3onedata
Make network communication more reliable

CAN232/485 Interface Converter

User Manual

Document Version: 04

Issue Date: 4/2/2019

Industrial Ethernet communication solutions experts

3onedata Co., Ltd.

Trademark statement

3onedata, 3onedata and 3One data are the registered trademark owned by 3onedata Co., Ltd. And other trademarks mentioned in this manual belong to their corresponding companies.

Notes

Purchased product, service or features should be constrained by 3ondedata commercial contracts and clauses. The whole or part product, service or features described in this document may beyond purchasing or using range. 3onedata won't make any statement or warranty for this document content unless any other appointment exists.

Due to product version upgrading or other reason, this document content will be upgraded aperiodically. Unless other appointment exists, this document only for usage guide, all statement, information and suggestion in this document won't constitute any warranty.

3onedata
Make network communication more reliable

3onedata Co., Ltd.

Headquarter address:	3/B, Zone 1, Baiwangxin High Technology Industrial Park, Song Bai Road, Nanshan District, Shenzhen, 518108, China
Technology support:	tech-support@3onedata.com
Service hotline:	+86-400-880-4496
E-mail:	sales@3onedata.com
Fax:	+86-0755-26703485
Website:	http://www.3onedata.com

Preface

The user manual of CAN interface converter has introduced this device:

- Product feature
- Network management method
- Network management relative principle overview

Readers

This manual mainly suits for engineers as follows:

- Network administrator responsible for network configuration and maintenance
- On-site technical support and maintenance staff
- Hardware engineer

Text Format Convention

Format	Description
“”	Words with "" represent the interface words. e.g.: "The port number".
>	Multi-level paths are separated by ">". Such as opening the local connection path description: Open "Control Panel> Network Connection> Local Area Connection".
Light Blue Font	Represent the words click to achieve hyperlink. The font color is as follows: ‘Light Blue’.
About this chapter	The section ‘about this chapter’ provides links to various sections of this chapter, as well as links to the Principles Operations Section of this chapter.

Icon Convention

Format	Description
 Notice	Remind the announcements in the operation, improper operation may result in data loss or equipment damage.
 Warning	Pay attention to the notes on the mark, improper operation may cause personal injury.

	Conduct a necessary supplements and explanations for the description of operation content.
	Configuration, operation, or tips for device usage.
	Pay attention to the operation or information to ensure success device configuration or normal working.

Port Convention

The port number in this manual is only an example, and does not represent the actual port with this number on the device. In actual use, the port number existing on the device shall prevail.

Revision Record

Version NO.	Revision Date	Revision Description
01	2013-7	Product release
02	5/23/2014	Software update, add data conversion mode
03	12/18/2018	Template update
04	4/2/2019	Combine CAN232 and optimize the manual

Content

1 Configuration Description

Because CAN-bus and RS-232/485 serial port have many communication parameters, CAN232/485 converter also opens most of its parameters, so user can set it on their own to meet the needs of actual applications. CAN232/485 converter configuration includes converter conversion method, serial port parameter and CAN-bus parameter. Parameter configuration is implemented via specified configuration software without hardware jumper configuration.

Before regular use, the conversion parameter of CAN232/485 converter needs to be configured in advance; without configuration, CAN232/485 converter would implement the parameter configured last time(if there are no configuration, the converter would implement the default configuration parameter).

1.1 Configuration Method

In order to make converter enter configuration mode, it has a special configuration switch—DIP switch Pin 2 besides RS-232/485 interface. Turn the Pin 2 upward, the converter is powered on and enter “configuration” mode; turn the Pin 2 downward, the converter is powered on and enter “normal operation” mode. The steps of entering configuration are as follows:

Step 1 Turn the Pin 2 of the converter DIP switch upward then power on it;

Step 2 CAN485 uses RS-485 to RS-232 converter (which can work normally at 19200bps baud rate) to connect converter and computer; CAN232 uses serial line to connect converter and computer;

Step 3 Open serial port configuration software, connect the device, open serial port and set parameters.

1.2 Software Description

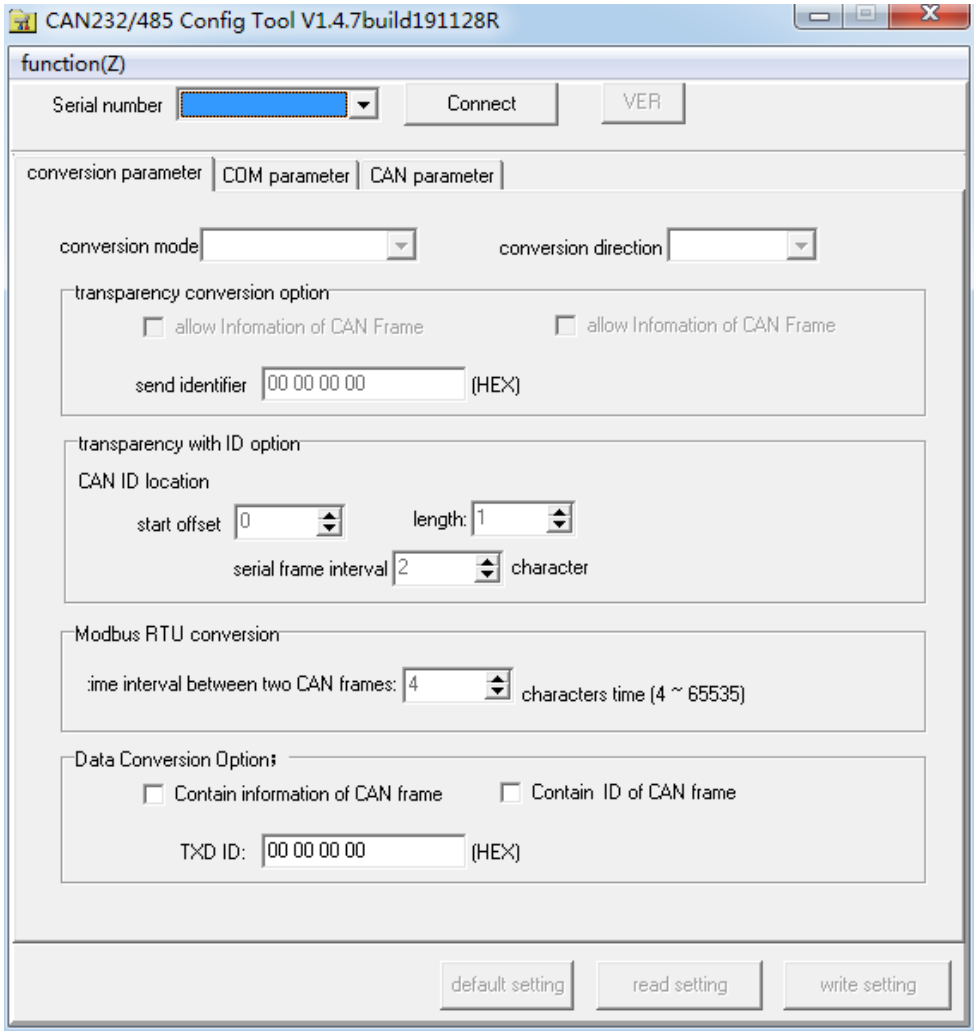


Figure 1.1 configuration interface

The name of configuration software of CAN232/485 converter is “serial port configuration tool”, and the setup software is included in the matching CD of the product. The software interface is shown in Figure 1.1. The setup software can memorize and display the parameters set by CAN232/485 converter last time (which won’t be saved if not set successfully) to prevent user from forgetting his/her own configuration. It could also be restored to default parameters with one click and has the function of reading the current parameters of CAN232/485.

The parameter can be set via this software only when the converter is in configuration mode, otherwise the software would assume that the converter is unconnected; After selecting a conversion mode, the software would open the parameters related to this conversion mode and set the parameters unrelated to itself to unavailable to avoid

wrong setup. Here refer to the detailed description of the meaning of major configuration parameter conducted by the configuration software.

1.2.1 Parameter Conversion

Conversion Parameter refers to the parameters of converter conversion rules and directions etc. The parameter conversion interface is shown in Figure 1.1.

Conversion mode: it includes four optional conversion modes: transparent conversion, transparent conversion with tag, Modbus protocol conversion and data conversion(refer to “Application description” for each mode’s specific function).

