Waveshare Modbus RTU Relay

Waveshare Industrial Modbus RTU 8-Channel Relay Module User Manual

Model: Modbus RTU Relay

1. Introduction

This manual provides detailed instructions for the Waveshare Industrial Modbus RTU 8-Channel Relay Module. This module is designed for industrial control applications, offering reliable relay switching via an RS485 interface using the Modbus RTU protocol. It features multiple isolation protection circuits for enhanced safety and stability.

Key features include:

- Configurable device address (1-255) for multi-device cascading on an RS485 bus.
- Flash-on, flash-off function for timed relay activation.
- Onboard unibody power supply isolation for stable isolated voltage.
- Onboard unibody magnetic isolation for signal integrity, high reliability, and strong anti-interference.
- Onboard TVS (Transient Voltage Suppressor) for surge and transient spike voltage suppression, providing lightning protection and anti-electrostatic capabilities.
- Onboard optocoupler isolation to prevent interference from external high-voltage circuits.
- Reverse-proof circuit to protect against accidental damage from incorrect power connection.
- High-quality relays with a contact rating of ≤10A 250VAC/30VDC.
- Rail-mounted ABS plastic enclosure for easy installation and safe use.
- Three LEDs indicating MCU status and signal transceiving status.

2. Package Contents

The standard package for the Waveshare Industrial Modbus RTU 8-Channel Relay Module includes the

following items:

- Waveshare Modbus RTU 8-Channel Relay Module
- Power adapter (may vary by region)

Image 2.1: The image displays the Waveshare Modbus RTU 8-Channel Relay Module along with a power adapter and a USB to RS485 converter (converter not included with the module).

3. Hardware Overview

The Modbus RTU 8-Channel Relay Module is designed with robust components and protection circuits. Understanding its internal structure is crucial for proper use and troubleshooting.

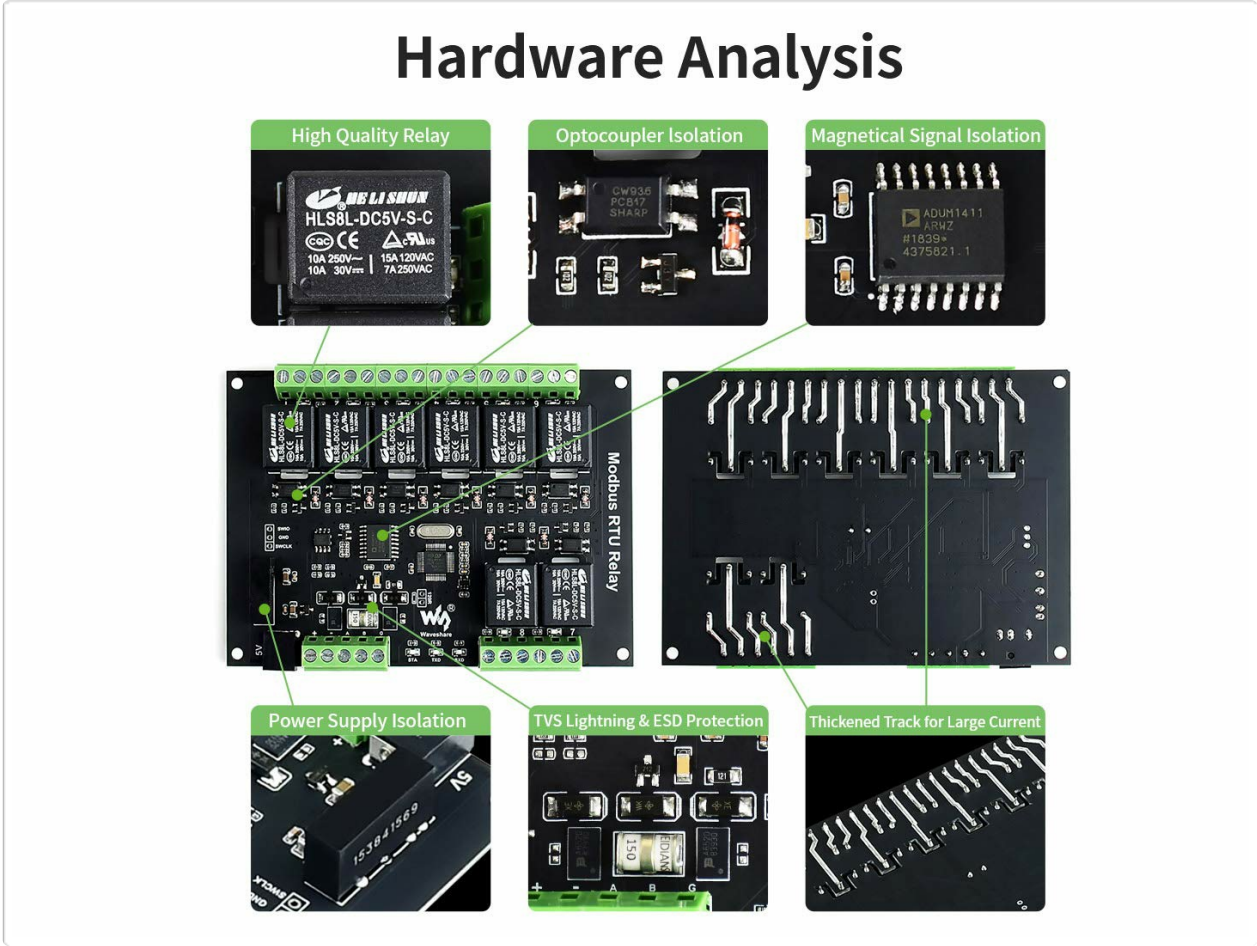


Image 3.1: This image provides a detailed view of the module's internal components, highlighting the high-quality relays, optocoupler isolation, magnetic signal isolation, power supply isolation, TVS lightning & ESD protection, and thickened tracks for large current.

3.1. Key Components

- **High Quality Relay:** Ensures reliable switching for connected devices.
- **Optocoupler Isolation:** Provides electrical isolation between control and load circuits.
- **Magnetic Signal Isolation:** Enhances signal integrity and anti-interference capabilities.
- **Power Supply Isolation:** Isolates the power supply for stability and safety.
- **TVS Lightning & ESD Protection:** Protects against voltage surges and electrostatic discharge.
- **Thickened Track for Large Current:** Designed to handle higher current loads safely.

3.2. Enclosure Design

The module is housed in a rail-mounted ABS plastic enclosure, facilitating easy installation in industrial environments and ensuring safe operation.

Image 3.2: The image shows the module's rail-mounted ABS plastic enclosure, designed for easy and secure installation.

4. Setup

Follow these steps to set up your Waveshare Industrial Modbus RTU 8-Channel Relay Module.

4.1. Power Connection

1. Connect the provided power adapter to the DC 5V input terminal of the relay module. Ensure the polarity is correct to prevent damage.
2. Verify that the power LED indicator on the module illuminates, indicating successful power-on.

4.2. RS485 Communication Setup

The module communicates via RS485. Multiple devices can be cascaded on the RS485 bus. If communication distance is long, RS485 repeaters may be necessary.

1. Connect the RS485 A+, B-, and GND terminals of the module to your RS485 master device (e.g., a computer with a USB to RS485 converter, or a PLC).
2. Ensure correct wiring: A+ to A+, B- to B-, and GND to GND.
3. For multiple modules, connect them in a daisy-chain configuration as shown in the diagram below.

