



# User Manual

Version 1.6 Jun 2022

# The UART to CAN Bus Converter

(I-7530, I-7530A, I-7565, I-7530-FT, tM-7530, tM-7530A, tM-7565)



# Table of Contents

---

<b>1. Introduction</b>	<b>5</b>
1.1 Features and differences of CAN converter series	7
1.2 Communication Mode Description	12
1.2.1 Normal mode	12
1.2.2 Pair Connection mode	14
1.2.3 Listen only mode	15
<b>2. Hardware information of CAN converter</b>	<b>17</b>
2.1 Specification Table	17
2.2 Pin Assignment	19
2.3 Block Diagram	22
2.4 Wiring method	24
2.5 Terminal resistor	28
2.6 Init / Normal DIP switch	32
2.7 LED indicator	34
<b>3. Software for setting and testing (VxCAN Utility)</b>	<b>37</b>
3.1 Search and setting interface	38
3.2 UART setting	40
3.3 CAN setting	43
3.4 Communication mode setting	46
3.5 Saving and default value	50
3.6 Testing interface	51
<b>4. UART command string</b>	<b>53</b>
4.1 Checksum algorithm	55
4.2 tIIILDD...[CHK]<CR>	56
4.3 TIIIL[CHK]<CR>	57
4.4 eIIIIIIILDD...[CHK]<CR>	58
4.5 EIIIIIIIL[CHK]<CR>	59
4.6 S[CHK]<CR>	60
4.7 C[CHK]<CR>	62
4.8 P0BBDSPA[E][CHK]<CR>	63
4.9 P1B [CHK]<CR>	65
4.10 P2BBDSPA[E][CHK]<CR>	66
4.11 P3SBCCCCCCCCMMMMMMMM [CHK]<CR>	68

4.12 RA[CHK]<CR> .....	70
4.13 Error Code .....	71
<b>5. Appendix .....</b>	<b>73</b>
5.1 Cable selection .....	74
5.2 I-7565 Driver Installation Instructions .....	75
5.2.1 Verifying the installation .....	79
5.2.2 Remove the driver .....	80
5.3 tM-7565 Driver Installation Instructions .....	81
5.3.1 Installing the hardware .....	84
5.3.2 Verifying the installation .....	87
5.3.3 Remove the driver .....	88
5.4 Calculation of user-defined CAN baud rate .....	89
5.4.1 Setting parameters of CAN Converter Utility .....	89
5.4.2 Parameter description .....	90
5.4.3 Example description .....	94
5.5 CAN ID filter setting .....	97
5.6 Application of Pair Connection mode .....	98
5.7 COM port software .....	101

## Important Information

### Warranty

---

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

### Warning

---

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, not for any infringements of patents or other rights of third parties resulting from its use.

### Copyright

---

Copyright © 2018 by ICP DAS Co., Ltd. All rights are reserved.

### Trademark

---

Names are used for identification purpose only and may be registered trademarks of their respective companies.

### Contact us

---

If you encounter any problems while operating this device, feel free to contact us via mail at: [service@icpdas.com](mailto:service@icpdas.com) .

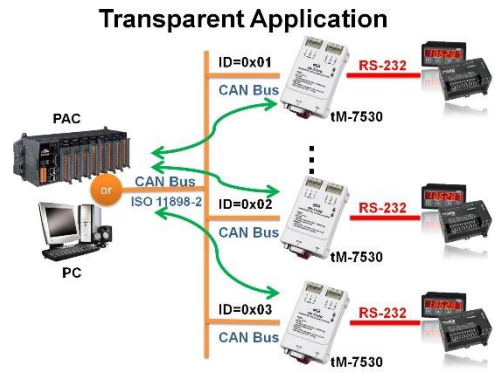
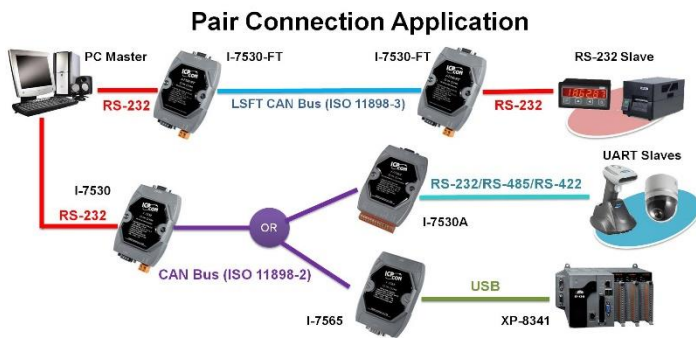
# 1. Introduction