Conversion direction:

- Bidirectional: the converter converts the data of serial bus to CAN bus, and the data of CAN bus to serial bus.
- Serial to CAN only: only convert the data of serial bus to CAN bus, rather than converting the data of CAN bus to serial bus.
- CAN to Serial only: only convert the data of CAN bus to serial bus, rather than converting the data of serial bus to CAN bus.

By choosing the conversion direction, data interference on the bus side that needs no conversion could be eliminated.

Allow CAN frame information to be forwarded to serial frame:

This parameter is only used in transparent conversion mode. When it is selected, the converter would add the frame information of CAN message to the first byte of serial frame. It doesn’t convert the frame information of CAN when it is not selected.

Allow CAN frame ID to be forwarded to serial frame:

This parameter is only used in transparent conversion mode. When it is selected, the converter would add the frame ID of CAN message in the front of frame data of serial frame and in the back of frame information (if frame information conversion is allowed). It doesn’t convert the frame ID of CAN when it is not selected.

Send identifier:

Send frame identification: only available in “transparent conversion” mode, it means when serial data is converted into CAN message, the frame of CAN message identifies the value (hexadecimal data) of domain (frame ID); Note that it is invalid in “transparent with tag” conversion mode, because the currently sent identifier (frame ID) is filled with data from the serial port above.

Note

- This identifier (frame ID) is an actual CAN message ID value (the order of bytes from left to right is from the high byte to the low byte), if the value is "00 00 00 0A" (HEX), it means the sent frame ID number is 10.
- This item is also related to "frame type" when used effectively. If "standard frame" is selected, then frame identifier is 11-bit valid, and the value range is 0~7FF (HEX). It would be invalid when it is out of range, and only takes lower 11 bits.
- If "extended frame" is selected, then frame identifier is 29-bit valid, and the value range is 0~1F FF FF FF (HEX). It would be invalid when it is out of range, and only takes lower 29 bits.

The position of CAN frame identification in serial frame:

The parameter is only used in "transparent conversion with tag" mode. The offset address of starting byte of CAN message frame ID in serial frame and the length of frame ID when serial data is converted into CAN message (refer to "transparent conversion with tag").

Note

Frame ID length could be filled with 1~2 bytes which correspond to ID1 and ID2 of CAN message when it is standard frame. When it is extended frame, it could be filled with 1~4 bytes, ID1, ID2, ID3 and ID4. The ID is 11 bits when it is standard frame, which means ID1.2~ID1.0 and ID2.7~ID2.0; and ID2.7~ID2.3 is reserved (filled with 0)

The corresponding relationship of the two is as shown in the table below:

The frame ID format	ID1.7~ID1.3	ID1.2~ID1.0	ID2.7~ID2.0
The actual value of frame ID	Reserve(00000)	ID.10~ID.08	ID.07~ID.00

The ID is 29 bits when it is extended frame, the higher 3 bits with ID1 is reserved (filled with 0). The corresponding relationship of the two is as shown in the table below:

ID1.4~ID1.0	ID1.7~ID1.5	ID1.4~ID1.0	ID2.7~ID2.0	ID3.7~ID3.0	ID4.7~ID4.0
The actual value of frame ID	Reserve(000)	ID.28~ID.24	ID.23~ID.16	ID.15~ID.08	ID.07~ID.00

The time interval between serial frames which is n character :

Only used in "transparent conversion with tag" mode. The minimum interval between two serial frames when user is sending serial frame to converter. This interval uses

"time of transmitting single character" as unit. Here n can be set to optional time of 2~10 characters.

Note

"The number of serial frame interval character" can only be set in "transparent conversion with tag" mode. The actual interval of user frame must be consistent with the setting, otherwise it might cause incomplete frame conversion. The meaning of "the time of transmitting single character": the time the serial port needs to transmit a character (10 bits) at the same baud rate, which is 10 divided by the corresponding baud rate.
for example: at baud rate of 9600bps, "the character number of serial frame interval" is 4, "the time of transmitting single character (each character is 10 bits)" is (10/9600)s, and the received actual interval of serial frame is (10/9600) *4 =4.17 (ms), that is the interval between two serial frames is at least 4.17ms.

Contains CAN information:

This parameter is only used in data conversion mode. When it is selected, the converter would add the frame information of CAN message to the first byte of serial frame. It doesn't convert the frame information of CAN when it is not selected.

Including CAN Frame identification:

This parameter is only used in "Data Conversion" mode. When it is selected, the converter would add the frame ID of CAN message in the front of frame data of serial frame and in the back of frame information (if frame information conversion is allowed). It doesn't convert the frame ID of CAN when it is not selected.

1.2.2 Serial Port Parameter

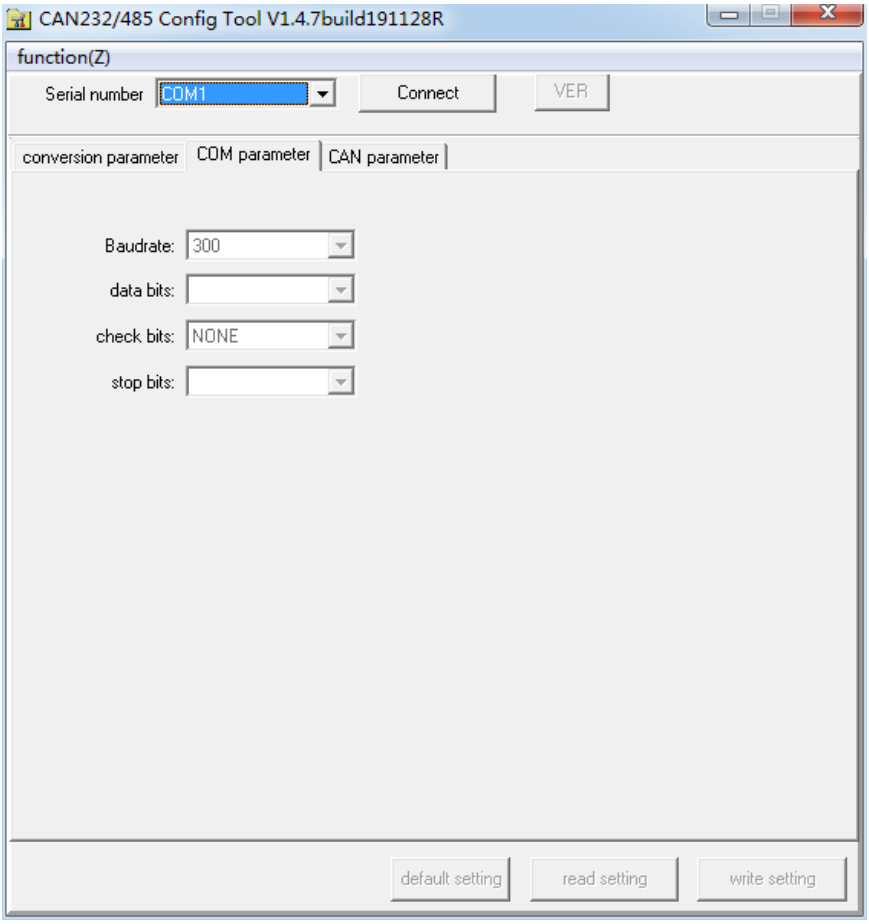


Figure1.2 Serial port parameter interface

Baud rate: the baud rate of serial port is optional between 300bps and 115200bps.

Serial port parity mode: six optional modes : no parity, odd parity, even parity, mark parity, space parity and 2 stop bit.

Note

Whatever parity modes, the serial port transmission data bit of the converter is 8 bits. No parity only adds 1 start bit and 1 stop bit (a total of 10 bits), the other 4 modes add 1 parity bit or stop bit (a total of 11 bits).

