

RS232/485/422 TO CAN

User Manual



Contents

1. OVERVIEW	1
1.1 Features	1
2. QUICK START	2
2.1 Transparent Transmission Test	2
3. FUNCTION INTRODUCTION	5
3.1 Hardware Features	5
3.2 Device Features	5
4. Module HARDWARE INTERFACE	7
4.1 Module Dimensions	7
4.1 Module Pin Definition	8
5. MODULE PARAMETER SETTING	9
5.1 Serial Server Configure Software	9
6. TRANSFORM PARAMETERS	11
6.1 Conversion Mode	11
6.2 Conversion Direction	12
6.3 CAN Identifier in UART	12
6.4 Whether CAN is Transmitted in UART	13
6.5 Whether CAN Frame ID is Transmitted in UART	13
7. UART PARAMETER SETTING	14
8. CAN PARAMETER SETTING	15
8.1 CAN Baud Rate Setting	15
8.2 CAN Filter Setting	16
9. CONVERSION EXAMPLE	18
9.1 Transparent Conversion	18
9.1.1 Serial Frame To CAN	18
9.1.2 CAN Frame To UART	20

RS232/485/422 TO CAN User Manual

9.2 Transparent Conversion with ID	21
9.2.1 UART Frame To CAN	21
9.2.2 CAN Frame To UART	23
9.3 Format Conversion	24
9.4 Modbus Protocol Conversion	25

shore avesome hordware

1. OVERVIEW

RS232/485/422 TO CAN is an industrial RS232/485/422 to CAN isolated converter with protection circuits such as built-in power supply isolation, ADI magnetical isolation, TVS diode, and industrial rail-mount case design. RS232/485/422 To CAN is easy to operate with a zero-delay automatic transceiver circuit. Also, as it features fast communication, stability and reliability, and better compatibility, it is an ideal choice for industrial control equipment and/or applications with high communication requirements.

1.1 FEATURES

- ♦ Configurable CAN parameters (such as baud rate) and UART parameters.
- Onboard unibody power supply isolation, providing stable isolated voltage, no extra power supply required for the isolated terminal.
- ♦ Onboard unibody digital isolation, for isolating the signal, reliable and jamproof, low power consumption.
- ♦ Onboard TVS (Transient Voltage Suppressor), effectively suppresses surge voltage and transient spike voltage in the circuit, lightning-proof & anti-electrostatic.
- ♦ Onboard self-recovery fuse and protection diodes, ensure the current/voltage stable outputs, provide over-current/over-voltage proof, and improve shock resistance.
- ♦ Onboard 15KV ESD isolated protection and 600W lightning-proof & anti-surge protection.
- ♦ Onboard 120R terminal resistor on the RS485/RS422 ports, configurable by jumper.
- ♦ 4x LEDs for indicating the power and transceiver status.
- Support one button (press the **reload** button for more than 5 seconds) to restore the factory settings.
- Industrial grade metal case, with wall-mount and rail-mount support, rugged & durable, easy to install.

2. QUICK START

RS232/485/422 TO CAN is the device that supports the bidirectional transmission between RS232/485/422 and CAN. The device's CAN parameters (such as baud rate) and UART parameters are configurable via the software.

The related software: <u>WS-CAN-TOOL</u>.

2.1 TRANSPARENT TRANSMISSION TEST

First, you can test it with the default parameters of the product, as shown below:

Item	Parameters
RS232/485/422	115200, 8, N, 1
CAN Operation Mode	Transparent Transmission, Bidirectional
CAN Baud Rate	250kbps
CAN Sending Frame Type	Extended Frames
CAN Sending Frame ID	0 x 12345678
CAN Filter	Disabled (Receive all CAN frames)

RS232 to CAN transparent transmission test:

Use the serial cable to connect the computer and the RS232 port (DB9) of the device, and connect the USB to CAN debugger (the first time you use it, you need to install the software and driver, please consult the relevant manufacturers of the USB to CAN debugger for the detailed use), and then the DC5V 1A power adapter to power on the device.



Figure 1.2.2: RS232 TO CAN Data Transparent Transmission

Open the SSCOM, select the COM port to be used, and set the UART parameters as shown in Figure 1.2.2. After setting, you can enter the serial port, open USB to CAN debugging software, and set the baud rate as 250kbps.

After following the above steps, the CAN and RS232 can send data to each other.

• RS485 to CAN Transparent Transmission:

Use the UART (/USB) to RS485 cable to connect the computer and the RS485 connector of the device (GREEN plug, A-A, B-B). After connecting the USB to CAN debugger (you need to install the software and the driver for the first time, and you can consult the manufacturer for details), you can use a DC5V 1A power adapter to power the device.

The test method is as same as the RS232 and CAN transparent transmission.

• RS422 to CAN Transparent Transmission:

Use UART (/USB) to RS422 cable to connect the computer and the RS422 connector of the device (GREEN plug, TA-RA, TB-B). After connecting the USB to CAN debugger (you need to install the software and the driver for the first time, and you can consult the manufacturer for details), you can use a DC5V 1A power adapter to power the device.

The test method is as same as the RS232 and CAN transparent transmission.

shore avesome hordwore

3. FUNCTION INTRODUCTION

RS232/485/422 TO CAN has onboard 1-channel RS232/485/422 interface and 1-channel CAN interface. The baud rate of the serial port supports 1200~460800bps; the baud rate of CAN supports 10kbps~1000kbps, and the firmware upgrade of the device can be realized through the TTL interface, which is very convenient to use.

Users can easily complete the interconnection of serial devices and CAN devices.

3.1 HARDWARE FEATURES

No.	Item	Parameters
1	Model	RS232/485/422 TO CAN
2	Power	5V@200mA
3	CPU	32-bit High-performance Processor
4	CAN Interface	ESD Protection, Anti-surge Protection, Excellent EMC
4	CAN Interface	Performance
5	RS232/485/422 Interface	The baud rate supports 1200~460800
6	Communication Indicator	RUN, COM, CAN indicator, easy to use
7	Reset/Restore Factory	Comes with the setting signal for Reset/ Restore Factory
	Setting	Setting
8	Operation Temperature	Industrial Grade: -40~85°C
9	Storage Temperature	-65~165℃

3.2 DEVICE FEATURES

- ♦ Support the bidirectional data communication between CAN and RS432/285/422.
- ♦ The device parameters are configurable through RS232/485/422.
- ♦ ESD Protection, Anti-surge Protection, Excellent EMC Performance.
- ♦ 14 set configurable filters.
- ✤ Four operation modes: transparent conversion, transparent conversion with identifiers, format conversion, and Modbus RTU protocol conversion.
- ♦ Offline detection and automatic recovery functionality.
- ♦ Compliance with CAN 2.0B specifications, compatible with CAN 2.0A; complies with ISO

HONOR

11898-1/2/3 standards.