CAN (Controller Area Network) bus is a serial communication interface, which is highly fault-tolerant even in harsh environments with noise. Therefore, there is often a need in the industry to convert interfaces such as RS-232, RS-485, RS-422 or USB to CAN interface for communication, so CAN converters are developed. For example, when the original equipment with RS-232, RS-485, RS-422 or USB interface (referred to as UART side, such as PC, I-8000, WinPAC-8000...etc) needs to control or monitor other new CAN interface equipment (referred to as CAN side), the electrical signal and data can be converted through the CAN converter, the data of the UART side device can be transferred to the CAN side device, or the data of the CAN side device can be transferred to the UART side device.

In addition, the CAN converter supports three communication modes with normal mode, pair connection mode and listen only mode. Among them, the pair connection mode has a function similar to transparent transmission, which can transparently transmit the data from one side to the other side.

The application architecture of the communication mode is shown in the below figure. For details, please refer to Section 1.2.





# 1.1 Features and differences of CAN converter series

## Common features :

- ◆ Support CAN 2.0A and CAN 2.0B specifications.
- ◆ According to the CAN bus specification, the maximum transmission distance is 1000 meters.
- ◆ Built-in Dual-Watchdog function in the module.
- ◆ Use software to set module parameters.
- ◆ Provide power, data flow, error LED indicator.
- ◆ Provide CAN and UART receiver software buffer.

◆ Comparison table of differences between converters:

	I-7530-FT	I-7530	I-7530A	I-7565	tM-7530	tM-7530A	tM-7565
<b>CAN interface</b>							
Controller	Microprocessor inside with 20 MHz Microprocessor inside with 48 MHz (*1)			Microprocessor inside with 48 MHz			
Specification	ISO-11898-3	ISO-11898-2					
Terminal Resistor	Built-In 1kΩ	Built-In 120Ω (Jumper-selected)			N/A	Built-In 120Ω (Wiring-selected)	Built-In 120Ω (Jumper-selected)
Max FPS (@1M bps)	279 ~ 360 423 ~ 519 (*1)			423 ~ 519			
Baud Rate (bps)	10k ~ 125k	10k ~ 1M					
FIFO	1000 frames 256 frames (*1)			256 frames			

(\*1) The module shell laser is marked V2, or the firmware version is after v3.90.



	I-7530-FT	I-7530	I-7530A	I-7565	tM-7530	tM-7530A	tM-7565
<b>UART interface</b>							
COM	RS-232	RS-232/485/422 (can't be used simultaneously)		USB (virtual COM)	RS-232	RS-232/485/422 (can't be used simultaneously)	USB (virtual COM)
Baud Rate (bps)	110~115200 110~230400 (*1)			110~921600	110~230400		110~230400 110~921600 (*2)
FIFO	900 frames 256 bytes (*1)				256 bytes		

(\*1) The module shell laser is marked V2, or the firmware version is after v3.90.

(\*2) The firmware version is after v3.90.

	I-7530-FT	I-7530	I-7530A	I-7565	tM-7530	tM-7530A	tM-7565
<b>Extended function</b>							
Listen only mode	Don't support				Support		
UART command string append timestamp (*1)	Don't support				Support		
User-defined CAN baud rate (*2)	Don't support	Support					
In pair connection mode, the UART command string append the response with CAN ID (*3)	Don't support				Support		
In pair connection mode, the user defines the end character of the UART command (*3)	Don't support				Support		

Command timeout in pair connection mode (*3)	Don't support	Support
--	---------------	---------

(\*1) Please refer to section 3.2 UART settings for the description in "Timestamp Response".