1.2.3 CAN Parameter

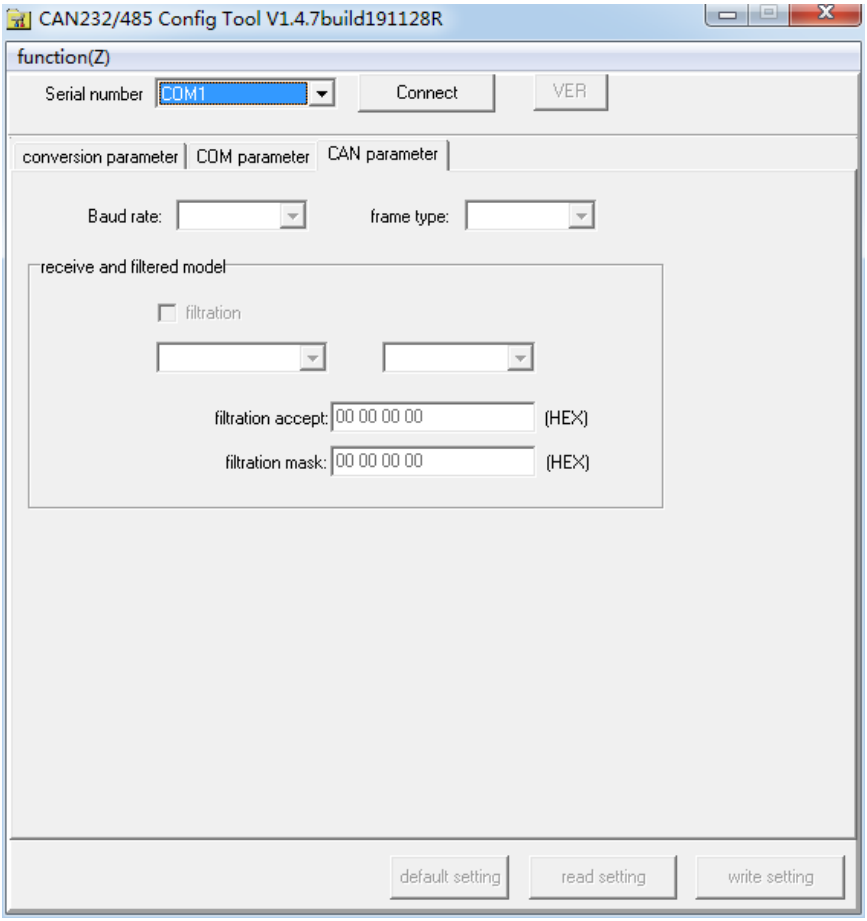


Figure1.3 CAN parameter interface

Baud rate: CAN bus baud rate, here are the standard baud rates recommended by CIA.

Frame type: the frame type of CAN message during conversion has two options, standard frame and extended frame. Remote frame is not supported.

Receiving filter mode: the mode of converter filtering the message on CAN bus side when it receives.

Filter options: valid filter or invalid filter

Received frame options: receives standard frame or extended frame only

ID options: distinguish ID or not (ID is related to identifier receiving)

Filter acceptance code (HEX):

The format of data filling is hexadecimal, and each 8-bit byte is separated by “Space” character.

If it is “standard frame”, then the frame ID only has 11 bits, so the maximum value of “filter acceptance code” is “0x07 FF”, beyond which is invalid. Software only takes the lower 11 bits.

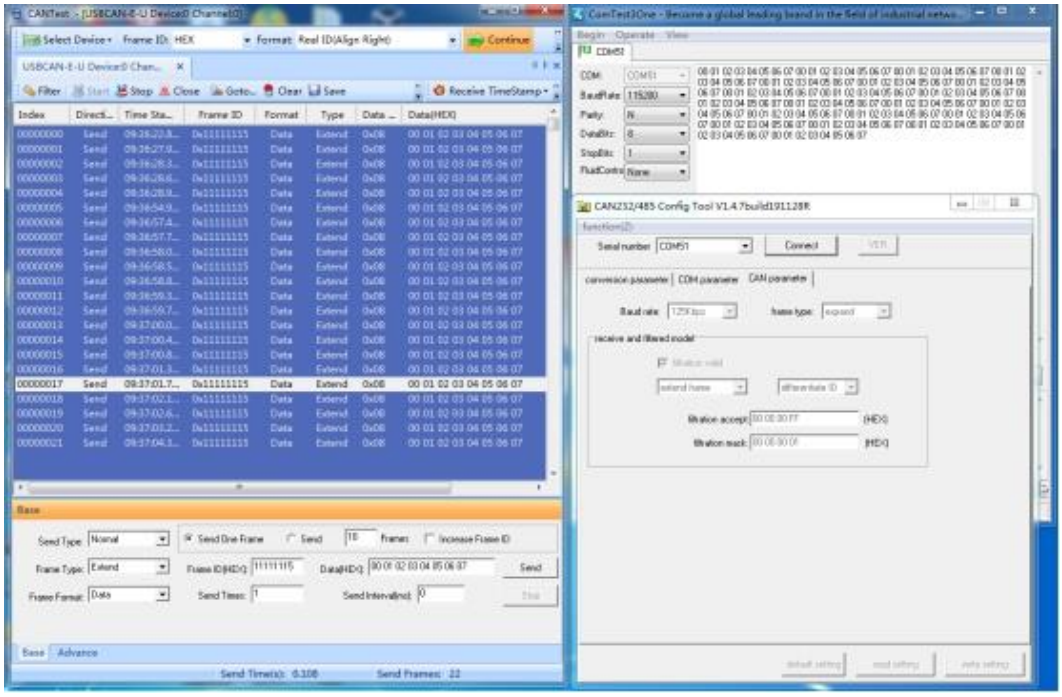
If it is “extended frame”, then the frame ID has 29 bits, so the maximum value of “filter acceptance code” is “0x1F FF FF FF”, beyond which is invalid. Software only takes the lower 29 bits.

Filter mask code (HEX):

The format of data filling is hexadecimal, and each 8-bit byte is separated by “Space” character.

- If it is “standard frame”, then the frame ID only has 11 bits, so the maximum value of “filter acceptance code” is “0x07 FF”, beyond which is invalid. Software only takes the lower 11 bits.
- If it is “extended frame”, then the frame ID has 29 bits, so the maximum value of “filter acceptance code” is “0x1F FF FF FF”, beyond which is invalid. Software only takes the lower 29 bits.
- If the bitwise AND value of the filter acceptance code and the filter mask code is equal to the bitwise AND value of the frame ID and the filter mask code, the frame is received. Otherwise not accepted.

As shown in Figure 1.4, set filter mode to receive extended frame only, distinguish ID and filter acceptance code is 00 00 00 FF, filter mask code is 00 00 00 01, and received frame ID is 11 11 11 15.



Set filter mode to receive extended frame only, distinguish ID and filter acceptance code is 00 00 00 FF, filter mask code is 00 00 FF 00, and received frame ID is 00 00 00 15.

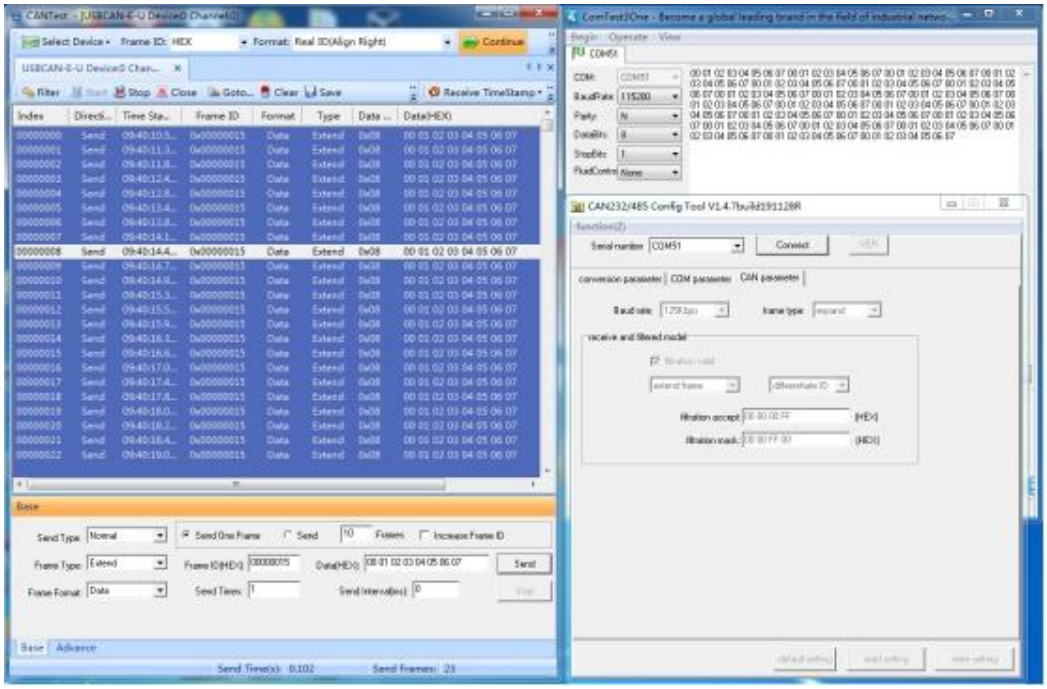


Figure 1.4 receiving filter mode

1.2.4 Key Description

Default configuration: the parameter could be restored to factory defaults.