- \diamond Baud rate range: 10kbps ~ 1000kbps.
- ♦ CAN buffer capacity of 1000 frames to prevent data loss.
- High-speed conversion: At a serial port baud rate of 115200 and CAN rate of 250kbps, the CAN sending speed can reach up to 1270 extended frames per second (close to the theoretical maximum of 1309). At a serial port baud rate of 460800 and CAN rate of 1000kbps, the CAN sending speed can exceed 5000 extended frames per second.

dre owesome



4. Module HARDWARE INTERFACE

4.1 MODULE DIMENSIONS



Unit: mm



4.1 MODULE PIN DEFINITION

Label	Description	Note
VCC	Module power positive	5~36V
GND	Module power negative	Ground
CAN-H	CAN differential positive	The module has an internal 120ohm resistor
CAN-L	CAN differential negative	-
CFG	Reset/restore factory setting	Press to restore in 5s, restore factory setting after 5s
	RS422 sends differential signal	
TA/A+	positive	-
	RS485 differential signal positive (A+)	
	RS422 sends differential signal	
TB/B-	negative	-
	RS485 differential signal negative (B-)	
RA	RS422 receive differential signal	_
	positive	
RB	RS422 receive differential signal	_
	negative	
	Shore	wesoli

5. MODULE PARAMETER SETTING

This module can be configured by "<u>WS-CAN-TOOL</u>" through the RS232/485/422 interface. If you fail to connect the device due to your careless setting, you can press the "CFG" key to restore the factory setting, (Press and hold the CFG key for 5s, and release it after the three green indicators blink at the same time).

5.1 SERIAL SERVER CONFIGURE SOFTWARE

rialPort <mark>COM42</mark> ~ 1	Baudrate AUTO 🗸 ParityBit NONE 🗸 DataBi	t ⁸ V StopBit 1 V Open Serial
Device Informati	on	
Mode	S/N	Version
Serial Parameter	s CAN Parameters	Transform Parameters
Baudrate	V Baudrate V Custom Baudrate	Mode 🗸 🗸 🗸
	CAN ID Frame Type	Direction
ParityBit	V Number Type Acess Control Acess	The Position of CAN ID
D + D'+		Offset
		Length
StopBit		□Add Frame info in serial
	Add 🗹 Modify 💆 Delete	□Add ID in serial

- Select the connected "Serial Port".
- Click on "Open Serial".
- Click on "Read Device Parameters".



∑ WS-CAN-TOOL Device Language About SerialPort[COM42 ∨ Bau	idrate AUT	0 ~ Par	ityBit NONE 🗸	DataBit 8	∽ StopBi	1 ~	- Close	e Seri	ia]
Device Information Mode ZQWL-RCAN-31	3T11)	S/N 2d9a94e	e9293		Version D	D119		
Serial Parameters Baudrate 115200 v	CAN Para Baudrat CAN II	neters e 250k bps) 1234567	 Custom Base Frame Type 	udrate 250.00	Tra k bps Dir	nsform Par: Mode ection	ameters)	~ ~
ParityBit NONE 🗸 DataBit 8 ✓	Number	Туре	Acess Control	Acess	Th	e Position)ffset 0 .ength 4	of CAN	ID	~ ~
StopBit 1	Add	Modify	🖞 Delete			dd Frame i dd ID in s	nfo in s erial	erial	1
Read Device Paramet	ers Sa	ve Device P	arameters	Restart Devi	ce I	Reset to Fa	actory se	etting	g

After reading the device parameters, you can modify them. You can click on "Save Device Parameters" to save your modification. Then you need to reboot the device.

The following content is for explaining the parameters in the configured software.

6. TRANSFORM PARAMETERS

This section specifies the device's conversion mode, conversion direction, the position of CAN identifiers in the serial sequence, whether CAN information is transformed to UART, and whether CAN frame IDs are transformed to UART.

6.1 CONVERSION MODE

Three conversion modes: transparent conversion, transparent conversion with identifiers, and format conversion.

• Transparent conversion

It involves converting bus data from one format to another without adding or modifying data. This method facilitates an exchange of data formats without modifying the data content, making the converter transparent to both ends of the bus. It doesn't add communication overhead for users and allows real-time, unaltered data conversion, capable of handling high-volume data transmission.

• Transparent conversion with identifiers

This is a special application of transparent conversion, also without adding a protocol. This conversion method is based on the common characteristics of typical serial frames and CAN messages, allowing these two different types of buses to seamlessly form a single communication network. This method can map the "address" from the serial frame to the identifier field of the CAN message. The "address" in the serial frame can be configured in terms of its starting position and length, enabling the converter to adapt to user-defined protocols to the maximum extent in this mode.

• Format conversion

Additionally, the format conversion is the simplest usage mode, where the data format is defined as 13 bytes, encompassing all information from the CAN frame.



6.2 CONVERSION DIRECTION

Three conversion directions: bidirectional, only UART to CAN, and only CAN to UART.

Bidirectional

The converter converts data from the serial bus to the CAN bus and also from the CAN bus to the serial bus.

• Only UART to CAN

It only translates data from the serial bus to the CAN bus and doesn't convert data from the CAN bus to the serial bus. This method effectively filters out interference on the CAN bus.

• Only CAN to UART

It exclusively translates data from the CAN bus to the serial bus and doesn't convert data from the serial bus to the CAN bus.

6.3 CAN IDENTIFIER IN UART

This parameter only be effective when it is in "Transparent conversion with identifiers" mode:

4	Transform Parameters
	Node Transform with 1~
	Direction two-way ~
	The Position of CAN ID
	Offset 0 ~
	Length 4
	Add Frame info in serial
	Add ID in serial

When converting serial data to CAN messages, the offset address of the frame ID's starting byte in the serial frame and the length of the frame ID are specified.

