-electrostatic
- Onboard resettable fuses and protection diodes, ensuring current/voltage stable output, preventing over current/voltage, better shock-resistance performance
- High quality relay, contact rating: ≤10A 250V AC or ≤10A 30V DC
- ABS protection enclosure with rail-mount support, easy to install, safe to use

Specifications

- Operating voltage: 7V~36V (industrial input voltage compatible)
- Relay channel: 6ch
- Communication Protocol: RS485,CAN
- Contact form: 1NO 1NC

Pinouts

- CAN bus

PIN	Raspberry Pi (BCM)	Description
GND	GND	Ground
SCK	SCK	SPI clock input
MOSI	MOSI	SPI data input

MISO	MISP	SPI Data output
CS	CE0	Data/Command Selection
INT	25	Interrupt output

- RS485 bus

PIN	Raspberry Pi (BCM)	Description
GND	GND	Ground
RXD	RXD	UART receive
TXD	TXD	UART transimit

- Relay interfaces

PIN	Raspberry Pi (BCM)	Description	GND!	GND	Ground
IN_CH1	GPIO5	Relay Channel 1			
IN_CH2	GPIO6	Relay Channel 2			
IN_CH3	GPIO13	Relay Channel 3			
IN_CH4	GPIO16	Relay Channel 4			
IN_CH5	GPIO19	Relay Channel 5			
IN_CH6	GPIO20	Relay Channel 6			

Install libraries

- BCM2835 libraries

```
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.68.tar.gz
tar zxvf bcm2835-1.68.tar.gz
cd bcm2835-1.68/
sudo ./configure && sudo make && sudo make check && sudo make install
```

- Install wringPi

```
sudo apt-get install wiringpi
wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
gpio -v
```

- Python

```
sudo apt-get update
sudo apt-get install python-serial
sudo pip install python-can
```

Download Examples

Open a terminal and run the following commands to download the demo codes

```
sudo apt-get install p7zip-full
wget https://www.waveshare.com/w/upload/2/2f/RPi_Zero_Relay_Code.7z
7z x RPi_Zero_Relay_Code.7z  -r -o./RPi_Zero_Relay_Code
sudo chmod 777 -R RPi_Zero_Relay_Code/
```

CAN Bus

Configuration

- Modify the config.txt file

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

Add the fololwing lines to the config.txt file and save

```
dtparam=spi=on
```

```
dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=25,spimaxfrequency=3000000
```

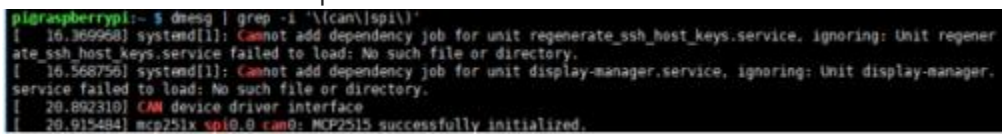
- Reboot

```
sudo reboot
```

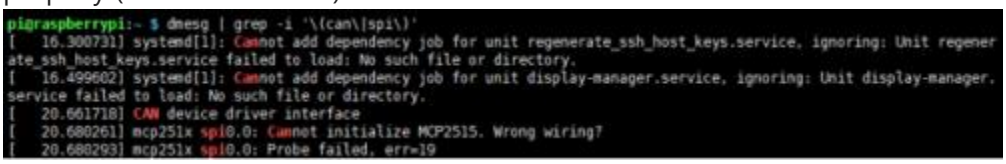
- Check if the driver is initalized normally

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

The result should like the picture:



If the MCP2515 driver is not initalized normall you need to reboot and check if the devices are connected properly (H to H and L to L).