Read configuration: read the current parameters of the converter and display them on the panel.

Write configuration: when the parameters are set, click the button to write the configuration parameters to the converter.

2 Application Description

2.1 Transparent Conversion

In transparent conversion, the converter would convert the data and send it to the other side of the bus when it receives data on one side of the bus. This processing in the form of data stream maximizes the speed of the converter and the utilization of the buffer, because the converter converts and sends at the same time as it receives, thus freeing up the buffer to receive.

2.1.1 Frame Format

Serial bus frame: it can be data stream, as well as data with protocol.
Communication format: 1 start bit, 8 data bits, and 1 stop bit.
CAN bus frame: the format of CAN message frame remains unchanged.

2.1.2 Conversion Mode

Serial Frame to CAN Message:

All serial frame data are filled in the data field of CAN message frame in order. Once the converter detects data on serial bus, it would receive and convert it immediately. The converted CAN message frame information (the frame type part) and frame ID come from user's previous configuration, and the frame type and frame ID remain unchanged during the conversion. The corresponding format of data conversion is shown in Figure 2.1.

If the length of received serial frame is less than or equal to 8 bytes, it would fill character 1 to n (n is the length of serial frame) in the 1 to n bytes of the data field of CAN message.

If the number of serial frame byte is larger than 8, the processor starts from the first character of the serial frame and fills the data field of CAN message with 8 characters in order for the first time. After the data is sent to the CAN bus, the remaining serial frame data is converted and then filled into the data field of the CAN message until the data is converted.

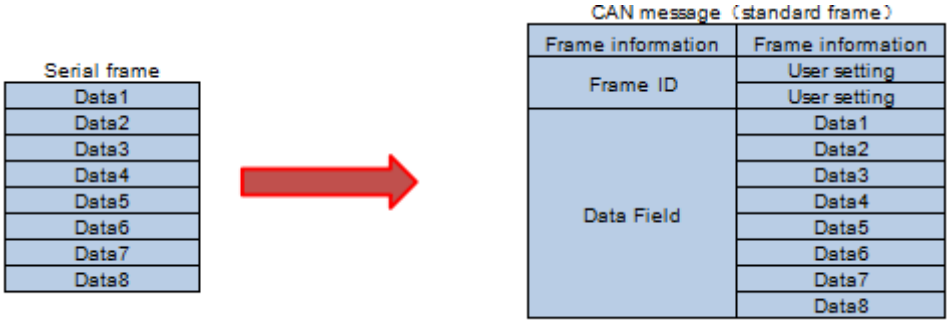


Figure2.1 Serial Frame to CAN Message (transparent Conversion)

CAN Message to Serial Frame:

CAN bus message is also received a frame immediately forward a frame. The corresponding data format is shown in the figure. Convert the data in the data field of CAN message to serial frame in order during conversion. If the “allows CAN frame information to be sent to serial frame” item is checked during configuration, the converter would fill “frame information” byte of CAN message in serial frame directly. If the “allows CAN frame ID to be sent to serial frame” item is checked during configuration, all “frame ID” byte of CAN message would be filled in serial frame.

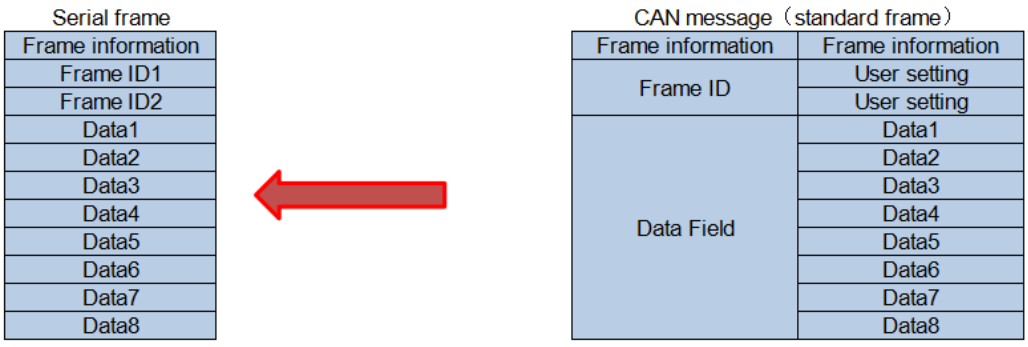


Figure 2.2 Convert CAN message into serial frame (transparent conversion)

2.1.3 Conversion Example

Serial Frame to CAN Message

If the conversion of CAN message frame information is configure to “standard frame”, frame ID1, ID2 are “00, 60” respectively, the conversion format is shown in Figure 2.3.

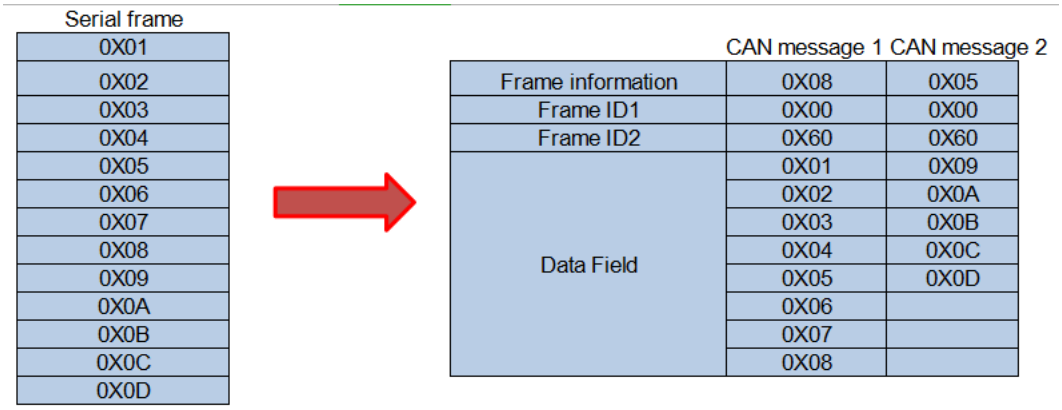


Figure2.3 Serial Frame to CAN Message (transparent Conversion)

2. CAN Message to Serial Frame:

Assume that CAN message has checked “allows CAN frame information to be forwarded to serial frame”, “allows CAN frame ID to be forwarded to serial frame”. CAN message and the converted serial frame are shown in Figure 2.2.

Note:

Whether or not to convert“frame information” and “frame ID” is configured in the “options for transparent conversion” in the “serial port configuration tool”.

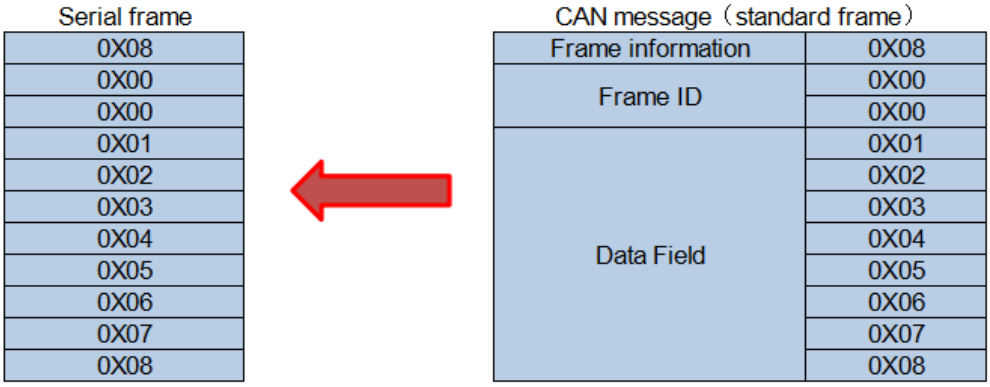


Figure 2.4 Convert CAN message into serial frame (transparent conversion)

2.2 Transparent Conversion with Tag

Transparent conversion with tag is a special usage of transparent conversion. It helps user in building their own networks via converter more conviniently and using custom application protocol.