The frame ID length can range from 1 to 2 bytes for standard frames, corresponding to ID1 and

ID2 in the CAN message. For extended frames, the ID length can range from 1 to 4 bytes, covering ID1, ID2, ID3, and ID4. In standard frames, the ID consists of 11 bits, while in extended frames, the ID consists of 29 bits.

6.4 WHETHER CAN IS TRANSMITTED IN UART

This parameter is only used in "Transparent Conversion" mode. When selected, the converter will include the frame information of the CAN message in the first byte of the serial frame. When deselected, the frame information of the CAN won't be converted into the serial frame.

6.5 WHETHER CAN FRAME ID IS TRANSMITTED IN UART

This parameter is exclusively used in "Transparent Conversion" mode. When selected, the converter will include the frame ID of the CAN message before the frame data in the serial frame, following the frame information (if frame information conversion is allowed). When deselected, the CAN frame ID won't be converted.

UART PARAMETER SETTING

- Baud rate: 1200~406800 (bps) ۲
- UART parity method: no parity, even, odd •
- Data bit: 8 and 9 •
- Stop bit: 1, 1.5 and 2 •

shore owesome hordwore

8. CAN PARAMETER SETTING

This part introduces how the converter CAN set the baud rate, CAN send ID, frame type and CAN filter of the converter. CAN baud rate supports 10kbps~1000kbps and also supports user's definition. Frame types support extended frames and standard frames. The frame ID of CAN is in hexadecimal format, which is valid in "transparent conversion" mode and "transparent conversion with ID" mode, and sends data to the CAN bus with this ID; This parameter is not valid in Format Conversion mode.

There are 14 groups of CAN receiving filters, and each group consists of "filter type", "filter acceptance code" and "filter mask code".

8.1 CAN BAUD RATE SETTING

Most common baud rates have been reserved in the list: this device does not support customization.

\smile	
k bps 🗸	125k
k hns	250k
bps	10k
bps	15k
bps	20k
bos	25k
bos	40k
bps	50k
bps	80k
k bps	100k
k bps	25k
k bps	200k
k bps	400k
k bps	500k
k bps	666k
k bps	800k
Ok bps	1000k
tomize	custon

8.2 CAN FILTER SETTING

14 groups of CAN receiving filters are disabled by default, which means the data of the CAN bus are not filtered. If users need to use filters, you can add them in the configured software, 14 groups can be added.

Number	Туре	Acess Control	Acess	
				_
🖪 Add	🖸 Modi	ify 🗗 Delete		

Filter mode: optional "Standard Frame" and "Extended Frame".

Filter acceptance code: used to compare the frame ID received by CAN to determine whether the frame is received in hexadecimal format.

Filter mask code: used to mask some bits in the acceptance code to determine whether some bits of the acceptance code participate in the comparison ((bit is 0 for non-participation, 1 for participation), in hexadecimal format.

Example 1: Filter type selected: "Standard Frame"; "Filter Acceptance Code" filled with 00 00 00 01; "Filter Mask Code" filled with 00 00 0F FF.

Explanation: As the standard frame ID consists of only 11 bits, the last 11 bits of both the acceptance code and the mask code are significant. With the mask code's final 11 bits all set to 1, it means that all the corresponding bits in the acceptance code will be considered for comparison. Therefore, the mentioned configuration allows the standard frame with an ID of 0001 to pass through.

Example 2: Filter type selected: "Standard Frame"; "Filter Acceptance Code" filled with 00 00 00 01; "Filter Mask Code" filled with 00 00 0F F0.

Explanation: Similar to example 1, where the standard frame has only 11 valid bits, the last 4 bits of the mask code are 0, indicating that the last 4 bits of the acceptance code will not be considered

for comparison. Hence, this configuration allows a group of standard frames ranging from 00 00 to 000F in ID to pass through.

Example 3: Filter type selected: "Extended Frame"; "Filter Acceptance Code" filled with 00 03 04 01; "Filter Mask Code" filled with 1F FF FF FF.

Explanation: Extended frames have 29 bits, and with the mask code's last 29 bits set to 1, it means that all the last 29 bits of the acceptance code will be involved in comparison. Therefore, this setting enables the passage of the extended frame with an ID of "00 03 04 01".

Example 4: Filter type selected: "Extended Frame"; "Filter Acceptance Code" filled with 00 03 04 01; "Filter Mask Code" filled with 1F FC FF FF.

Explanation: Based on the provided settings, a group of extended frames ranging from "00 00 04 01" to "00 0F 04 01" in ID can pass through.

Ctondond		
Standard	00 00 00 01	00 00 03 FF
Standard	00 00 00 01	00 00 03 FO
Extended	00 03 04 01	1F FF FF FF
Extended	00 03 04 01	1F FC FF FF
	Standard Extended Extended	Standard 00 00 00 01 Extended 00 03 04 01 Extended 00 03 04 01

9. CONVERSION EXAMPLE

9.1 TRANSPARENT CONVERSION

In transparent conversion mode, the converter promptly converts and sends the data received from one bus to the other bus without delay.

9.1.1 SERIAL FRAME TO CAN

The entire data of the serial frame is sequentially populated into the data field of the CAN message frame. Once the converter receives a frame of data from the serial bus, it immediately transfers it to the CAN bus. The information of the converted CAN message frame (the frame type section) and the frame ID are pre-configured by the user, and throughout the conversion process, the frame type and frame ID remain unchanged.

The data conversion follows the following format:

If the length of the received serial frame is less than or equal to 8 bytes, characters 1 through n (where n is the length of the serial frame) are sequentially placed into positions 1 through n of the CAN message's data field (with n being 7 in the illustration).

If the number of bytes in the serial frame is more than 8 bits, the processor starts from the first character of the serial frame, takes the first 8 characters, and fills them sequentially into the data field of the CAN message. Once this data is sent to the CAN bus, the remaining serial frame data is converted and filled into the data field of the CAN message until all the data has been converted.

13

		CAN messag	ge (Standard frame)
Serial Frame		Frame Information	Frame information
Format		Frame ID	User configuration
			User
			configuration
Data 1	16		Data 1
Data 2		[Data 2
Data 3			Data 3
Data 4		Data	Data 4
Data 5		Division	Data 5
Data 6			Data 6
Data 7			Data 7

For example, the CAN parameter setting selects "Standard Frame", and the CAN ID is 00000060, note that only the last 11 bits of the standard frame are valid.