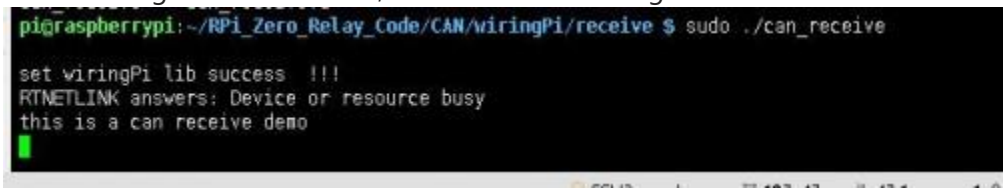


C examples

- Run the receive example

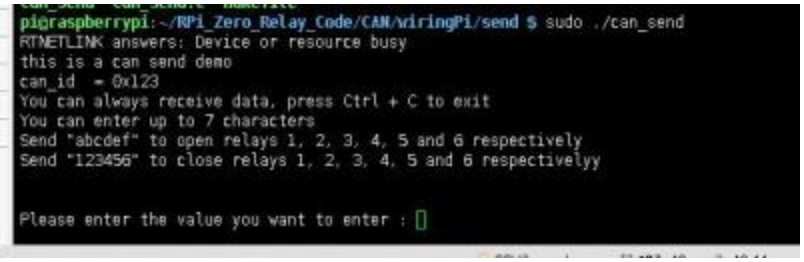
```
cd ~/RPi_Zero_Relay_Code/CAN/wiringPi/receive/
make clean
make
sudo ./can_receive
```

After running the receive code, the terminal is waiting for data

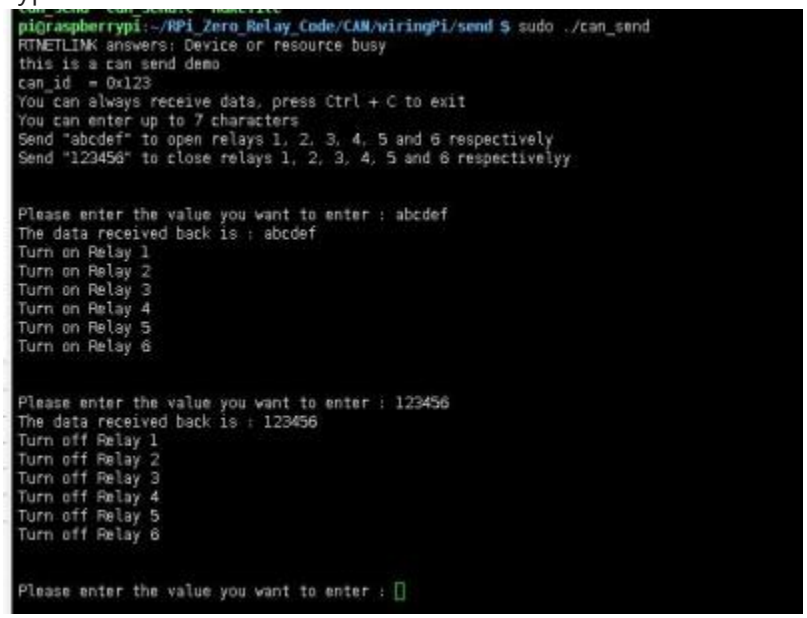


- Run the Send examples:

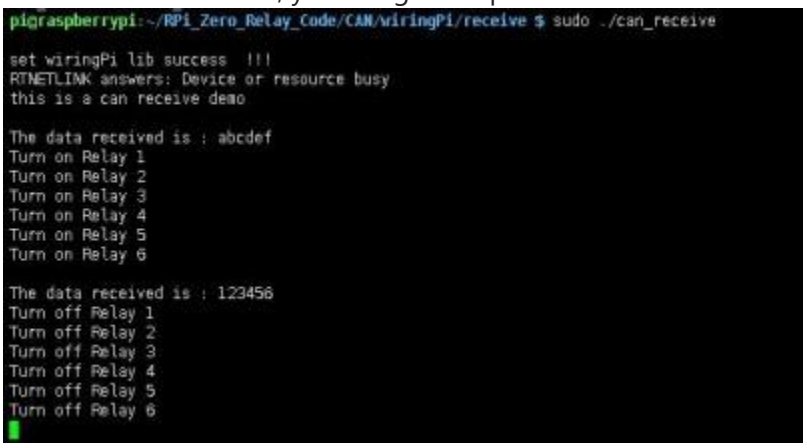
```
cd RPi_Zero_Relay_Code/CAN/wiringPi/send/
make clean
make
sudo ./can_send
```



Type data to send



In the receive terminal, you will get the packet related to id.