This mode converts the address information in serial frame into the frame ID of CAN bus automatically. As long as the converter is informed of the address in the start position and length of the serial frame in the configuration, it extracts the frame ID and fills it in the frame ID field of the CAN message during the conversion, serving as the ID of the CAN message when the serial frame is forwarded. When CAN message is converted into serial frame, its ID is also converted in the corresponding position of serial frame.

Note that in this conversion mode, the "send identifier" for the "CAN parameter" item of the configuration software is invalid because the currently sent identifier (frame ID) is filled with data from the serial frame above.

2.2.1 Frame Format

Serial bus frame: when it is converted with tag, the complete serial data frame must be obtained. The converter uses the interval between the two frames to divide frames. And this interval could be set by user. The maximum length of serial frame is the length of buffer zone: 255 bytes.

The first data detected by the converter in the idle state of serial bus is regard as the first character of receiving frame. The interval between the character of this frame during transmission must be less than or equal to the time (the time of transmitting one character is calculated by using the number of bits in this character to divide the corresponding baud rate) of transmitting n characters (the value of n is configured by upper computer in advance). If the converter has not received any character after receiving one character in less than or equal to the transmission time of n characters, it would assume that this frame finishes transmitting and regards this character as the last character of this frame; Characters after n character times do not belong to that frame, but to the contents of the next frame.

CAN bus frame: the format of CAN message remains unchanged, but the corresponding frame ID of CAN would be converted to the serial frame.

2.2.2 Conversion Mode

Serial Frame to CAN Message

The start address and length of CAN identification brought by serial frame can be set by configuration. The range of start address is 0~7, and the range of length is 1~2 (standard frame) or 1~4 (extended frame).

During conversion, all CAN frame IDs in serial frames are converted to the frame ID field of CAN message according to the prior configuration (if the number of frame ID is less than the number of CAN message, then the filling order of CAN message is frame ID1~ID4, and the remaining ID is filled in as 0). Other data are converted in order, as shown in figure 2.5.

If a CAN message fails to convert the serial frame data, the same ID is still used as the CAN message's frame ID to continue the conversion until the serial frame conversion is completed.

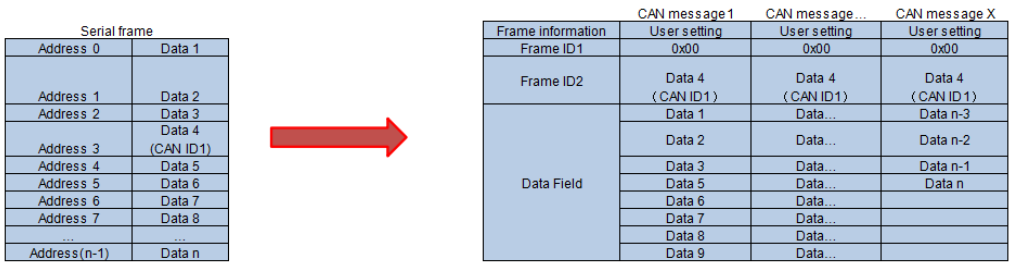


Figure2.5 Serial Frame to CAN Message (Transparent Conversion with Tag)

CAN Message to Serial Frame:

For CAN message, one frame is forwarded immediately once it is received, and the ID in the received CAN message is converted according to the position and length of the CAN frame ID configured in advance in the serial frame. Other data is forwarded in order as shown in Figure 2.6.

It's worth noting that the frame format (standard frame or extended frame) should conform to the frame format demands of previous configuration in application no matter it is serial frame or CAN message, otherwise it might cause unsuccessful communication.

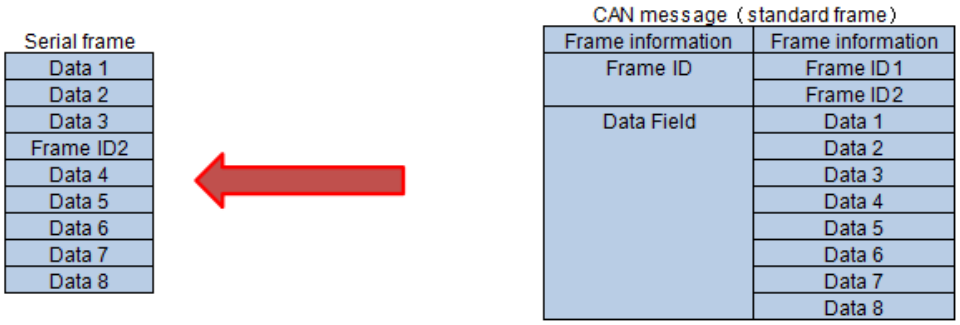


Figure 2.6 Convert CAN message into serial frame (Transparent Conversion with Tag)

2.2.3 Conversion Example

Serial Frame to CAN Message

Assume the position of CAN ID in serial frame, the start offset is 2 and the length is 3 (in extended frame), the result of serial frame and converted CAN message is shown in Figure 2.7. Among them, two frames of CAN message use the same ID for conversion.

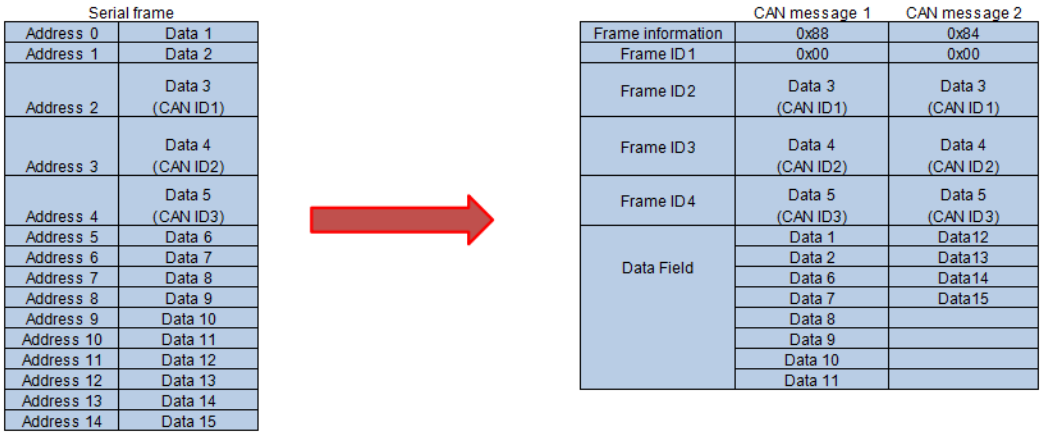


Figure2.7 Serial Frame to CAN Message (Transparent Conversion with Tag)

CAN Message to Serial Frame

Assume the position of CAN ID in serial frame, the start offset is 2 and the length is 3 (in extended frame), the result of CAN message and converted serial frame is shown in Figure 2.8.

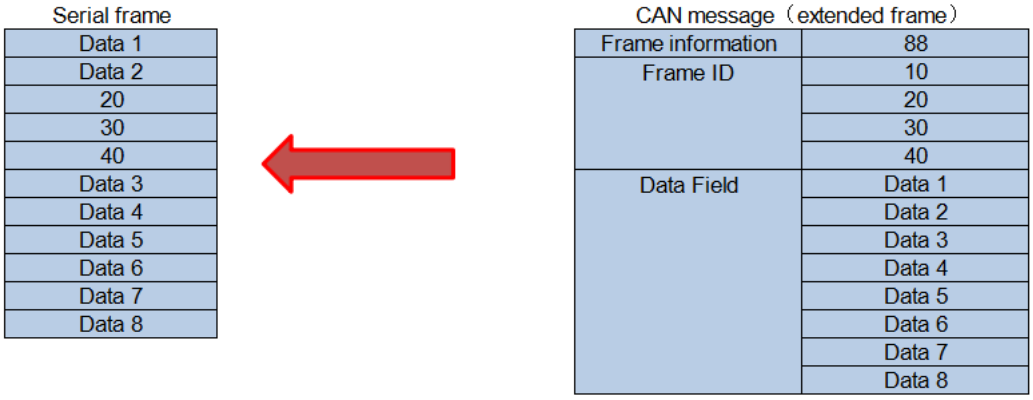


Figure 2.8 Convert CAN message into serial frame (Transparent Conversion with Tag)

2.3 ModbusRTU Conversion

Modbus protocol is a standard application layer protocol, which is widely used in various industrial control sites. The protocol is open, and has high real time and great communication authentication mechanism, which greatly applies to sites demanding high reliability in communication.