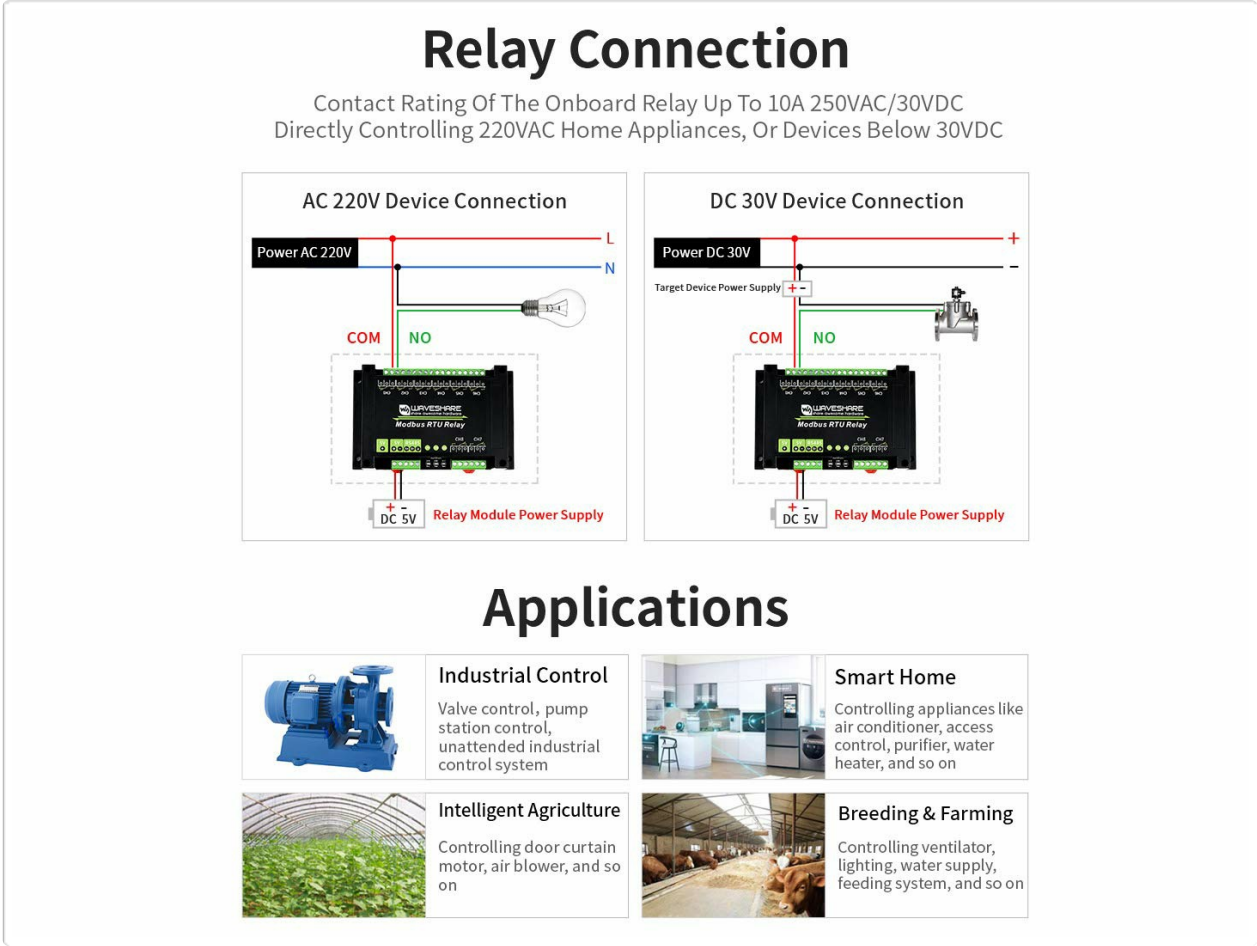


Image 4.1: This diagram illustrates how to connect multiple Modbus RTU relay modules in a daisy-chain configuration using an RS485 bus, typically connected to a USB to RS485 converter.

4.3. Device Address Configuration

Each module on the RS485 bus must have a unique device address, configurable from 1 to 255. Refer to the Modbus RTU protocol documentation for your specific software or controller to set the device address.

5. Operating Instructions

The Waveshare Modbus RTU 8-Channel Relay Module operates using the standard Modbus RTU protocol over an RS485 interface. This section outlines how to control the relays.

5.1. Modbus RTU Protocol

The module responds to standard Modbus RTU commands. You will typically use Modbus function codes to read coil status, write single coils, or write multiple coils to control the relays. Consult the Modbus RTU specification for detailed command structures.

5.2. Relay Control

Each of the 8 channels corresponds to a relay. Relays can be individually controlled (turned ON/OFF) using Modbus commands. The module also supports a flash-on, flash-off function, allowing you to activate a relay for a specified duration and then automatically deactivate it.

5.3. LED Indicators

- **STA (Status) LED:** Keeps flashing when the MCU is working normally.
- **TXD (Transmit) LED:** Lights up when the module is sending data.
- **RXD (Receive) LED:** Lights up when the module is receiving data.

6. Relay Connection

The module's relays have a contact rating of ≤10A 250VAC/30VDC. This allows direct control of various AC and DC devices. Always ensure that the connected load does not exceed the relay's maximum current and voltage ratings.

6.1. AC 220V Device Connection

For connecting AC 220V devices, ensure proper insulation and safety precautions. Connect the AC load to the COM (Common) and NO (Normally Open) terminals of the desired relay channel.

6.2. DC 30V Device Connection

For connecting DC devices up to 30V, connect the DC load to the COM (Common) and NO (Normally Open) terminals of the desired relay channel. Ensure correct polarity for DC loads.