(\*2) Please refer to section 3.3 CAN settings for the description in "User Defined CAN Baud Rate".

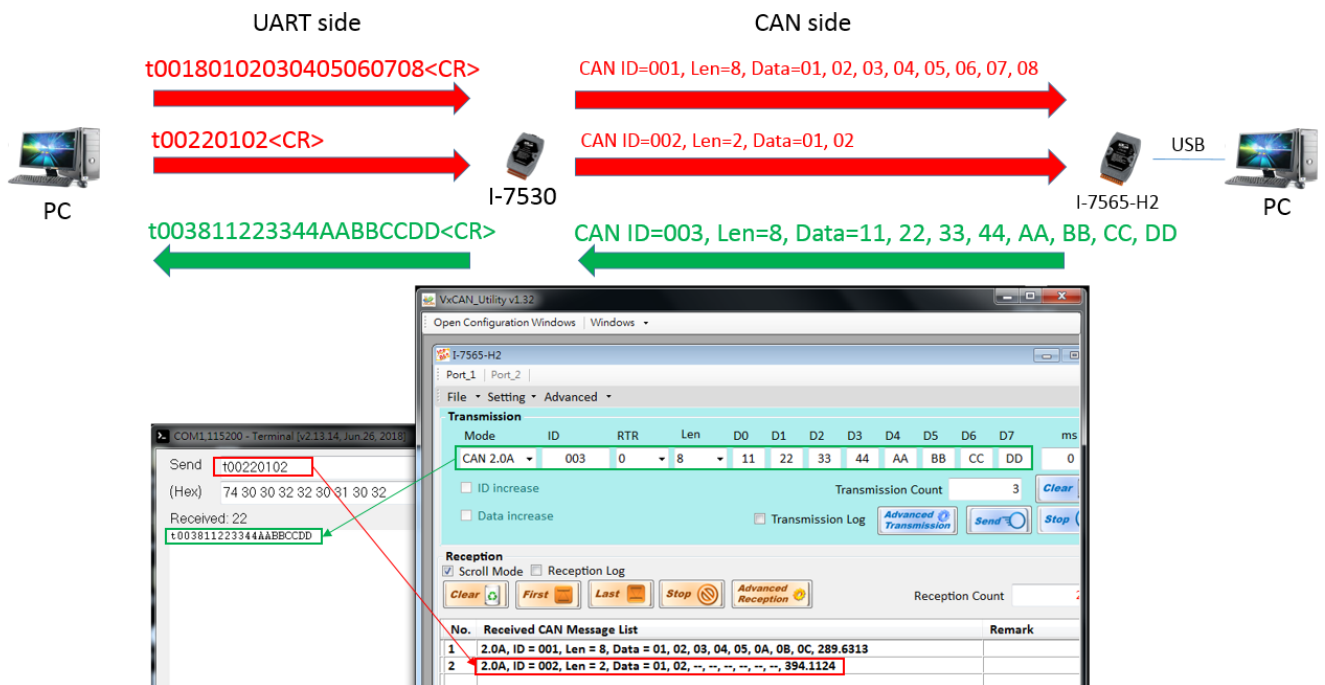
(\*3) Please refer to Section 3.4 for the description of "Response with CAN ID", "End Characters of UART Commands", "Pair Connection Command Timeout" in Paired Connection Mode.

# 1.2 Communication Mode Description

The CAN converter supports three communication modes with Normal mode, Pair connection mode and Listen only mode as below description. In addition, please refer to section 3.4 for the setting method of the communication mode. If you need to use the COM port software during the test, please refer to section 5.7.