			CAN message 1	CAN message 2
Serial Frame Format		Frame Information	08	05
		Frame ID 1	00	00
		Frame ID 2	60	60
01			01	09
02			02	10
03			03	11
04	25	Data	04	12
05		Division	05	13
06			06	
07			07	
08			08	
09	1			
10				
11				
12				



9.1.2 CAN FRAME TO UART

On the CAN bus message, it promptly forwards one frame upon receiving one frame. The data format corresponds as shown in the diagram.

During conversion, all the data present in the data field of the CAN message is sequentially converted into the serial frame.

If, during configuration, the setting "Whether CAN information is to be converted into serial" is enabled, the converter will directly fill the "Frame Information" byte of the CAN message into the serial frame.

Similarly, if the setting "Whether CAN Frame ID is to be converted into serial" is enabled, all the bytes of the CAN message's "Frame ID" will be filled into the serial frame.

Serial Frame Format	CAN messa	nge (Standard frame)
Frame information	Frame Information	Frame information
Frame ID 1	Emma ID	Frame ID 1
Frame ID 2	r rame ID	Frame ID 2
Data 1		Data 1
Data 2		Data 2
Data 3	1000	Data 3
Data 4	Data	Data 4
Data 5	Division	Data 5
Data 6		Data 6
Data 7		Data 7

For example, if "Convert CAN Message to Serial" is enabled but "Convert CAN Frame ID to Serial" is disabled, the conversion of a CAN frame to a serial format would be as depicted in the

following diagram:



9.2 TRANSPARENT CONVERSION WITH ID

Transparent conversion with ID is a specialized use of transparent conversion that facilitates users in constructing their networks more conveniently and employing custom application protocols.

This method automatically converts the address information from a serial frame into the frame ID of the CAN bus. By informing the converter about the starting address and length of this address in the serial frame during configuration, the converter extracts this frame ID and converts it into the frame ID field of the CAN message. This serves as the ID of the CAN message when forwarding this serial frame. When converting a CAN message into a serial frame, the ID of the CAN message is also translated into the respective position within the serial frame. It's important to note that, in this conversion mode, the "CAN ID" setting in the "CAN Parameter Settings" of the configuration software is invalid. This is because, in this scenario, the transmitted identifier (frame ID) is populated from the data within the aforementioned serial frame.

9.2.1 UART FRAME TO CAN

Upon receiving a complete serial data frame, the converter promptly forwards it to the CAN bus.

The CAN ID carried within the serial frame can be set within the configuration, specifying its starting address and length within the serial frame. The range for the starting address is from 0 to 7, while the length ranges from 1 to 2 for standard frames and 1 to 4 for extended frames.

During conversion, based on the pre-configured settings, all CAN frame IDs within the serial frame are entirely translated into the frame ID field of the CAN message. If the number of frame IDs within the serial frame is fewer than the number of frame IDs within the CAN message, the remaining IDs within the CAN message are filled in the order of ID1 to ID4, with the remaining one filled with "0". The rest of the data undergoes sequential conversion as shown in the diagram.

If a single CAN message frame does not complete the conversion of the serial frame data, the same ID continues to be used as the frame ID for the CAN message until the entire serial frame has been completely converted.

-						
Serial For	Frame mat			CAN message 1	CAN message	CAN message x
Address 0	CAN frame ID		Frame Information	User configuration	User configuration	User configuration
Address 1	Data 1		Frame ID 1	00	00	00
Address 2	Data 2		Frame ID 2	Data 4 (CAN frame ID 1)	Data 4 (CAN frame ID 1)	Data 4 (CAN frame ID 1)
Address 3	Data 3			Data 1	Data	Data n-4
Address 4	Data 5	\square		Data 2	Data	Data n-3
Address 5	Data 6		Data	Data 3	Data	Data n-2
Address 6	Data 7		Division	Data 5	Data	Data n-1
Address 7	Data 8			Data 6	Data	Data n
				Data 7	Data	
Address	Data n			Data 8	Data	
(n-1)	Data II			Data 9	Data	

For example, the initial address of the CAN ID in the serial frame is 0, the length is 3 (in the extended

frame), the serial frame and the CAN message are as shown below. Note that the two frames of CAN messages are converted in the same ID.

Seria F	al Frame ormat			CAN message 1	CAN message 2
Address 0	Data 1 (CAN frame ID 1)		Frame Information	88	85
Address 1	Data 2 (CAN frame ID 2)		Frame ID 1	00	00
Address 2	Data 3 (CAN frame ID 3)		Frame ID 2	Data 1 (CAN frame ID 1)	Data 1 (CAN frame ID 1)
Address 3	Data 1		Frame ID 3	Data 2 (CAN frame ID 2)	Data 2 (CAN frame ID 2)
Address 4	Data 2	\implies	Frame ID 4	Data 3 (CAN frame ID 3)	Data 3 (CAN frame ID 3)
Address 5	Data 3			Data 1	Data 9
Address 6	Data 4			Data 2	Data 10
Address 7	Data 5		Data	Data 3	Data 11
Address 8	Data 6		Division	Data 5	Data 12
Address 9	Data 7			Data 6	
Address 10	Data 8			Data 7	
Address 11	Data 9			Data 8	
Address 12	Data 10				
Address 13	Data 11				
Address 14	Data 12				
		12			

9.2.2 CAN FRAME TO UART

If the initial address of the configured CAN ID is 0 in the serial frame and a length of 3 (in the case of extended frames), the CAN message and the result of converting it to a serial frame is shown below:

Serial Frame Format		CAN	message
20		Frame Information	87
30			10
40		E	20
Data 1		Frame ID	30
Data 2	\leq		40
Data 3	10000		Data 1
Data 4			Data 2
Data 5		Data	Data 3
Data 6		Division	Data 4
Data 7			Data 5
	-		Data 6
			Data 7

9.3 FORMAT CONVERSION

Data conversion format as shown below. Each CAN frame includes 13 bytes, and they include CAN information + ID +data.