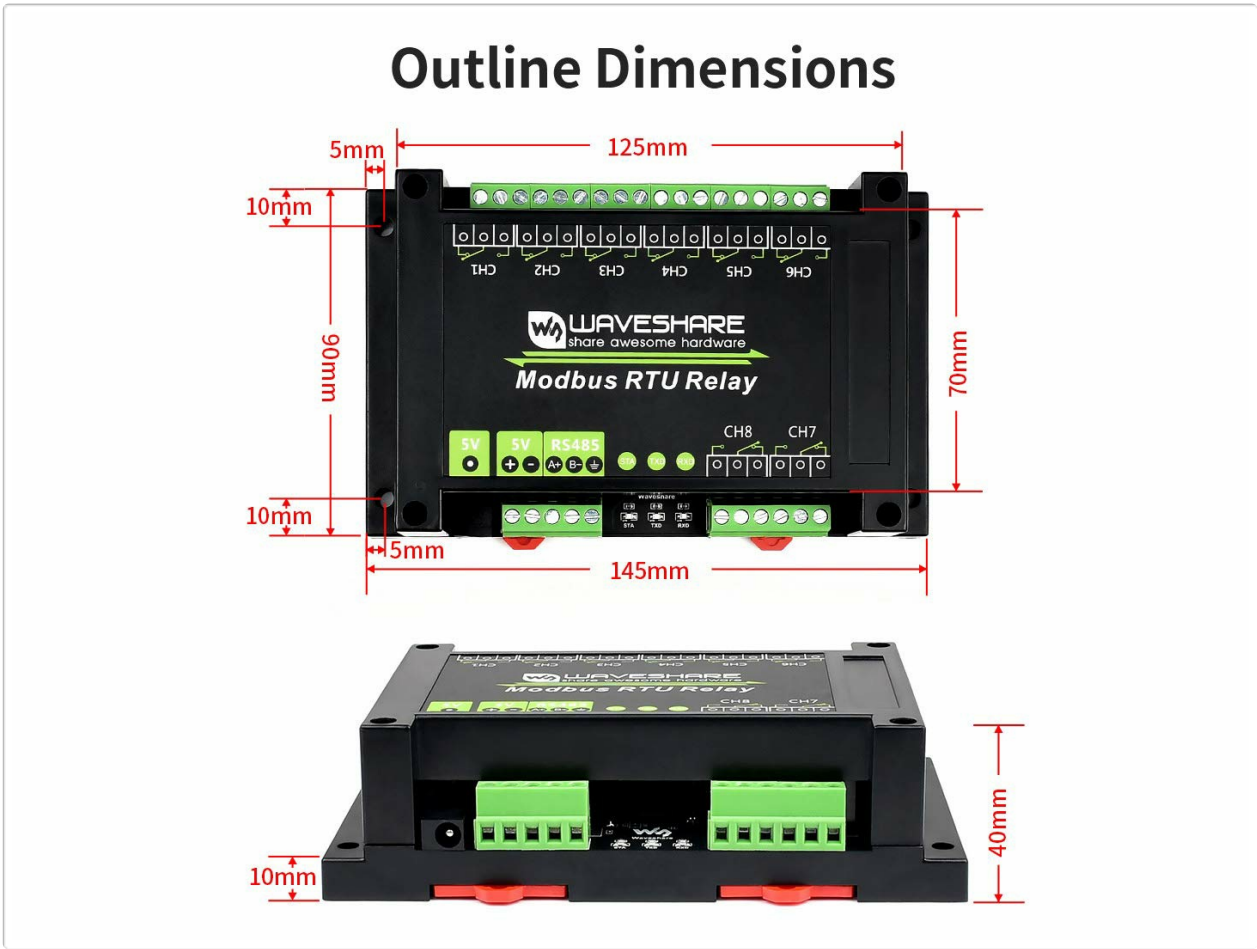


Image 6.1: This diagram illustrates how to connect both AC 220V and DC 30V devices to the relay module. It shows wiring for a light bulb (AC) and a motor (DC) using the COM and NO terminals, with the module powered by DC 5V.

7. Applications

The Waveshare Industrial Modbus RTU 8-Channel Relay Module is versatile and suitable for a wide range of applications:

- **Industrial Control:** Valve control, pump station control, and unattended industrial control systems.
- **Smart Home:** Controlling appliances like air conditioners, access control systems, purifiers, and water heaters.
- **Intelligent Agriculture:** Controlling door curtain motors, air blowers, and other agricultural equipment.
- **Breeding & Farming:** Controlling ventilators, lighting, water supply systems, and feeding systems.

Specifications

PRODUCT	Modbus RTU Relay	Modbus RTU Relay (B)
POWER SUPPLY	5V	7~36V
COMMUNICATION INTERFACE	RS485	
BAUDRATE	4800, 9600, 19200, 38400, 57600, 115200, 128000, 256000	
DEFAULT COMMUNICATION FORMAT	9600, N, 8, 1	
RELAY CHANNELS	8	
CONTACT FORM	1NO 1NC	
CONTACT RATING	≤10A 250VAC/30VDC	
COMMUNICATION PROTOCOL	Standard Modbus RTU protocol	
RS485 ADDRESS	1~255	
LED INDICATORS	STA: MCU indicator, keeps flashing when the MCU normally working TXD: TX indicator, lights up when sending data RXD: RX indicator, lights up when receiving data	

Image 7.1: This image displays four common application areas for the relay module: Industrial Control, Smart Home, Intelligent Agriculture, and Breeding & Farming, showing examples of equipment that can be controlled.

8. Specifications

Detailed technical specifications for the Modbus RTU Relay Module:

Feature	Modbus RTU Relay	Modbus RTU Relay (B)
Power Supply	5V	7-36V
Communication Interface	RS485	
Baudrate	4800, 9600, 19200, 38400, 57600, 115200, 128000, 256000	
Default Communication Format	9600, N, 8, 1	
Relay Channels	8	
Contact Form	1NO 1NC	
Contact Rating	≤10A 250VAC/30VDC	
Communication Protocol	Standard Modbus RTU protocol	
RS485 Address	1-255	
LED Indicators	STA: MCU indicator, keeps flashing when the MCU normally working TXD: TX indicator, lights up when sending data RXD: RX indicator, lights up when receiving data	

9. Maintenance

The Waveshare Industrial Modbus RTU 8-Channel Relay Module is designed for robust and reliable operation with minimal maintenance. To ensure longevity and optimal performance:

- **Keep Clean:** Periodically clean the exterior of the module with a soft, dry cloth. Avoid using harsh chemicals or abrasive materials.
- **Environmental Conditions:** Operate the module within its specified temperature and humidity ranges. Avoid exposure to excessive dust, moisture, or corrosive environments.
- **Connection Integrity:** Regularly check all wiring connections (power, RS485, and relay loads) to ensure they are secure and free from corrosion or damage.
- **Firmware Updates:** Check the official Waveshare website for any available firmware updates that may improve performance or address issues. Follow update instructions carefully.

10. Troubleshooting

If you encounter issues with your Waveshare Industrial Modbus RTU 8-Channel Relay Module, refer to the following troubleshooting tips:

- **Module Not Powering On:**
  - Ensure the power adapter is correctly connected and providing the specified voltage (5V for this model).
  - Check the power outlet and cable for functionality.
  - Verify the power LED indicator. If it's off, there might be a power issue.
- **No RS485 Communication:**
  - Verify RS485 wiring: A+ to A+, B- to B-, and GND to GND.
  - Check the baud rate, data bits, parity, and stop bits settings in your Modbus master software. These must match the module's default or configured settings (e.g., 9600, N, 8, 1).
  - Ensure the module's RS485 device address is unique and correctly configured in your master software.
  - Observe the TXD and RXD LEDs. If they are not flashing during communication attempts, check wiring and software settings.
  - For long distances or multiple devices, consider adding RS485 terminators or repeaters.
- **Relays Not Activating:**
  - Confirm that the module is powered on and communicating correctly (check communication LEDs).
  - Verify the Modbus commands sent to activate the specific relay channels.
  - Check the wiring of the load connected to the relay. Ensure the load's power supply is active and within the relay's contact rating.
  - Ensure the load itself is functional.
- **Intermittent Operation:**
  - Check for loose connections on power, RS485, and relay terminals.
  - Ensure the operating environment is stable and free from excessive electrical noise or temperature fluctuations.
  - Verify that the power supply is stable and sufficient for both the module and the connected loads.

11. Outline Dimensions

The physical dimensions of the Waveshare Industrial Modbus RTU 8-Channel Relay Module are provided below for installation planning.

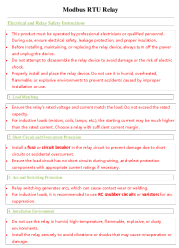
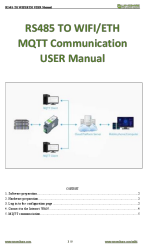
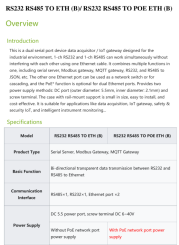
Image 11.1: This image provides detailed measurements of the module, including its length (145mm), width (90mm), and height (40mm), along with other specific dimensions for mounting.