The converter uses standard Modbus RTU protocol format at the serial port side, so the converter not only supports Modbus RTU protocol used by user, but also connects with other devices that support Modbus RTU protocol directly.

At the CAN side, a easy-to-use segmented communication format is made to realize the communication of Modbus. The converter's role in it is protocol authentication and forwarding. It supports Modbus protocol transmission instead of master device or slave device of Modbus. User can communicate according to Modbus protocol.

2.3.1 Frame Format

Serial bus frame: the serial interface uses standard Modbus RTU protocol, so it would be good if the user frame conforms to this protocol (refer to the appendix: Modbus protocol introduction). If the transmitted frame doesn't conform to Modbus RTU format, the converter would discard the received frames instead of converting them.

The Modbus RTU transmission format adopted by the converter is 1 start bit, 8 data bits, and 1 stop bit.

The maximum length of Modbus RTU frame is the length of buffer zone: 255 bytes.

CAN bus frame: if the device at the CAN side adopts Modbus protocol, it needs to define a reliable transmission format for it. Here it adopts a segmented protocol to implement it, which defined a method to segment and regroup information longer than 8 bytes.

The development of segmented protocol has referred to the transmission protocol of segmented message in DeviceNet. Segmented packet format such as table 2.1 (take extended frame as an example, the standard frames are just different in the length of the frame ID, and other formats are the same), and the transmitted Modbus protocol content can begin from "Data 2" bytes. If the protocol content is larger than 7 bytes, so the rest of the protocol would continue to convert in accordance with this segmented format until the conversion finishes.

The description of CAN bus frame format is as follows:

Table 2.1 CAN 2.0 extended frame format

	7	6	5	4	3	2	1	0
Frame Information	FF	RTR	x	x	DLC (data length)			
Frame ID1	x	x	x	ID.28—ID.24				
Frame ID2	ID.23—ID.16							
Frame ID3	ID.15—ID.8							
Frame ID4	ID.7—ID.0							
Data1	Segmented Identification		Segmented counter					
Data2	Character 1							
Data3	Character 2							
Data4	Character 3							
Data5	Character 4							
Data6	Character 5							
Data7	Character 6							
Data8	Character 7							

- Segmented message identification: indicate whether this message is the segmented message. It represents individual message when it is 0, a frame that belongs to the segmented message when it is 1.

- Segment type: to indicate whether it is not segmented,the first segment, the middle segment or the last segment. The value defines as shown in Table 2.2.

Table 2.2 segment type position

Position	Definition	Note
00	Not segmented	It indicates it is not segmented
01	The first one is not segmented	If the segment counter contains value 0, then it is the first segment of this segment series
10	The middle segment	It indicates it is a middle segment
11	The last segment	It indicates it is the last middle segment

- Segment counter: The symbol of each segment, the serial number of the segment in the entire message, and the value of the counter is the number of the segment. This could authenticate whether there are lost segment during receiving, and the segment counting range is 1-63(bit0~5).

2.3.2 Conversion Mode

During the the serial port side to CAN side conversion, the converter will only convert when it receives a complete and correct Modbus RTU frame, otherwise there is no action.

As shown in Figure 2.9, the address domain of Modbus RTU protocol is converted into ID4 (extended frame) ID2 (standard frame) of the frame ID in CAN message, and the identification remains unchanged during the conversion process.

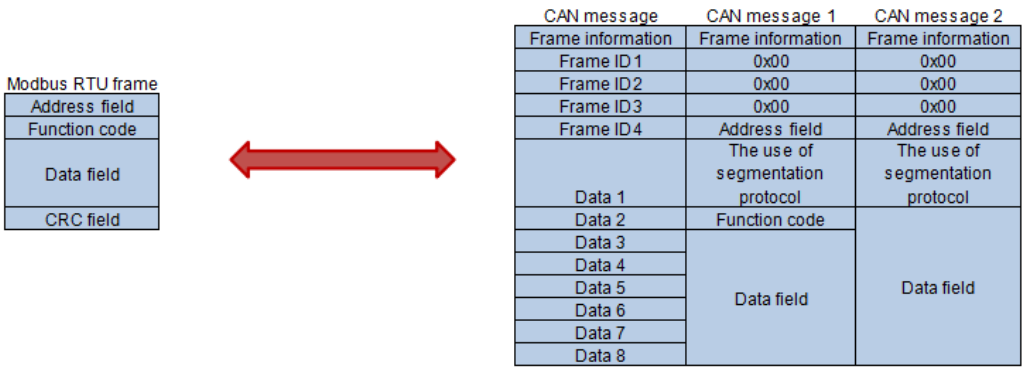


Figure 2.9 Communication frame conversion format (ModbusRTU conversion)

CRC check bytes are not converted into CAN messages, and CAN messages needn't to take serial frame check bytes, because CAN bus itself has a good check mechanism.

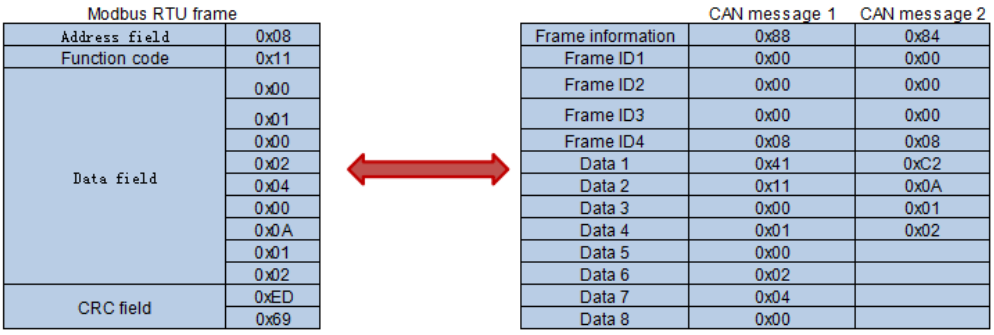
The protocol contents of Modbus RTU are converted -- function code and data field, which are converted into the data field of CAN message frame (starting from the second data byte, the first data byte is used for segmented protocol) in turn. The length of Modbus RTU frame varies according to the function code. However, a CAN message CAN only transmit 7 data in one frame, so the converter will convert the longer Modbus RTU frames into CAN messages and send them out with the above CAN segmentation protocol. User takes function code and data field for processing when receiving on CAN node.

For Modbus protocol data of CAN bus, there is no need to implement cyclic redundancy check (CRC16). The converter receives the data in accordance with the segmented protocol, and automatically adds cyclic redundancy check (CRC16) after receiving the data, and converts it into Modbus RTU frame and sends it to the serial bus. If the received data doesn't conform to segmented protocol, it would discard this group of data instead of converting it.

2.3.3 Conversion Example

When it is configured as an extended frame, as shown in figure 2.10, the Modbus RTU frame is converted into a CAN message, the address 0x08 is directly filled into frame ID4, and the other frame IDs are filled into 0x00, and the frame ID is kept unchanged during the conversion process.

When one CAN message cannot process one Modbus message, CAN message adopts segment protocol. The "data 1" of each CAN message is used to fill the segment information (0x41, 0xC2), which is not converted to the Modbus RTU frame and is only used as the frame format to confirm the frame information. The values of function code and data field are successively filled into data 2~8 of CAN message.



Modbus RTU frame	
Address field	0x08
Function code	0x11
Data field	0x00
	0x01
	0x00
	0x02
	0x04
	0x00
	0x0A
	0x01
	0x02
CRC field	0xED
	0x69

	CAN message 1	CAN message 2
Frame information	0x88	0x84
Frame ID1	0x00	0x00
Frame ID2	0x00	0x00
Frame ID3	0x00	0x00
Frame ID4	0x08	0x08
Data 1	0x41	0xC2
Data 2	0x11	0x0A
Data 3	0x00	0x01
Data 4	0x01	0x02
Data 5	0x00	
Data 6	0x02	
Data 7	0x04	
Data 8	0x00	

Figure 2.10 example of communication frame interconversion format (ModbusRTU conversion)

2.4 Data Conversion

Data conversion is a special use of transparent conversion, which satisfies the one-to-one data conversion between serial frames and CAN frames to the greatest extent, and enables the complete conversion between serial frames and CAN frames.