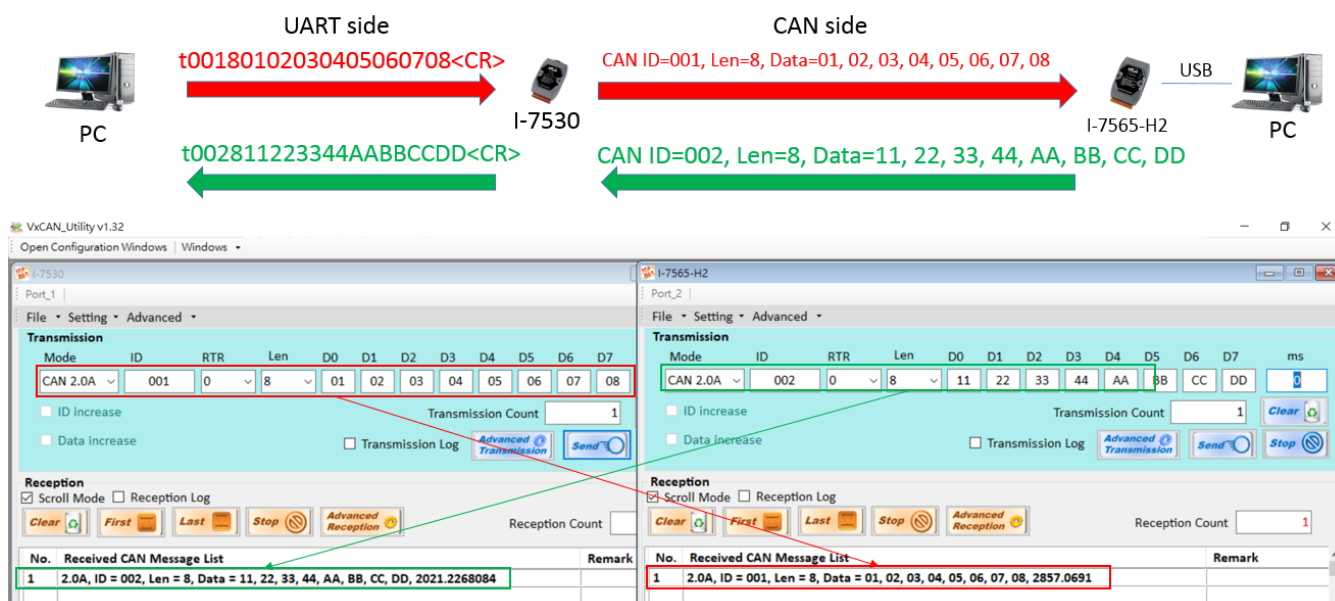
## 1.2.1 Normal mode

After inputting a fixed-format "UART command string" from the UART side of the CAN converter, the CAN converter will respond whether the command is valid. If the command is valid, it will output the data to the CAN side, but if the command is invalid, it will not act. Conversely, when data be inputed from the CAN side of the CAN converter, the CAN converter will also output a fixed-format "UART command string" to the UART side as below figure. For the detailed command format, please refer to the UART command string table in chapter 4.



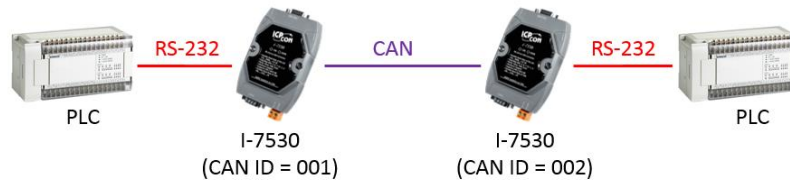
In addition, if the customer only needs to exchange CAN messages with the CAN device through the CAN converter (for example: I-7530), and does not need to know the UART command string on the UART side, the VxCAN Utility can be used to achieve this requirement.

The test interface of VxCAN Utility can convert UART command strings into CAN messages as below figure, and the used method of test interface in VxCAN Utility, please refer to section 3.6.