FF: Frame identifier, 1 indicates an extended frame and 0 indicates a standard frame. RTR: Frame identifier, 1 is the remote frame, 0 is the data frame. retain: Reserve The reserved value is 0 and cannot be written to 1. $D3 \sim D0$: Identifies the data length of the CAN frame.



[Frame ID: The length is 4 bytes, the standard frame is 11 bits, the extended frame is 29 bits.										
I	ow Byte			High By	/te	Low B	yte		Hig	<mark>gh Byt</mark> e	
	12h	34h	56h	78h		00 h	001	h 03	h	FFh	
Т	he above nu	e is the mber 0	extende (123456	ed frame 578	ID	The ak	oove is tl num	he stand ber 0x3	dardfra FF	ame ID	
[Fram	e data:	The leng by the v	th is 8 by alue of D	tes, and 3~D0 of	the effe the fram	ctive ler ne inforr	ngth is natior	detern 1.	nined
	DATA1							DATA8	-		
	11h	22h	33h	44h	55h	66h	77h	88h			
	рата 1	т	he aboy 8	ve is the bytes of	representa valid data	ation of		DATAS			
	11h	22h	33h	44h	55h	66h	00h	00h			
		Th	e abov 6 k	e is the ro oytes of v	epresenta alid data	tion of					
								X	Ţ	94	-
Tł	ne follow	ving ex	ample	is the re 8 bytes	presentat of data (1	ion of a 1h, 22h,	n exten 33h, 44	ded dat Ih, 55h,	ta fran 66h, 1	ne witł 77h, 88	n ID h)
0)	(125450		annig								
0) 88h	Ox12345678 containing 8 bytes of data (11h, 22h, 33h, 44h, 55h, 66h, 77h, 88h) 8h 12h 34h 56h 78h 11h 22h 33h 44h 55h 66h 77h 88h										
0 > 88h	125436 12h The fc ID 0x3	34h ollowing	56h g exam taining	78h 1 ple is the 6 bytes	1h 22h e represen of data (1	1 33h ntation 6 11h, 22h,	44h of a star 33h, 44	55h ndard d 4h, 55h,	66h ata fr 66h)	77h ame,	88h

Note: Each frame is fixed to 13 bytes, and any deficiency must be filled with 0.

9.4 MODBUS PROTOCOL CONVERSION

Convert the standard Modbus RTU serial data protocol to the specified CAN data format, and this conversion generally requires the editable CAN bus device message.

The serial data must be compliant with the standard Modbus RTU protocol, otherwise it can not be converted. Please note that CRC parity can not be converted to CAN.

The CAN formulates a simple and efficient segment communication format to realize Modbus RTU communication, which does not differentiate between host and slave, and users only need to communicate according to the standard Modbus RTU protocol.

The CAN does not require CRC checksum, and after the converter receives the last CAN frame, the CRC will be added automatically. Then, a standard Modbus RTU data packet is formed and sent to the serial port.

In this mode, the **[CAN ID]** of the **[CAN Parameter Setting]** of the configuration software is invalid, because the identifier (frame ID) sent at this time is filled by the address field (node ID) in the Modbus RTU serial frame.

(1) Serial frame format (Modbus RTU)

Serial parameters: baud rate, data bits, stop bits and parity bits can be set via configuration software. The data protocol needs to conform to the standard Modbus RTU protocol.

(2) CAN

The CAN side designs a set of segment protocol formats, which defines designed a segmentation protocol format that defines a method for segmenting and reorganizing a message that is greater than 8 bytes in length, as shown below. Note that when the CAN frame is a single frame, the segmentation flag bit is 0x00.

		-						
Bit No.	7	6	5	4	3	2	1	0
Frame	FF	FTR	X	X	Ι	DLC (da	ta length)
Frame ID1	X X X ID.28-ID.24							
Frame ID2		ID.23	8-ID.16					
Frame ID3		ID.15	5-ID.8					
Frame ID4		l	ID.7-ID.0	(Modbus	RTU add	dress)		
Data 1	segmentation	segmer	ntation		segmen	ntation c	ounter	
Dutu I	Segmentation	Jeginei						
	flag	ty	pe					
Data 2	flag	ty	pe Cł	naracter 1	L			
Data 2 Data 3	flag	ty	pe Cł Cł	naracter 1 naracter 2	L 2			
Data 2 Data 3 Data 4	flag	ty	pe Cł Cł Cł	naracter 1 naracter 2 naracter 3	L 2 3			
Data 2 Data 3 Data 4 Data 5	flag	6 5 4 3 2 1 0 FTR X X DLC (data length) X X ID.28-ID.24 ID.23-ID.16 ID.15-ID.8 ID.7-ID.0 (Modbus RTU address) ID.7-ID.0 (Modbus RTU address) On segmentation type Segmentation counter Character 1 Character 2 Character 3 Character 4						
Data 2 Data 3 Data 4 Data 5 Data 6	flag	ty	CH CH CH CH CH CH	naracter 1 naracter 2 naracter 3 naracter 4 naracter 5	L 2 3 4			
Data 2 Data 2 Data 3 Data 4 Data 5 Data 6 Data 7	flag	ty	CF CF CF CF CF CF CF	naracter 1 naracter 2 naracter 3 naracter 4 naracter 5 naracter 6	1 2 3 4 5			

The CAN frame message can be set by the configuration software (remote or data frame; standard or extended frame).

The transmitted Modbus protocol starts from "Data 2" byte, if the protocol content is more than 7 bits, and the rest of the protocol content is converted in this segmented format until the conversion is



complete.

Data 1 is segmentation control message (1 byte, 8bit), and the meaning as shown below:

• Segmentation Flag

The segmentation mark occupies one bit (Bit7), and indicates whether the message is a segmented message or not. "0" indicates a separate message, and "1" indicates a frame in a segmented message.

• Segmentation Type

The segmentation type occupies 2 Bits (Bit6, Bit5), and indicates the types of the report in this segment report.

Bit Value (Bit6, Bit5)	Description	Note
00	The first segmentation	If the segmentation counter
00	The first segmentation	this is the first segmentation.
		Indicates this is the middle
01	The middle segmentation	segmentation, and there are
	The middle segmentation	multiple segmentation or there
		are no middle segmentation.
10	The last segmentation	Indicates the last segmentation

• Segmentation Counter

Occupies 5 bits (Bit4-Bit0), used to distinguish the serial number of segments in the same frame Modbus message, enough to verify whether the segments of the same frame are complete.