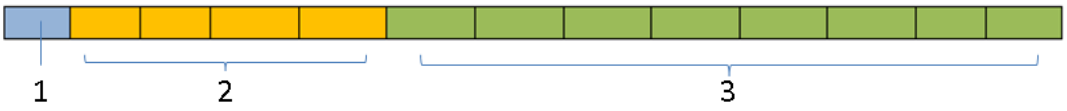
2.4.1 Frame Format

CAN Data Transformation Format

One TCP or UDP frame contains several CAN frames, maximum 50 frames, minimum 1 CAN frame.

CAN Frame	CAN Frame	CAN Frame	CAN Frame	...	CAN Frame

One CAN frame contains 13 bytes, as the picture bellow:



Frame Structure of the Frame Information

In the CAN frame structure picture, "1" represents the frame information: The length is 1 byte, and it's used for identifying some information of the CAN frame, such as type, length and so on.

Frame structure of the frame information as follow:

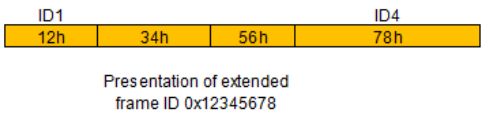
Bit7 Bit0

FF	RTR	Reserved	Reserved	D3	D2	D1	D0

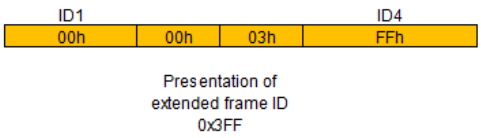
- FF: Identification of standard frames and extend frame, 1 is extend frame, and 0 is standard frame.
- RTR: Identification of remote frame and data frame, 1 is remote frame, 0 is data frame.
- Reserved value is 0, don't enter 1.
- D3~D0: Identify the data length of CAN frame.

Frame structure description of the frame ID

In the CAN frame structure picture, "1" represents the frame ID: the length is 4 bytes, the valid bit of standard frame is 11 bits, and the valid bit of extend frame is 29 bits. Frame structure of the frame ID as follow:



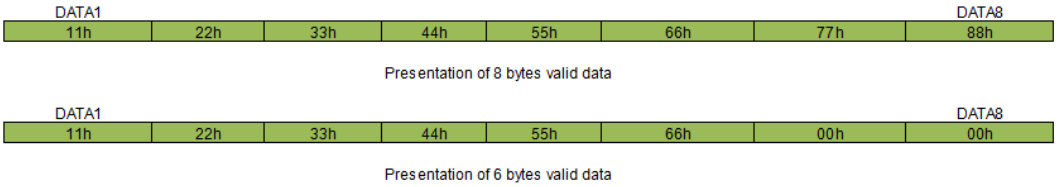
Presentation of extended frame ID 0x12345678



Presentation of extended frame ID 0x3FF

Frame structure description of the frame data

In the CAN frame structure picture, "3" represents frame data: the length is 8bytes, and the valid length depends on the D3~D0 value of the frame information.



Following example is an extended data frame, ID is 0X12345678, and it includes the frame expression of 8 bytes data (11h, 22h, 33h, 44h, 55h, 66h, 77h, 88h):

88h	12h	34h	56h	78h	11h	22h	33h	44h	55h	66h	77h	88h

Following example is a standard data frame, ID is 0X3FF, and it includes the frame expression of 6 bytes data (11h, 22h, 33h, 44h, 55h, 66h):

06h	00h	00h	03h	FFh	11h	22h	33h	44h	55h	66h	00h	00h

2.4.2 Conversion Mode

Serial Frame to CAN Message

All serial frame data are filled in the data field of CAN message frame in order. Once the converter detects data on serial bus, it would receive and convert it immediately. The converted CAN message frame information (the frame type part) and frame ID come from user's previous configuration, and the frame type and frame ID remain unchanged during the conversion.

If the length of received serial frame is less than or equal to 8 bytes, it would fill character 1 to n (n is the length of serial frame) in the 1 to n bytes of the data field of CAN message.

If the number of bytes in a serial frame is greater than 8, it can't meet the conversion requirements and can't convert.

Whether"CAN frame information" and "CAN frame ID" are included in serial frame, it can be configured in the "options for data conversion" in the "serial port configuration tool"：

- When "contains CAN frame information" and "contains CAN frame ID"are checked, then the sent serial frame includes frame information and frame ID, the number of bytes is 13 bits;
- When "contains CAN frame information" and "contains CAN frame ID"are not checked, then the sent serial frame doesn't include frame information and frame ID, the number of bytes is 8 bits; At present, the send identifier is frame ID, which can be configured via "serial port configuration tool".

Assuming that "contains CAN frame information" and "contains CAN frame ID" are not checked in the data conversion options, the serial frame sent by the user is 8 bytes, and the frame ID in the CAN message is entered in the "send identifier" :

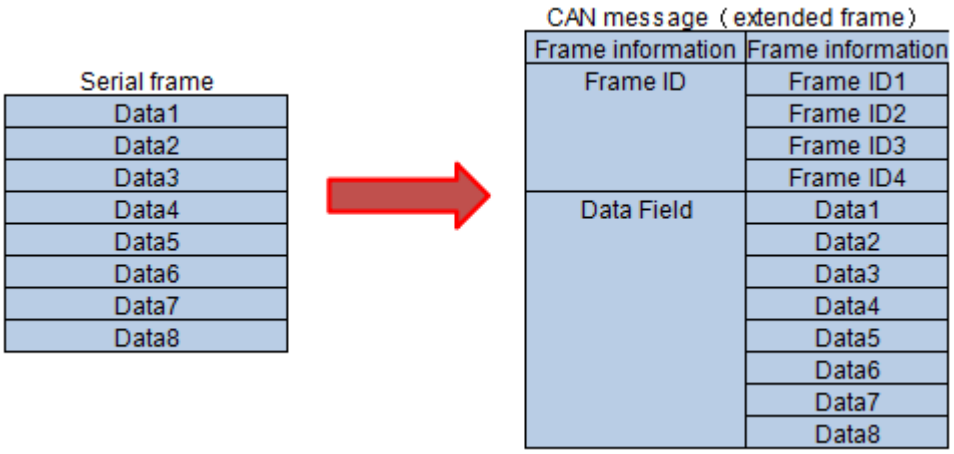


Figure 2.11 Serial Frame to CAN Message (Data Conversion)

CAN Message to Serial Frame:

CAN bus message is also received a frame and immediately forward a frame. The corresponding data format is shown in the Figure 2.12.

Convert the data in the data field of CAN message to serial frame in order during conversion.

Assuming check “contains CAN frame information” and “contains CAN frame ID”, the frame information of CAN message and frame ID would be filled to serial frame completely.

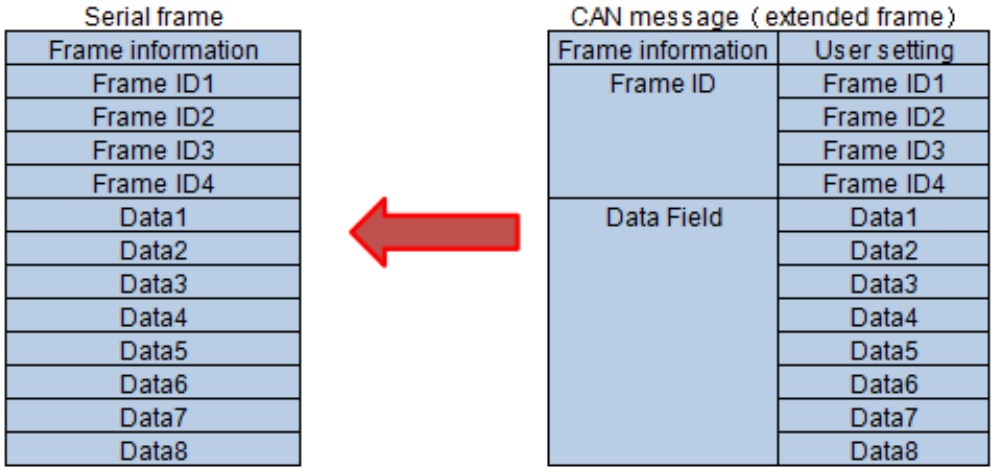


Figure 2.12 Convert CAN message into serial frame (Data Conversion)

2.4.3 Conversion Example

Serial Frame to CAN Message

Assuming the configuration has not checked “contains CAN frame information” and “contains CAN frame identification”, the identifier is “11 22 33 44”, the conversion format would be as shown in the Figure 2.13.