Python examples

Open a terminal and run the commands

```
cd RPi_Zero_Relay_Code/CAN/python/
#For receiving:
sudo python can_reveive.py
#For sending:
sudo python can_send.py
```

Connect to other CAN device

If you need to connect the RPi Zero Relay to other CAN devices, please note that:

- Please check the connection, it should be H to H and L to L
- Make sure that the baud rate of both sides are the same, the default baud rate of examples is 100K
- Check if the CAN ID of both sides are the same
- If data is always lost while transmitting, please decrease the baud rate and test it again.

RS485 Interface

Configuration

Open a terminal and run the command

```
sudo raspi-config
```

Choose Interfacing Options -> Serial -> No -> Yes

Disable the login shell function and enable the hardware serial port.

Reboot

```
sudo reboot
```

- Connect the device, A to A and B to B

C examples

- Run the receive example

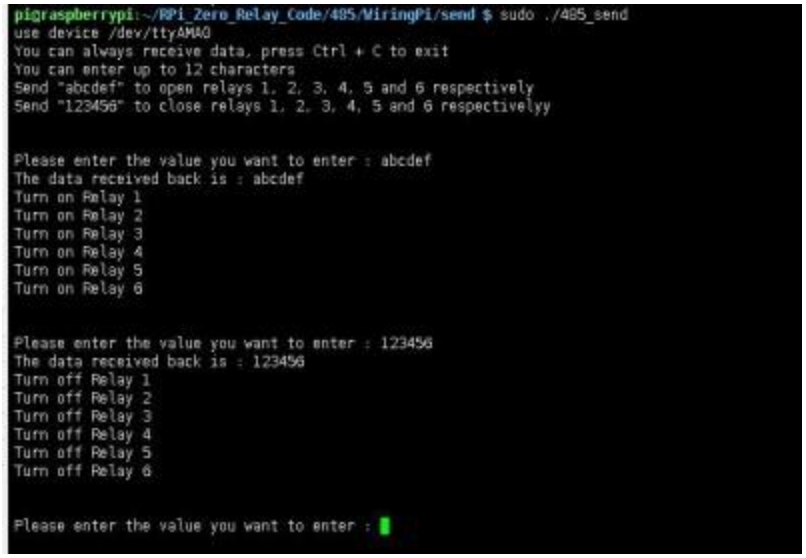
```
cd RPi_Zero_Relay_Code/485/WiringPi/send
make clean
make
sudo ./485_receive
```

After running the receive code, the terminal is waiting for data

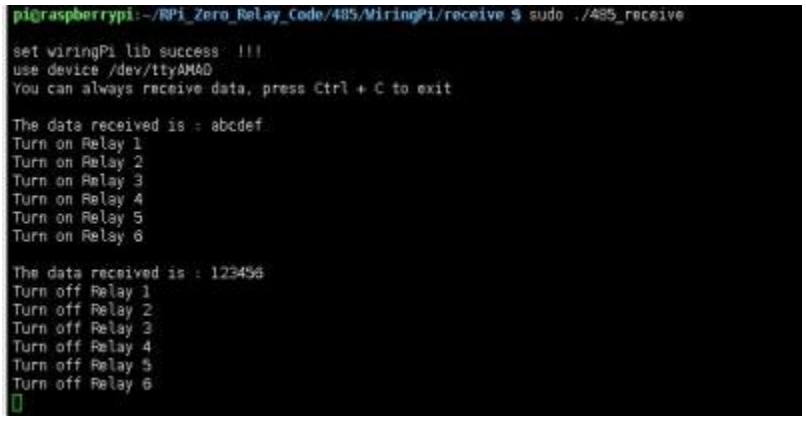
Run the Send example

```
cd RPi_Zero_Relay_Code/485/WiringPi/send
make clean
make
sudo ./485_send
```

- Input data and send:



- The receive terminal recieve and print:



Python examples

Open a terminal and run the commands:

```
cd RPi_Zero_Relay_Code/485/python/
#Receive example：
sudo python receive.py
#Send example：
sudo python send.py
```

Troubleshooting

If the RS485 cannot work normally, please check:

Please check if you have disabled the login shell function

Please check the connection, it should be A to A and B to B

Please test with USB to RS485, make sure that all the settings are correct.

- Schematic
- Demo codes
- SN65HVD230 Datasheet
- MCP2515 Datasheet
- SP348 Datasheet