(3) Conversion Example:

The serial port side Modbus RTU protocol (in hex).

01 03 14 00 0A 00 00 00 00 00 14 00 00 00 00 00 17 00 2C 00 37 00 C8 4E 35

 \mathcal{O}_{1}

The first byte **01** is the Modbus RTU address code, converted to CAN ID.7-ID.0;

The last 2 bytes (4E 35) are Modbus RTU CRC checksums, which are discarded and not converted.

The final conversion to CAN data message is as follows:

Frame 1 CAN message: 81 03 14 00 0A 00 00 00 00

Frame 2 CAN message: **a2** 00 00 14 00 00 00 00 00

Frame 3 CAN message: a3 00 17 00 2C 00 37 00

CAN message frame 4: c4 c8

The frame type (standard or extended frame) of the CAN telegrams is set via the configuration software;

The first data of each CAN message is filled with segmented information (81, a2, a3 and c4), which is not converted into Modbus RTU frames, but only serves as acknowledgment control information for the message.

	OM V5.13.1 Se	erial/Net (data debugge	r.Author:Tintin.a	2618058@aa.com(f	Newest version	on)		×
PORT	COM_Settings	Display	Send_Data	Multi_Strings	Tools Help 联	系作者 大虾	论坛		
[18:38:48	3. 291]0VT→¢0	11 03 14 0	0 04 00 00 00	00 00 14 00 00	00 00 00 17 00 20	00 37 00 C8	4E 35 🗆		~
ClearDa	te∫_OpenFile) nmui and chàb⊒⊡	[Show Saw Price	SendFile Si	top ClearSe	ad⊂ OnTopI⊄ I	English SeveConfig	
ComNum C	OM1 通信端口	More Se	ttings V Show	Show SaveData	ReceivedToFile	e 🔽 SendHEX s No 1 BytesT	SendEvery: 50	None 🗸 🕻	CrLf 3
	DTR BaudRe	at 115200	• 01 03 1	4 00 0A 00 00	00 00 00 14 00 00 (00 00 00 17 0	00 2C 00 37 00	C8 4E 35	0
为了更好地	地发展SSCOM软件 基立合地结尾客口	‡ s)	CHID						~
1青72©王册哥 ▲Q群满员	。立已即结尾各月 【了,没有第二个	*群.有需要		注册]★合宙高的	上价比4G模块 ★RT-Th	mread中国人的	开源免费操作系统	统 ★ ★8KM近距离	wiFi可自由
www.daxi	ia.com S:25	F	ξ:0	COM1 Closed	115200bps,8,1,Non	e,None			
BCAN V1.20									
onfigure		Reply t	o reply						
: COM89 ~	Refresh	No Re	ceive ID Re	ply Frame Type	Reply Frame Form	nat Reply I	ID Reply Data	۱ ()	Clear Dele
ps: 2000000 🗸	Change bps								Add
l Close	Open								Auto Answe
onfimme		More fr	ames to send	The second	a la Trace de				
Normal mode		C J W.	Frank Trees	Frame 1		Data		Interval (ms)	The second secon
Normal mode Standard fram	~ Ie ~	Send No	rrame type		Format Frame ID	Data		100	Clear Dele
Normal mode Standard fram 250kbps		0 1	Standard fr Standard fr	ame Datafi ame Datafi	format frame ID rame 00000000 rame 00000000	00 01 02 0 00 01 02 0	03 04 05 06 07 03 04 05 06 07	100 100	Clear Dele Add
Normal mode Standard fram ops: 250kbps nual set bps 0n		0 1	Standard fr Standard fr	ame Datafi ame Datafi	format frame LD rame 00000000 rame 00000000	00 01 02 0 00 01 02 0	03 04 05 06 07 03 04 05 06 07	100 100	Clear Dele Add Send selected fro
Normal mode Standard fram ops: 250kbps nual set bps 0n ced 20 bytes to send	ne ~ Ny send once and receive	0 1	Standard fr Standard fr	ame Datafi ame Datafi	format Frame ID rame 00000000 rame 00000000	00 01 02 0 00 01 02 0	03 04 05 06 07 03 04 05 06 07	100 100	Clear Dele Add Send selected fro Sequence Cy
Mormal mode Standard fram ops: 250kbps uual set bps On ced 20 bytes to send r ID:0x 00000000	e ~ vily send once and receive	0	Standard fr Standard fr	ame Datafi ame Datafi	"ormat Frame IJ rame 00000000 rame 00000000	00 01 02 0 00 01 02 0	03 04 05 06 07 03 04 05 06 07	100 100	Clear Dele Add Send selected fro Sequence Cy Stop
Wormal mode Standard fram ps: 250kbps vual set bps On cd 20 bytes to send r ID:0x 00000000 SD:0x S	e ~ Jy send once and receive et and Start	0 1	Standard fr Standard fr Standard fr	ame Data fi ame Data fi mat: Data fran	format Frame ID rame 00000000 rame 00000000 ne V ID: 0x 0	00 01 02 0 00 01 02 0 00 01 02 0	03 04 05 06 07 03 04 05 06 07 03 04 05 06 07	100 100 3 44 55 66 77 88	Clear Dele Add Send selected fr Sequence Cy Stop Send a single fr
Normal mode Normal mode Standard fram ps: 250kbps ual set bps On ed 20 bytes to send In: 0x D00000000 S D: 0x 00000000 S tate	ve v ly send once and receive set and Start Total	5000	Frame Type Standard fr Standard fr For	ame Data fi ame Data fi mat: Data fram	rame 00000000 rame 00000000 ne ID: 0x 0	00 01 02 0 00 01 02 0 00 01 02 0 0000123 Da	03 04 05 06 07 03 04 05 06 07 .ta:0x 11 22 3 aar Pause Co	100 100 3 44 55 66 77 88 ntinue Save [Clear Dele Add Send selected fro Sequence Cy Stop Send a single fro Auto save Ex
Normal mode Standard fram ps: 250kbps ual set bps On of 20 bytes to send D D: 0x00000000 S tate e error: O	e v ly send once and receive set and Start Total: No Dir.	5000 ection	Frame Type Standard fr Standard fr For [Time scale	ame Data fr ame Data fr mat: Data fran Display recei Frame Type	rame 00000000 rame 000000000 ne 00000000 ne ID: 0x ive only Auto B Frame Format	00 01 02 0 00 01 02 0 00 01 02 0 0000123 Da tefresh Cle Frame ID	03 04 05 06 07 03 04 05 06 07 	100 100 3 44 55 66 77 88 ntinue Save [Data(LDouble-ol	Clear Dele Add Send selected fr Sequence Cy Stop Send a single fr Auto save Ex Lick Hex->Dec)