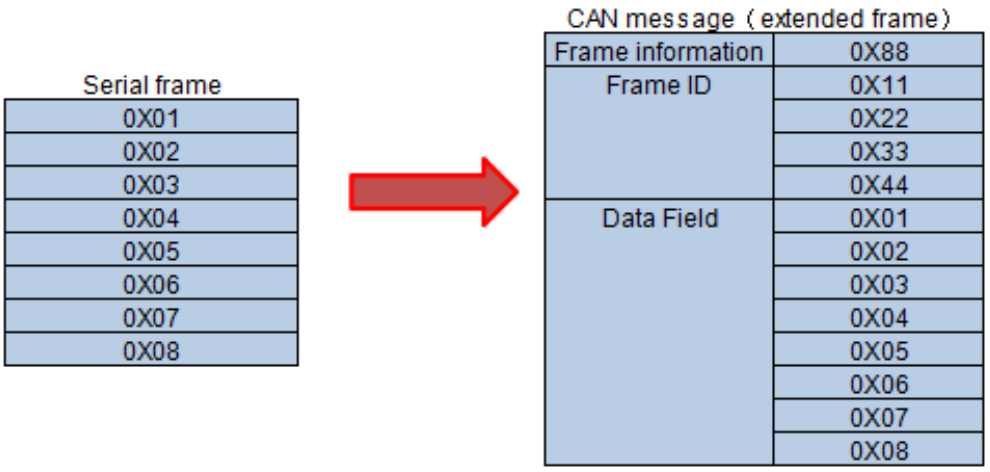
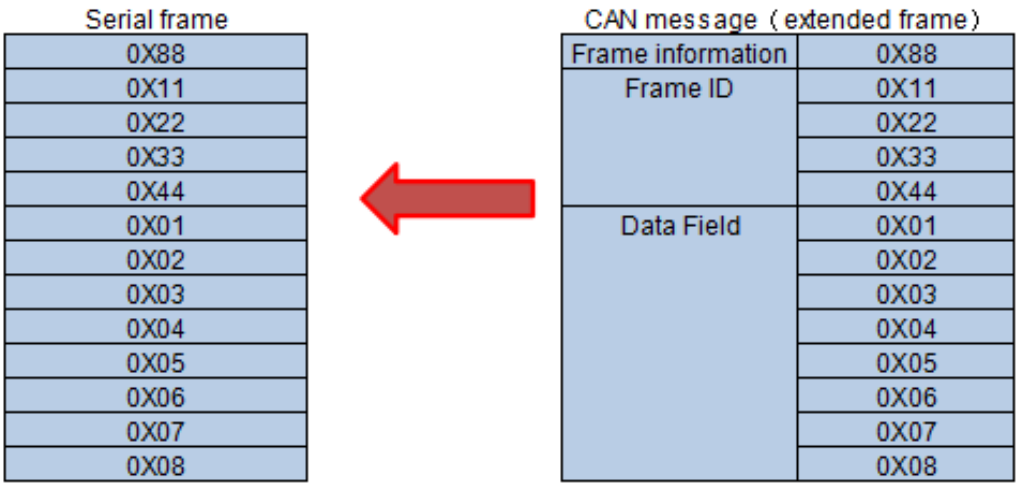


Figure 2.13 Serial Frame to CAN Message (Data Conversion)

CAN Message to Serial Frame:

The frame configured as CAN message is "extended frame", the frame identifier is "11, 22, 33, 44", and the forwarding data is "01, 02, 03, 04, 05, 06, 07, 08". CAN message and the converted serial frame are shown in figure.



2.5 Application Notes

- It’s recommended to be used in low speed systems, the converter is not suitable for high speed data transmission.
- After switching between "configuration mode" and "normal working mode", the power must be recharged, otherwise the original working mode will still be executed and the switch cannot be successfully implemented.
- In “transparent Conversion with Tag” and “Modbus Conversion”, note that the frame type of the CAN network must be the same as the configured frame type, otherwise it will not communicate successfully.
- In “transparent Conversion with Tag” and “Modbus Conversion”, the transmission of serial frames must conform to the configured time requirements, otherwise communication errors may occur.
- CAN bus is half duplex, so in the process of data conversion, try to ensure the order of the two sides of the bus data. If both buses send large amounts of data to the converter at the same time, the data conversion may be incomplete.
- When using CAN485, attention should be paid to the rationality of the baud rate of both sides of the bus and the time interval of data sent by both sides of the bus, and the data bearing capacity of the bus with lower baud rate should be considered during the conversion.
- For example, when CAN bus data is transferred to serial bus, CAN bus speed can reach thousands of frames per second, but serial bus can only reach hundreds of frames per second. Therefore, when the speed of CAN bus is too fast, the data conversion will be incomplete.
- In general, CAN baud rate should be about 3 times of that of serial port, so the data transmission will be relatively uniform (because other functional fields are added to the data transmission in CAN bus, which is equivalent to increasing the length of data, so the time of CAN transmission will be longer than that of serial

bus under the same baud rate).

3 Maintenance and Service

Since the date of product delivery, our company provides three-year product warranty. According to our company's product specification, during the warranty period, if the product exists any failure or functional operation fails, our company will be free to repair or replace the product. However, the commitments above do not cover damage caused by improper usage, accident, natural disaster, incorrect operation or improper installation.

In order to ensure that consumers benefit from our company's interface converter, consumers can get help and solutions in the following ways:

Internet Service, Call technical support office, product repair or replacement.

3.1 Internet Service

More useful information and tips are available via our company's website. Website: http://www.3onedata.com

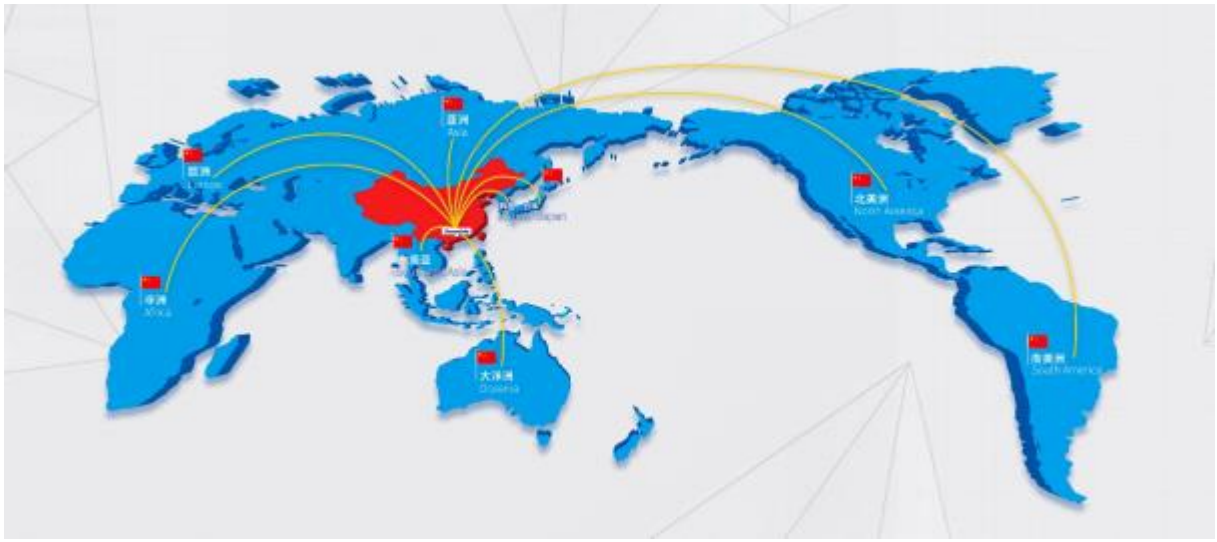
3.2 Call technical support office;

Users using our company products can call technical support office. Our company has professional technical engineers to answer the questions and help solve the products or usage problems ASAP. Free service hotline: +86-400-880-4496

3.3 Product repair or replacement

As for the product repair, replacement or return, customers should firstly confirm with the company technical staff, and then contact the company salesmen and solve the problem. According to the company's handling procedure; customers should negotiate with our company's technical staff and salesmen to complete the product maintenance, replacement or return.

3onedata
Make network communication more reliable



3onedata Co., Ltd.

Address: 3/B, Zone 1, Baiwangxin High Technology Industrial park, Nanshan District, Shenzhen, 518108 China

Tel: +86-755-26702668

E-mail: sales@3onedata.com

Fax: +86-755-26703485

Website: http://www.3onedata.com