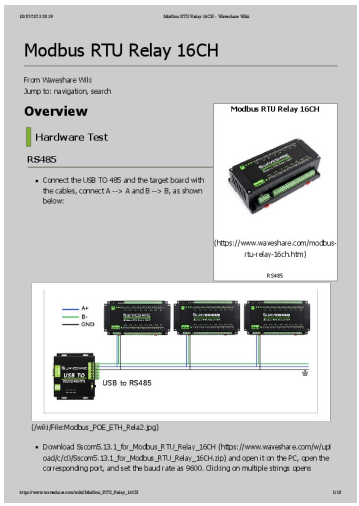
12. Warranty and Support

Waveshare products are designed for quality and reliability. For specific warranty information, please refer to the warranty policy provided with your purchase or visit the official Waveshare website. For technical support, documentation, and additional resources, please visit the Waveshare support portal or contact their customer service directly.

Official Waveshare Website: www.waveshare.com

Related Documents - Modbus RTU Relay

	[Modbus RTU Relay: User Manual and Technical Guide] Comprehensive guide to the Modbus RTU Relay, covering safety instructions, hardware connection, software setup with SSCOM and Modbus Poll, and detailed command protocols for various operations. Includes examples for Raspberry Pi, STM32, Arduino, and PLC integration.
	[Waveshare Industrial 8-Channel Relay Module for Raspberry Pi Pico User Manual] User manual for the Waveshare Industrial 8-Channel Relay Module for Raspberry Pi Pico (Pico-Relay-B). Details features, compatibility, enclosure, and pinout for industrial control applications.
	[Waveshare WS-TTL-CAN User Manual: TTL to CAN Converter Guide] Explore the Waveshare WS-TTL-CAN module with this comprehensive user manual. Learn about its TTL and CAN communication capabilities, hardware features, parameter configuration using WS-CAN-TOOL, and various conversion examples.
	[Waveshare RS485 to WiFi/ETH MQTT Communication User Manual] Comprehensive user manual for the Waveshare RS485 to WiFi/ETH module, guiding users through software and hardware preparation, network configuration, and establishing MQTT communication with platforms like EMQX.
	[Waveshare RS232/485 to Ethernet Converter User Manual] This is an industrial-grade RS232/485 to Ethernet module from Waveshare, enabling bidirectional transparent data transmission between RS232, RS485, and RJ45 Ethernet ports. It features a 32-bit ARM M4 processor, 10/100M Ethernet, multiple operating modes (TCP Server/Client, UDP, HTTPD), Modbus gateway, and Websocket support, offering high speed, low power consumption, and stability for industrial IoT applications.
	[RS232/RS485 to Ethernet & PoE Ethernet Gateway Technical Specification] This document provides detailed specifications, hardware and software features, and testing procedures for the Waveshare RS232/RS485 to Ethernet and PoE Ethernet gateways. These devices function as serial servers, Modbus gateways, and MQTT gateways for industrial data acquisition and IoT connectivity.

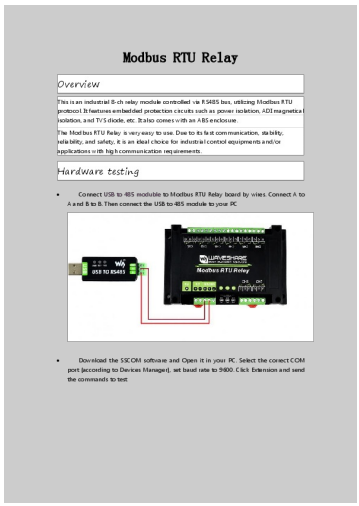


[pdf]

Sparwan Distributor of IoT IT equipment Modbus RTU Relay 16CH Waveshare Wiki v 1718975546 cdn shopify s files 1 0747 1438 2601 |||

18/07/2023 08:39 Modbus RTU Relay 16CH - Waveshare Wiki Modbus RTU Relay 16CH From Waveshare Wiki Jump to: navigation, search Overview Hardware Test RS485 Connect the USB TO 485 and the target board with the cables, connect A -- A and B -- B, as shown below: Modbus RTU Relay 16CH https://www...

lang:en score:38 filesize: 1.25 M page_count: 18 document date: 2023-07-18

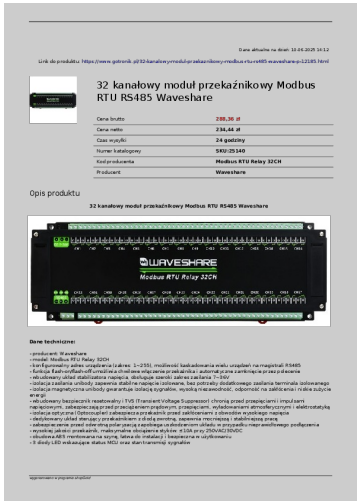


[pdf] User Manual

Administrator Modbus RTU Relay 20210526021836ModbusRTURelay myosuploads3 banggood products 20210526

Modbus RTU Relay Overview This is an industrial 8-ch relay module controlled via RS485 bus, utilizing Modbus RTU protocol. It features embedded protection circuits such as power isolation, ADI magnetical isolation, and TVS diode, etc. It also comes with an ABS enclosure. The Modbus RTU Relay is ver...

lang:en score:33 filesize: 286.44 K page_count: 4 document date: 2021-05-26

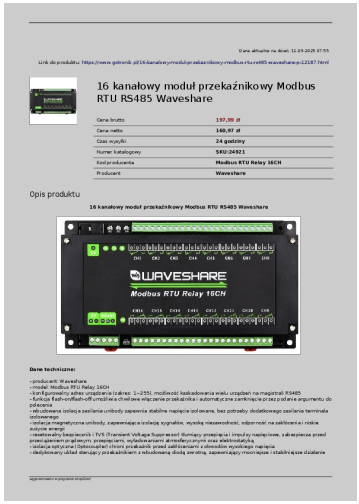


[pdf]

Produkt shopGold 32 kanałowy moduł przekaźnikowy Modbus RTU RS485 Waveshare Apr 27 2025 · producent model Relay 32CH konfigurowalny adres urządzenia zakres 1~255 możliwość kaskadowania wielu kanałowy modul przekaznikowy modbus rtu rs485 waveshare gotronik pl karta 12185 html |||

Dane aktualne na dzie: 10-06-2025 14:12 Link do produktu: https://www.gotronik.pl/32-kanalowy-modul- ... to Czas wysylki Numer katalogowy Kod producenta Producent 288,36 zl 234,44 zl 24 godziny SKU:25140 Modbus RTU Relay 32CH Waveshare Opis produktu 32 kanalowy modul przekanikowy Modbus RTU RS485 Wave...

lang:pl score:26 filesize: 1.73 M page_count: 6 document date: 2025-06-10


16 kanałowy moduł przekaźnikowy Modbus RTU RS485 Waveshare

Opis produktu

[pdf]

Produkt shopGold 16 kanałowy moduł przekaźnikowy Modbus RTU RS485Jun 25 2025 — producent Waveshare model Relay 16CH konfigurowalny adres urządzenia zakres 1~255 możliwość kaskadowania wielu urządzeń na magistrali16 kanalowy modul przekaznikowy modbus rtu rs485 wavesharegotronik pl waveshare karta 12187 html |||

Dane aktualne na dzie: 11-09-2025 07:55 Link do produktu: https://www.gotronik.pl/16-kanalowy-modul- ... to Czas wysylki Numer katalogowy Kod producenta Producent 197,99 zl 160,97 zl 24 godziny SKU:24921 Modbus RTU Relay 16CH Waveshare Opis produktu 16 kanalowy modul przekanikowy Modbus RTU RS485 Wave...