Normal mode Standard fram ps: 250kbps ual set bps On of 20 bytes to send S DD: 0x 0000000 D: 0x 0000000 state e error: it error: 0	ve v uly send once and receive set and Start Total: No Dir. O Rec	5000 ection eive	Frame 1996 Standard fr Standard fr For [Time scale 18:38:48:299	ame Data fr ame Data fr mat: Data fran Display recei Frame Type Data frame	reme 00000000 rame 00000000 ne ID: 0x 0 ive only Auto R Frame Format Standard frame	00 01 02 0 00 01 02 0 00 01 02 0 0000123 Da tefresh Cle Frame ID 00000001	03 04 05 06 07 03 04 05 06 07 ata:0x 11 22 3 ar Pause Co Data Length 8	100 100 3 44 55 66 77 88 ntinue Save [Data(LDouble-c] 81 03 14 00 0a	Clear Dele Add Send selected fro Sequence Cy Stop Send a single fro Auto save Ex Lick Hex->Dec) 00 00 00
Normal mode Standard fram 250kbps usal set bps On r ID:0x 00000000 D: 0x 00000000 State e error: 0 it error: 0 Normal ate: Bue-on Bue-on	vevvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv	5000 ection eive eive	Franke 1996 Standard fr Forn Time scale 18:38:48:209 18:38:48:301	ame Data fr ame Data fr nat: Data fram Display recei Frame Type Data frame Data frame Data frame	reme 00000000 rame 00000000 ive only Auto R Frame Format Standard frame Standard frame	0000123 Data 0000123 Data 1efresh Cle Frame ID 00000001 00000001	03 04 05 06 07 03 04 05 06 07 .ta:0x 11 22 3 .ta: 0x 0 11 22 3 .ta: 0x 0 0 0 ata Length 8 8 8	100 100 3 44 55 66 77 88 ntinue Save [Bata(LDouble-c] 81 03 14 00 0a a2 00 00 14 00 a3 00 17 00 2c	Clear Dele Add Send selected fr. Sequence Cy Stop Auto save Ex Lick Hex->Dec) 00 00 00 00 00 00
Normal mode Standard fram ys: 250kbps ual set bps On r D:0x 00000000 State error: e error: 0 it error: 0 ste: Bus-on	v ally send once and receive iet and Start Total: No Dire 1 Rec 2 Rec 3 Rec	5000 ection eive eive eive eive	Frame 1996 Standard fr For Time scale 18:38:48:299 18:38:48:301 18:38:48:305	ame Data fr ame Data fran- Display recei Frame Type Data frame Data frame Data frame Data frame	reme 00000000 rame 00000000 ive only Auto R Frame Format Standard frame Standard frame Standard frame Standard frame	0000123 Data 0000123 Data tefresh Cle Frame ID 00000001 00000001	03 04 05 06 07 03 04 05 06 07 ata:0x 11 22 3 ar Pause Co Data Length 8 8 8 8 2	100 100 3 44 55 66 77 88 ntinue Save [Data(LDouble-ol 81 03 14 00 0a a2 00 10 14 00 a3 00 17 00 2c o4 o8	Clear Dele Add Send selected fri Send selected fri Send selected fri Send a single fri Send a single fri Auto save Ex Lick Hex->Dec) 00 00 00 00 00 00 00 37 00
Normal mode Standard fram 250kbps ual set bps On cd 20 bytes to send r ID:0x 00000000 State serror: 0 it error: Normal Bus-on Monitor year ID	v v v v v v v v v v v v v v	5000 ection eive eive eive	Frank 1996 Standard fr Standard fr [] Time scale 18:38:48:299 18:38:48:303 18:38:48:305	ame Data fr ame Data fran Display received Frame Type Data frame Data frame Data frame	reme 00000000 rame 00000000 rame 00000000 rwe only Auto B Frame Format Standard frame Standard frame	000123 Da 1000123 Da 1000120	03 04 05 06 07 03 04 05 06 07 ata:0x 11 22 3 aar Pause Co Data Length 8 8 8 8	100 100 3 44 55 66 77 88 ntinue Save [Data(LDouble-c] 81 03 14 00 0a a2 00 00 14 00 a3 00 17 00 2c c4 c8	Clear Dele Add Send selected fra Send selected fra Stop Send a single fra Stop Send a single fra Stop Lick Hax=>Dec) 00 00 00 00 00 00 00 00 00 00 00 00 00 00 37 00 Stop
Normal mode Standard fram 250kbps ual set bps On od 20 bytes to send r ID:0x 00000000 Starter On other 00000000 Starter On it error: 0 Monitor ve ID Configure ve ID Configure Add	v v v v v v v v v v v v v v	5000 5000 ection eive eive eive eive	Frame 1996 Standard fr Standard fr [[Time scale 18:38:48:300 18:38:48:305	ame Data fr ame Data fr Display recei Frame Type Data frame Data frame Data frame	reme 00000000 rame 00000000 rame 00000000 reme TID: 0x 0 reme Format Standard frame Standard frame Standard frame	000123 Ba 1000123 Ba 100000001 100000001 100000001	03 04 05 06 07 03 04 05 06 07 ata:0x 11 22 3 mer Pause Co Data Length 8 8 8 2	100 100 3 44 55 66 77 88 ntinue Save [Data(LDouble-c] 81 03 14 00 0a 20 00 01 400 a3 00 17 00 2c c4 c8	Clear Dele Add Send selected fro Sequence Cyn Stop Send a single fro Auto save Ex Lick Hax->Dec) 00 00 00 00 00 00 00 37 00