lang:pl score:26 filesize: 1.62 M page_count: 6 document date: 2025-09-11


Przemysłowy 8 kanałowy moduł przekaźnikowy Modbus RTU z RS485 Waveshare

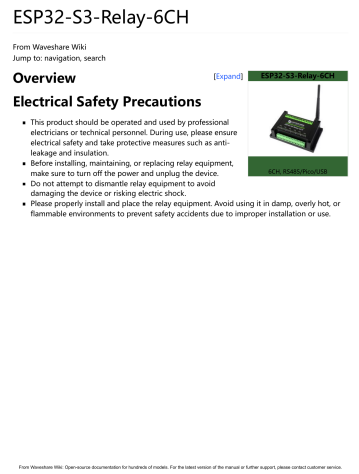
Opis produktu

[pdf]

Produkt shopGold przemyslowy 8 kanalowy modul przekaznikowy modbus rtu z rs485 wavesharegotronik pl waveshare karta 12190 html |||

Dane aktualne na dzie: 28-08-2025 00:51 Link do produktu: https://www.gotronik.pl/przemyslowy-8-kana ... to Czas wysylki Numer katalogowy Kod producenta Producent 127,90 zl 103,98 zl 24 godziny SKU:17658 Modbus RTU Relay Waveshare Opis produktu Przemyslowy 8 kanalowy modul przekanikowy Modbus RTU z RS4...

lang:pl score:26 filesize: 2.14 M page_count: 7 document date: 2025-08-27


ESP32-S3-Relay-6CH

From Waveshare Wiki
Jump to: navigation, search

Overview

Electrical Safety Precautions

- This product should be operated and used by professional electricians or technical personnel. During use, please ensure electrical safety and take protective measures such as anti-leakage and insulation.
- Before installing, maintaining, or replacing relay equipment, make sure to turn off the power and unplug the device.
- Do not attempt to dismantle relay equipment to avoid damaging the device or risking electric shock.
- Please properly install and place the relay equipment. Avoid using it in damp, overly hot, or flammable environments to prevent safety accidents due to improper installation or use.

[pdf] User Manual Installation Guide Instructions Documentation

ESP32 S3 Relay 6CH Waveshare Wiki This example is a collection of RS485 interface control Bluetooth Web page near distance and cloud long distance81jgXbUc2eLm media amazon images I 81jgXbUc2eL ref dp product quick view |||

ESP32-S3-Relay-6CH From Waveshare Wiki Jump to: navigation, search Overview Expand ESP32-S3-Rela ... uick to get started. Components Preparation ESP32-S3-Relay-6CH x 1 USB cable Type-A to Type-C x 1 Modbus RTU Relay x 1 Pico-RTC-DS3231 x 1 USB TO RS232/485 x 1 Pico-2CH-RS232 x 1 or Pico...

lang:en score:20 filesize: 497.79 K page_count: 23 document date: 2025-07-31


RS485 CAN HAT (B)

Overview

RS485 CAN HAT (B) is a HAT with RS485 and CAN communication functions developed by Waveshare for Raspberry Pi, with RS485 and CAN communication functions.

Features

- Standard Raspberry Pi 40PIN GPIO extension header, supports Raspberry Pi series boards
- With 1-ch CAN, the controller MCP2515 and CAN transceiver, convert SPI to CAN
- Adopts SC16IS752 + SP3485 dual-chip combination, converts SPI to RS485, data rate up to 921600bps
- Onboard power conversion circuit, supports 8~28V wide voltage power supply, can supply power to Raspberry Pi at the same time
- Onboard integrated power isolation, provides stable isolation voltage, and no additional power supply is required at the isolation end

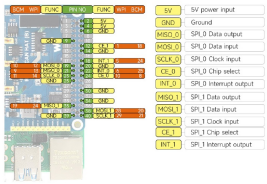
- Onboard integrated digital isolation for signal isolation with high reliability, strong anti-interference, and low power consumption
- Onboard TVS (Transient Voltage Suppressor), effectively suppresses surge voltage and transient spike voltage in the circuit, lightning-proof & anti-electrostatic
- Onboard auto-recovery fuse and protection diodes, ensures the current/voltage stable outputs, provides over-current/over-voltage proof, improve shock resistance
- On-board 120Ω terminal resistor, enabled by jumper cap settings
- It can be wired via onboard terminals or pin headers, making wiring more convenient
- Breakout SPI control pins, for connecting with host control boards

Specification

Parameters

Expanded Interface	2-Ch RS485 + 1-Ch CAN
Communication Bus	SPI
CAN Controller	MCP2515
CAN Receiver	SIT65HVD230DR/SN65HVD23D
CAN Baud Rate	≤1Mbps
UART Expansion Chip	SC16IS752
RS485 Receiver	SP3485
RS485 Baud Rate	300~921600 bps
Power Supply	External screw terminal or Raspberry Pi
External Voltage Range	DC 8~28V
Operating Voltage	3.3V
Dimensions	65 × 56.5 mm

Interface



5V	5V power input
GND	Ground
MISO_0	SPI_0 Data output
MOSI_0	SPI_0 Data input
SCLK_0	SPI_0 Clock input
CE_0	SPI_0 Chip select
INT_0	SPI_0 Interrupt output
MISO_1	SPI_1 Data output
MOSI_1	SPI_1 Data input
SCLK_1	SPI_1 Clock input
CE_1	SPI_1 Chip select
INT_1	SPI_1 Interrupt output

- CAN bus（CAN_0, control via SPI0）

Func	BCM	Description
5V	5V	5V power input
GND	GND	Ground
SCLK_0	11 (SCK)	SPI clock input
MOSI_0	10 (MOSI)	SPI data input
MISO_0	9 (MISO)	SPI data output
CE_0	8 (CE0)[1]	data/command selection
INT_0	D25[2]	interrupt output

- RS485 bus（control RS485_0 & RS485_1 via SPI1）

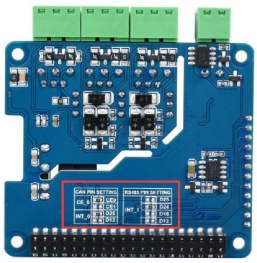
Func	BCM	Description
5V	5V	5V power input
GND	GND	Ground
SCLK_1	D21	SPI clock input
MOSI_1	D20	SPI data input
MISO_1	D19	SPI data output
CE_1	D18	data/command selection
INT_1	D24[3]	interrupt output

Default Solder Joint Switching Description

When the default pin of the board conflicts with other external device pins, you can try to modify the back pin pad, that is, change the 0R resistance of the corresponding collision pin to other non-conflicting pins.

1. ↑ CE_0 soldered to 8 (CE0) by default, you can change it to CE1 by modifying the 0R resistance on the back.
2. ↑ Soldered to D25 by default, you can change it to D13 by modifying the 0R resistance on the back.
3. ↑ Soldered to D24 by default, you can change it to D25/D16/D12 by modifying the 0R resistance on the back.

- The detailed soldering joint on the back is shown below:



Hardware Description

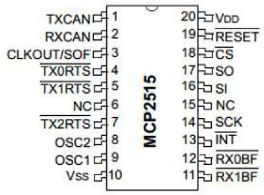
CAN Bus

The CAN module is to process all the reception and transmission of the CAN bus. When sending the report, the report will be loaded in the correct report buffer and control register first. With the SPI interface, you can set the corresponding bit or the enabled pin to send the report. Also, you can check the communication and errors by reading the corresponding register. It will check all the reports in the CAN bus, and then match them with the user-defined filters to make sure the report can be transmitted to one of the receiving buffers.

As the Raspberry Pi cannot support CAN bus, you must use the CAN controller with SPI interface to match with a receiver to finish the CAN function.

Microchip Technology's MCP2515, a CAN protocol controller, totally supports CAN V2.0B specification and can send and receive standard and extended data frames as well as remote frames.

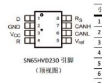
The MCP2515 comes with two acceptance mask registers and six acceptance filter registers that filter out unwanted packets, thus reducing the overhead of the main microcontroller (MCU). The MCU is connected to the device through the SPI interface, that is, the Raspberry Pi connects the chip through the SPI interface, and for the Raspberry Pi to use the chip, it does not need to write a driver, just open the core driver in the device tree to use.



For more details, please refer to the datasheet.

SN65HVD230 is a Texas Instruments 3.3V CAN transceiver suitable for serial communication with high communication rates, good interference immunity, and high-reliability CAN buses.

SN65HVD230 has three different operating modes: high speed, slope, and wait. Its operating mode control is possible via the Rs control pin. The output pin Tx of the CAN controller is connected to the data input D of the SN65HVD230, which can transfer the data sent by this CAN node to the CAN network; The receive pin Rx of the CAN controller is connected to R on the data output of the SN65HVD230 for receiving data.



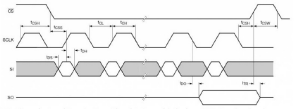
SN65HVD230 引脚
（俯视图）

引脚	名称	说明
1	D	驱动输入 Driver input
2	GND	电源地线 Ground
3	V_{CC}	电源线 Supply voltage
4	R	接收输出 Receiver output
5	V_{ref}	参考输出 Reference output
6	CANL	低总线输出 Low bus output
7	CANH	高总线输出 High bus output
8	RS	工作模式斜制端 Standby/slope control

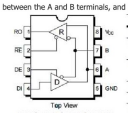
RS485 Bus

This product uses SC16IS752 as the control chip, and SC16IS752 is a dual-channel high-

performance UART expansion chip supporting SPI and I2C communication. This module uses an SPI interface, onboard power isolation, ADI magnetical isolation, TVS (transient voltage suppression transistor), reset-up fuse and protection diode, and automatic transceiver conversion circuitry. It can effectively suppress the surge voltage and transient peak voltage in the circuit, lightning protection and anti-static, anti-overcurrent overvoltage, improve the ability to resist shock, and can carry out signal isolation, which has the advantages of high reliability, strong anti-interference, and low power consumption.
The communication protocol is shown as below:



CS: Slave device chip selection. The chip is enabled when CS is in a low level.
SCLK: SPI communication clock.
MOSI/SI: SPI communication host sends and the slave device receives.
MISO/SO: SPI communication host receives and the slave device sends.
Time order: CPHL = 0, CPOL = 0 (SPI0)
【Note】For more SPI communication information, you can search online.
The SP3485 is a low-power half-duplex transceiver that meets the specifications of RS485 serial protocols. It operates from a single +3.3V supply and uses half-duplex communication. The RO and DI terminals are the output of the receiver and the input of the driver, respectively; (RE)' and DE are the receive and transmit enable sides, respectively, when (RE)' is logic 0, the device is in the receiving state; When DE is logic 1, the device is in the transmit state; The A side and the B side are the receiving and transmitting differential signal terminals, respectively, when the A-B > +0.2V, the RO output logic 1; When A-B <-0.2V, the RO outputs logic 0. A matching resistor is added between the A and B terminals, and a 100Ω resistor is generally optional.

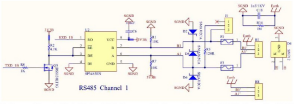


SP485/MAX485/SP3485/MAX3485
引脚（顶视图）

引脚	名称	说明
1	RO	接收器输出 Receiver Output
2	RE	接收输出使能 Receiver Output Enable 低电平有效 Active LOW
3	DE	发送输出使能 Driver Output Enable 高电平有效 Active HIGH
4	DI	输出驱动器输入 Driver Input
5	GND	地 Ground Connection
6	A	差分信号正向端 Driver Output/Receiver Input, Non-inverting
7	B	差分信号反向端 Driver Output/Receiver Input, Inverting
8	V_{CC}	

SP485 / MAX485 是 5V 的 RS485 收发器
SP3485 / MAX3485 是 3.3V 的 RS485 收发器

Among them: the RE and DE pins of the SP3485 chip are set to receive and send;
This module uses the automatic control of hardware to receive and send.



Data receiving: TXD is high at this time, and is in an idle state. At this time, the MOS is turned on, the RE pin of the SP3485 chip is low, the data reception is enabled, and the RO begins to receive data, and the data received by the 485AB port is transmitted to the MCU.
Data transmission: There will be a pull-down level in TXD indicating that data is being sent at the beginning, at which point the MOS cuts off, the DE pin is high, and data transmission is enabled. At this time, if the transmitted data is '1', the MOS will be on, although the reception will become a valid state, but because the chip is in a high-impedance state during the transmission phase, it still maintains the transmit state and transmits '1' normally.

Working With Raspberry Pi

Install Libraries

BCM2835

```
#Open the terminal of the Raspberry Pi and run the following command
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.a
irspayce.com/mikem/bcm2835/
```

WiringPi

```
#Open the terminal of the Raspberry Pi and run the following command
cd
sudo apt-get install wiringpi
#For Raspberry Pi systems after May 2019 (not done before the previous on
e) may require an upgrade:
wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
gpio -v
# Running gpio -v will appear version 2.52, if there is no description ins
tallation error

#The Bullseye branch system uses the following command:
git clone https://github.com/WiringPi/WiringPi
cd WiringPi
./build
gpio -v
# Running gpio -v will appear version 2.60, if there is no description ins
tallation error
```

python

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

Install python3 library:

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

Driver Configuration

Please insert the module into the Raspberry Pi and change the start-up script "config.txt".

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

Add the following content to the last command:

```
dtparam=spi=on
dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=25,,spimaxfrequency=100
0000
dtoverlay=sc16is752-spi1,int_pin=24
```

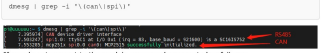
Save and exit, and reboot the Raspberry Pi:

```
sudo reboot
```

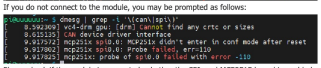
Check Whether The Configuration Is Correct

After rebooting, the drivers of SC16IS752 and mcp251x will be loaded into the system kernel, run the command to check whether the initialization is successful:

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



If you do not connect to the module, you may be prompted as follows:



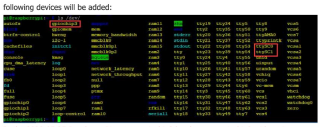
Please check if the module is connected, whether the SPI, and MCP2515 kernel is enabled and whether it is restarted.

RS485

Also, you can run the following commands:

```
ls /dev
```

Check whether the RS485 bus is correctly configured. If the configuration is correct, the following devices will be added:



In the 2020-05-27 Raspberry Pi system gpiochip3 has become gpiochip2 without it.