Normal mode Standard fram 250kbps ual set bps On od 20 bytes to send r ID:0x 00000000 State error: e error: 0 it error: 0 Mormal Bus-on Monitor ve ID Configure tet Add	vev	5000 5000 eotion eive eive eive eive eive	For Even Standard fr Standard fr For Even Time scale 18:38:48:299 18:38:48:303 18:38:48:305	ame Data fr ame Data fr nat: Data fran Display recei Frame Type Data frame Data frame Data frame Data frame	reme 00000000 rame 00000000 ne 00000000 ive only Auto R Frame Format Standard frame Standard frame Standard frame	0000123 Da 10000123 Da 10000123 Da 10000123 Da 100000001 00000001 00000001 00000001	03 04 05 06 07 03 04 05 06 07 Ata:Ox 11 22 3 ear Pause Co Data Length 8 8 8 2	100 100 3 44 55 66 77 88 ntinue Save [Data(LDouble-c] 81 03 14 00 0a a 200 00 14 00 a3 00 17 00 2c o4 c8	Clear Dele Add Send selected fro Sequence Cyn Stop Send a single fro Auto save Ex Lick Hex->Dec) 00 00 00 00 37 00
Wormal mode Standard fram >ps: 250kbps aual set bps On cd 20 bytes to send r ID:0x 00000000 ID: 0x 00000000 State e error: 0 .it error: 0 Monitor ive ID Configure ite Add Receive ID	ve v hly send once and receive tet and Start Total: No Dir. 0 Rec 2 Rec 3 Rec	5000 5000 ection eive eive eive eive	Form 5 1996 Standard fr Form [Time scale 18:38:48:209 18:38:48:303 18:38:48:305	ame Data fr ame Data fr nat: Data fran Display recej Frane Type Data frame Data frame Data frame Data frame	reme 00000000 rame 00000000 ive only Auto R Frame Format Standard frame Standard frame Standard frame	000123 Da 1000123 Da 1000123 Da 1000123 Da 10000001 Cla 00000001 00000001 00000001	03 04 05 06 07 03 04 05 06 07 ata:0x 11 22 3 ear Pause Co Data Length 8 8 8 8 2	100 100 3 44 55 66 77 88 ntinue Save [Data(LDouble-ol 81 03 14 00 0a a2 00 00 14 00 a3 00 17 00 2c o4 o8	Clear Dele Add Send selected fro Sequence Cyo Stop Send a single fro Auto save Ex Liok Hex->Deo) 00 00 00 00 37 00
Normal mode Standard fram Standard fram 250kbps nual set bps On cd 20 bytes to send r ID:0x 00000000 State e error: 0 it error: 0 Mormal Bus-on Monitor Ive ID Configure ite Add Receive ID ID	v ily send once and receive iet and Start Total: No Dir. 0 Rec. 2 Rec. 3 Rec.	5000 ection eive eive eive	For Standard fr Standard fr For [Time scale 18:38:48:299 18:38:48:301 18:38:48:305	ame Data fr ame Data fr mat: Data fran Display recei Frame Type Data frame Data frame Data frame Data frame	reme 00000000 rame 00000000 ive only Auto R Frame Format Standard frame Standard frame Standard frame	000123 0000123 0000123 0000123 0000123 0000001 00000001 00000001 00000001	03 04 05 06 07 03 04 05 06 07 hta: 0x 11 22 3 ear Pause Co Data Length 8 8 8 2	100 100 3 44 55 66 77 88 ntinue Save [Data(LDouble-ol 81 03 14 00 0a a2 00 00 14 00 a3 00 17 00 2c o4 c8	Clear Dele Add Send selected fro Sequence Cy Stop Send a single fro Auto save Ex Lick Hex->Dec) 00 00 00 00 00 00 00 37 00
Normal mode Standard fram Standard fram 250kbps 250kbps 00000000 ID: 0x 00000000 State * error: 0 Normal ate: 0 Monitor ive ID Configure tte Add Receive ID	v ily send once and receive iet and Start Total: No Direc 1 Rec 2 Rec 3 Rec	5000 eotion eive eive eive	Frame 1996 Standard fr Forn [Time scale 18:38:48:299 18:38:48:305 18:38:48:305	ame Data fr ame Data fr Data franc Display recei Frame Type Data frame Data frame Data frame Data frame	reme DO000000 rame 00000000 ive only Auto R Frame Format Standard frame Standard frame Standard frame	000123 000123 000123 000123 0000123 0000001 00000001 00000001	03 04 05 06 07 03 04 05 06 07 .ta:0x 11 22 3 mar Pause Co Data Length 8 8 8 2	100 100 3 44 55 66 77 88 ntinue Save [Data(Llouble-cl 81 03 14 00 0a a2 00 00 14 00 a3 00 17 00 2c o4 c8	Clear Dele Add Send selected fr Sequence Cy Stop Send a single fr Auto save Ex lick Hex->Dec) 00 00 00 00 37 00
Normal mode Standard fram Standard fram 250kbps ual set bps 1250kbps on 110:0x 000000000 Standard fram it error: 0 Monitor ve ID Configure te Add	v ily send once and receive iet and Start Total: No Dir 0 Receive 2 Rec 3 Rec	5000 eotion eive eive eive	Frank 1996 Standard fr Standard fr For Time scale 18:38:48:299 18:38:48:301 18:38:48:305	ame Data fr ame Data fr nat: Data fran Display received Frane Type Data frame Data frame Data frame	reme 00000000 rame 00000000 ive only Auto R Frame Format Standard frame Standard frame Standard frame	000123 Da 1000123 Da 1000123 Da 1000123 Da 10000001 00000001 00000001 00000001	03 04 05 06 07 03 04 05 06 07 ta:0x 11 22 3 ta: 0x Co Data Length 8 8 8 8 8 8 8 8	100 100 3 44 55 66 77 88 ntinue Save [Bata(LDouble-c] 81 03 14 00 0a a2 00 10 14 00 a3 00 17 00 2c c4 c8	Clear Dele Add Send selected fro Sequence Cy. Stop Send a single fro Auto save Ex Lick Hex->Dec) 00 00 00 00 00 00 00 37 00

The conversion principle of data from CAN side to ModBus RTU is the same as the above, after the CAN side receives the above four messages, the converter will combine the received CAN messages into a frame of RTU data according to the CAN segmentation mechanism mentioned above, and add CRC checksum at the end.