CAN

Open CAN

```
sudo ip link set can0 up type can bitrate 1000000
```

```
sudo ifconfig can0 txqueuelen 65536
```

- For more CAN kernel commands, please refer to:

```
https://www.kernel.org/doc/Documentation/networking/can.txt
```

- Check ifconfig:

```
ifconfig
```

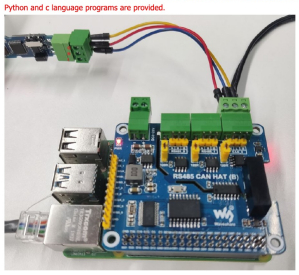
```
pi@raspberrypi:~ $ ifconfig
can0: flags=193<UP,RUNNING,NOARP>  mtu 16
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 10
(UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.10.88  netmask 255.255.252.0  broadcast 192.168.11.255
        inet6 fe80::7af1:9de7:6749:e572  prefixlen 64  scopeid 0x20<link>
        ether dc:a6:32:e6:84:86  txqueuelen 1000  (Ethernet)
        RX packets 7064  bytes 673465 (657.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 218  bytes 28256 (27.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

How To Use CAN

Hardware Connection

This demo uses a Raspberry Pi, an RS485 CAN HAT (B) module and a USB-CAN-A module. Python and c language programs are provided.



Simple Test

- Open CAN

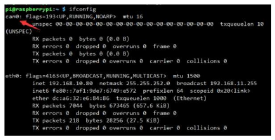
```
sudo ip link set can0 up type can bitrate 1000000
sudo ifconfig can0 txqueuelen 65536
```

- For more CAN kernel commands, please refer to:

```
https://www.kernel.org/doc/Documentation/networking/can.txt
```

- Check ifconfig:

```
ifconfig
```



- Install can-utils:

```
sudo apt-get install can-utils
```

- Receive:

The command for inputting the receiving data in the terminal:

```
candump can0
```

The receiving tool is blocked. When running the tool without parameters, it will always be in the receiving state. Use Ctrl+C to exit.
A description of the parameters of the tool can be used:

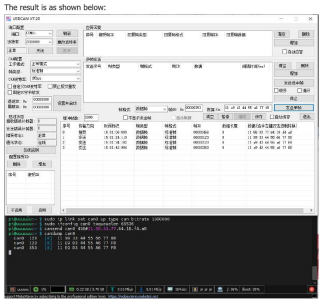
```
candump -h
```

- Sending

Input the command for sending the data in the terminal:

```
cansend can0 000#11.22.33.44
```

The result is as shown below:



- Self-test

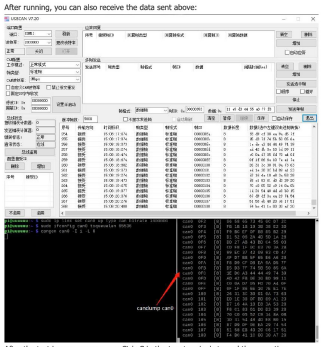
Input the command in the terminal:

```
cangen can0 -I i -L 8
#-I sets the device ID, 'i' is the incremental mode; -L sets the data length
```

At this time, data with the increasing device ID, length 8, and random content will be continuously sent to CAN. At the same time, this command enables loopback mode by default and can be run in another terminal window:

```
candump can0
```

After running, you can also receive the data sent above:



After the test is over, you can use Ctrl+C in the two terminals to end the operation.

Download The Demo

Run in the terminal of the Raspberry Pi:

```
sudo apt-get install unzip
```

[pdf]

RS485 CAN HAT B is a with and communication functions developed by Waveshare for Raspberry Pi C1XffryERaL m media amazon images I |||

RS485 CAN HAT B Overview RS485 CAN HAT B is a HAT with RS485 and CAN communication functions de ... from CH1 to CH8 according to the number and loop continuously. For more programs, you can refer to Modbus RTU Relay module wiki to change the serial port number of the program. and pay attention Re...

lang:en score:18 filesize: 5.02 M page_count: 1 document date: 2022-11-09

```
wget https://www.waveshare.com/w/upload/9/92/Rs485_CAN_HAT_B.zip
unzip RS485_CAN_HAT_B.zip
sudo chmod 777 -R RS485_CAN_HAT_B/
cd RS485_CAN_HAT_B
```

C

Note:

1. Please make sure the hardware connection is correct, that is, H-H and L-L connection.
2. The baud rate of both sides must be the same and the default demo setting is 1Mbps.
3. If the data loses frame due to the long time transmission, you can try to lower the baud rate.

- Send, the Raspberry Pi open the terminal, and run:

```
cd RS485_CAN_HAT_Code/CAN/wiringPi/send/
make clean
make
sudo ./can_send
```

- Blocking receiving, the Raspberry Pi opens the terminal and runs:

```
cd RS485_CAN_HAT_Code/CAN/wiringPi/receive/
make clean
make
sudo ./can_receive
```

The receiving program is blocked until the data is read.

Note: This program can only receive data whose frame ID is 123. If you need to receive other ID data, you can modify the program yourself.

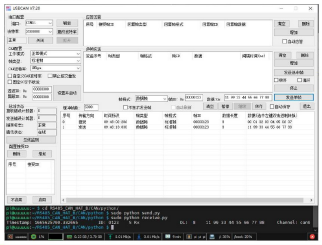
Take USB to CAN debug tool as an example:

Python

The Raspberry Pi opens the terminal and runs:

```
#the sending terminal:
sudo python can_send.py
cd RS485_CAN_HAT_B/CAN/python/
#Run the receiving program before sending the data from your computer
sudo python can_revelve.py
```

The effect is shown below:



How to use RS485

Hardware Connection

Detailed connection:

- Running the C and python demo requires an additional USB TO RS485 bidirectional converter to channel 1 to see the effect.
- In Python, test.py requires a connection between channel 1 and channel 2.

Func	BCM	Description
5V	5V	5V Power Input
GND	GND	Ground
SCLK_1	D21	SPI clock input
MOSI_1	D20	SPI data input
MISO_1	D19	SPI data output
CE_1	D18	Data/command selection
INT_1	D24	interrupt output

Test

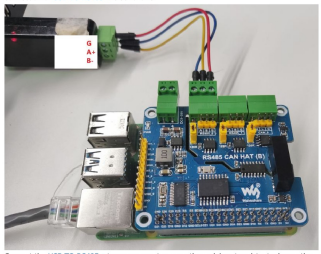
- Download and run the test demo

```
sudo apt-get install p7zip-full
wget https://www.waveshare.com/w/upload/9/92/Rs485_CAN_HAT_B.zip
unzip RS485_CAN_HAT_B.zip
sudo chmod 777 -R  RS485_CAN_HAT_B.zip
cd RS485_CAN_HAT_B/
```

C Program

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

The demonstration here is: The connection between channel 1 of RS485_0 and the A,B connection of USB TO RS485 is as follows:



Connect the USB TO RS485 to your computer, open the serial port assistant, choose the corresponding serial port, and set the baud rate as 115200.

- Execute the C program, the module returns the data that the computer sends to execute the C program, as shown below:

(Note: You need to add a carriage return and line feed, otherwise the data will not be returned; the program directory of the example is based on the actual.)

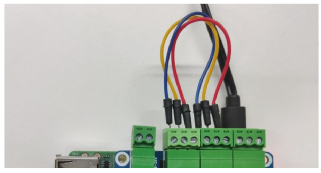
Python

```
cd python
cd examples
sudo python3 main.py
```

The hardware connection of this program and the C program is the same as the phenomenon:

You can try the following test demo when you don't have USB TO RS485:

- Connect the A, B of channel 1 to the A, B of channel 2.



- Run test.py and you can get the result below:

```
sudo python3 test.py
```

```
pi@raspberrypi:~/RS485_CAN_HAT_B/RS485/python/examples $ sudo python test.py
/dev/ttySC0
/dev/ttySC1
Channel 1 send channel 2 received successfully
waveshare_2_CH_RS485_HAT_1to2

Channel 2 send channel 1 received successfully
waveshare_2_CH_RS485_HAT_2to1


Channel 1 send channel 2 received successfully
waveshare_2_CH_RS485_HAT_1to2

Channel 2 send channel 1 received successfully
waveshare_2_CH_RS485_HAT_2to1


Channel 1 send channel 2 received successfully
waveshare_2_CH_RS485_HAT_1to2

Channel 2 send channel 1 received successfully
waveshare_2_CH_RS485_HAT_2to1
```

Modbus Application Example

This example requires a Raspberry Pi, an RS485 CAN HAT, and a Mobus RTU Relay.

Hardware Connection

- Install the HAT to the Raspberry Pi, and connect the RS485 interface of the HAT to the Modbus's RS485 interface by A-A and B-B.
- Power the Raspberry Pi and the Modbus module respectively.

Download Demo

- Download demo:

```
wget https://www.waveshare.com/w/upload/b/ba/RS485_Modbus_RTU_Relay_Cod
e.zip
```

- Enter the path after unzipping:

```
unzip RS485_Modbus_RTU_Relay_Code.zip
cd RS485_Modbus_RTU_Relay_Code/Modbus_RTU_Relay_Code/Python3
```

- Run the program

```
sudo python main.py
```

- After running the program, the relay relays are sequentially turned on from CH1 to CH8 and then turned off from CH1 to CH8 according to the number and loop continuously.
- For more programs, you can refer to Modbus RTU Relay module wiki and pay attention to change the serial port number of the program.

Resource

Document

- Schematic

Program

- Demo

Datasheet

- SP3485
- SN65HVD230
- MCP2515

FAQ

Question: Can't send and receive data?

Answer:

1. Make sure the baud rates on both sides are the same;
2. The fixed frame ID is set in the routine: 0X123, please set the sending and receiving CAN ID of the other end of your CAN to be x0123;

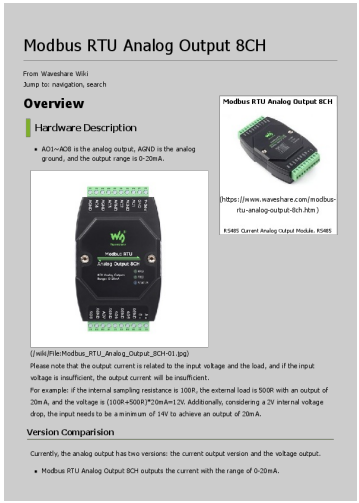
Question: How to configure the config.txt file when the Ubuntu system is installed on the Raspberry Pi?

Answer:

The mainstream Ubuntu system config.txt file is usually in the /boot/firmware folder.
Also, you 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 identify the SD card) through a card reader.

Support

If you require technical support, please go to the page and open a ticket.



Modbus RTU Analog Output 8CH

From Waveshare Wiki
Jump to: navigation, search

Overview

Hardware Description

- AO1~AO8 is the analog output, AGND is the analog ground, and the output range is 0-20mA.

(https://www.waveshare.com/modbus-rtu-analog-output-8ch.htm)

RS485 Current Analog Output Module, RS485

(/wiki/File:Modbus_RTU_Analog_Output_8CH-01.jpg)

Please note that the output current is related to the input voltage and the load, and if the input voltage is insufficient, the output current will be insufficient.
For example: if the internal sampling resistance is 100R, the external load is 500R with an output of 20mA, and the voltage is (100R+500R)*20mA=1.2V. Additionally, considering a 2V internal voltage drop, the input needs to be a minimum of 14V to achieve an output of 20mA.

Version Comparision

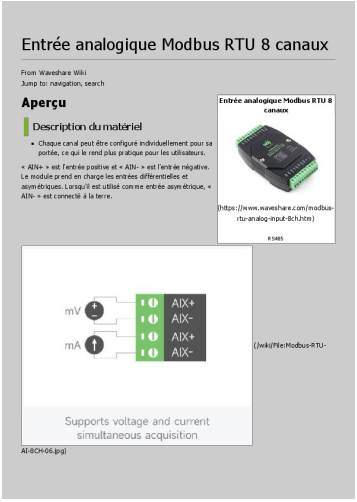
Currently, the analog output has two versions: the current output version and the voltage output.

- Modbus RTU Analog Output 8CH outputs the current with the range of 0-20mA.

[pdf]

Modbus RTU Analog Output 8CH Waveshare Wiki Sparwan Distributor of IoT IT equipment 26419 v 1726064800 cdn shopify s files 1 0747 1438 2601 |||

Modbus RTU Analog Output 8CH From Waveshare Wiki Jump to: navigation, search Overview Hardware Description AO1-AO8 is the analog output, AGND is the analog ground, and the output range is 0-20mA. Modbus RTU Analog Output 8CH https://www.waveshare.com/modbusrtu-analog-output-8ch.htm RS485 Curren... lang:en score:17 filesize: 1.01 M page_count: 16 document date: 2024-09-11


Entrée analogique Modbus RTU 8 canaux

From Waveshare Wiki
Jump to: navigation, search

Aperçu

Description du matériel

- Chaque canal peut être configuré individuellement pour sa portée, ce qui le rend plus pratique pour les utilisateurs.

« AIN+ » est l'entrée positive et « AIN- » est l'entrée négative. Le module prend en charge les entrées différentielles et asymétriques. Lorsqu'il est utilisé comme entrée asymétrique, « AIN- » est connecté à la terre.

Entrée analogique Modbus RTU 8 canaux

RS485

(https://www.waveshare.com/modbus-rtu-analog-input-8ch.htm)

mV
mA
AIX+
AIX-
AIX+
AIX-

Supports voltage and current simultaneous acquisition

(/wiki/File:Modbus-RTU-AI-8CH-06.jpg)

[pdf]

Modbus RTU Analog Input 8CH Waveshare Wiki Sparwan Distributor of IoT IT equipment 7a98261e 331b 4f5c bb9d d57e4700e0c2 v 1738579420 cdn shopify s files 1 0747 1438 2601 |||

Entre analogique Modbus RTU 8 canaux From Waveshare Wiki Jump to: navigation, search Aperu Entre ... oot If you want to insert the RS485 CAN HAT into the Raspberry Pi, connect the A and B ports of the Modbus RTU Relay module and the RS485 CAN HAT module. If you use other 485 devices, you need to make... lang:fr score:17 filesize: 1.78 M page_count: 23 document date: 2025-02-03


Modbus RTU Analog Input 8CH

From \Waveshare \Wiki
Jump to: navigation, search

Overview

Hardware Description

- Each channel can be individually configured for its range, making it more convenient for users. AIN+ is the positive input, and AIN- is the negative input. The module supports both differential and single-ended input. \When used as a single-ended input, AIN- is connected to the ground.

Modbus RTU Analog Input 8CH

RS485

File:Modbus-RTU-Analog-Input-6.png (/w/index.php?title=Special:Upload&wpDestFile=Modbus-RTU-Analog-Input-6.png)

Note: \When inputting the different powers, it is important to connect the ground wire to establish a common ground. Otherwise, the collected data may be inaccurate.

- Opening the device case reveals jumpers near the device terminals, corresponding to the eight channels AI1–AI8. You need to select the jumper mode based on the measurement signal; otherwise, the measurement data will be inaccurate.
  - \When measuring voltage signals, the jumper wires for the corresponding channels should be disconnected.
  - \When measuring current signals, the jumpers wires the corresponding channels should be connected.

[pdf]

Sparwan Distributor of IoT IT equipment Modbus RTU Analog Input 8CH Waveshare Wiki v 1718975891 cdn shopify s files 1 0747 1438 2601 |||

Modbus RTU Analog Input 8CH From Waveshare Wiki Jump to: navigation, search Overview Hardware Descr ... oot If you want to insert the RS485 CAN HAT into the Raspberry Pi, connect the A and B ports of the Modbus RTU Relay module and the RS485 CAN HAT module. If you use other 485 devices, you need to make... lang:en score:16 filesize: 1.44 M page_count: 18 document date: 2023-11-07