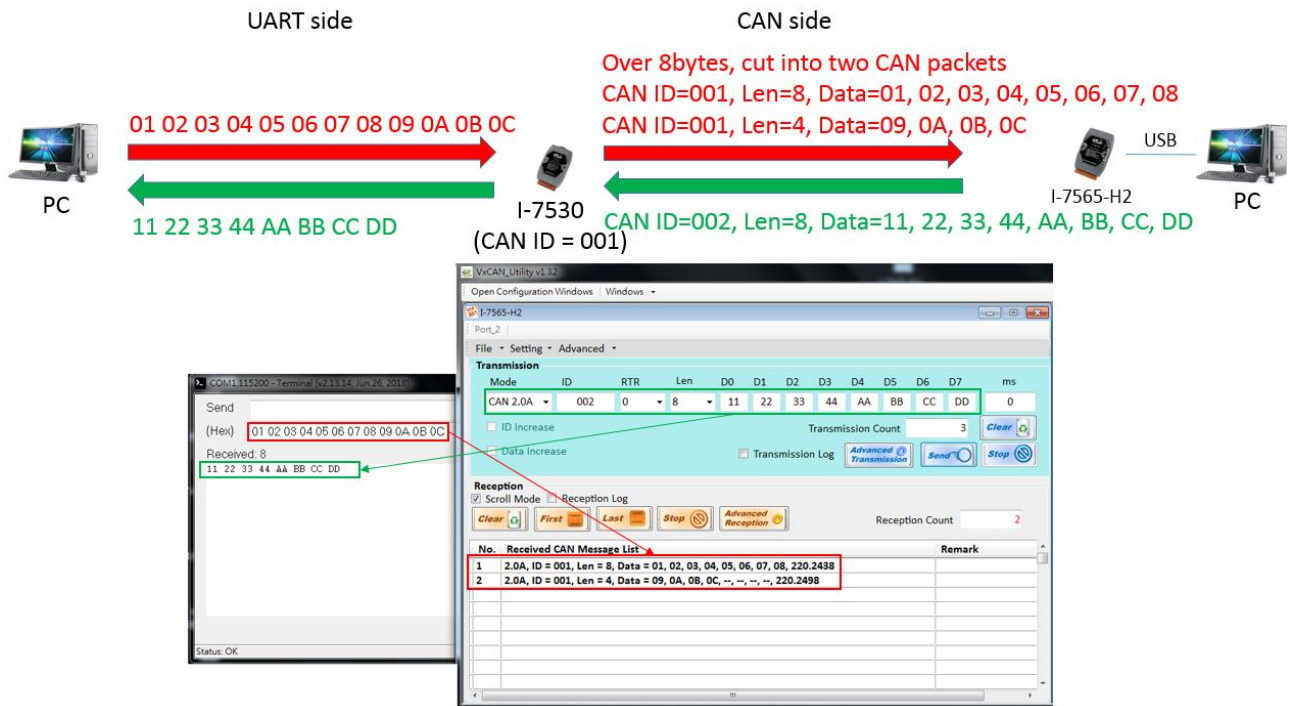


## 1.2.2 Pair Connection mode

As the name suggests, this mode is designed to use two CAN converters for pair connection. The main purpose will transfer the interface from UART to CAN, and then transfer again the interface from CAN to UART, so that the interface conversion effect can be achieved without modifying the program of the field device as below figure .



In addition, if only one CAN converter (for example: I-7530) uses this mode, the effect will be similar to the function of transparent transmission. The data be inputed from the UART side of the CAN converter will be directly output to the CAN message as below figure.



### 1.2.3 Listen only mode

This mode only supports tM-7530, tM-7530A and tM-7565, and be used to monitor the messages of the CAN network without converting the message, so it will not send any CAN messages to the CAN network, such as Error Frame or ACK signal.

For example, there are two CAN devices (such as I-7530 and I-7565-H2) in the communicating, and the user want to obtain CAN messages between CAN devices, the user can set the tM-7565 to listen only mode, and then connect the CAN side into the CAN network, and connect the UART side to the PC for monitoring.

If the user uses a common communication port (COM port) testing software to monitor in the PC as shown in figure 1.2.3.1, and will get the UART command string.

If the user uses the testing interface of VxCAN Utility to monitor in the PC as shown in figure 1.2.3.2, and will get CAN messages.

Among them, if the user need to use the COM port testing software, please refer to section 5.7. If the user need to use the testing interface of VxCAN Utility, please refer to section 3.6.

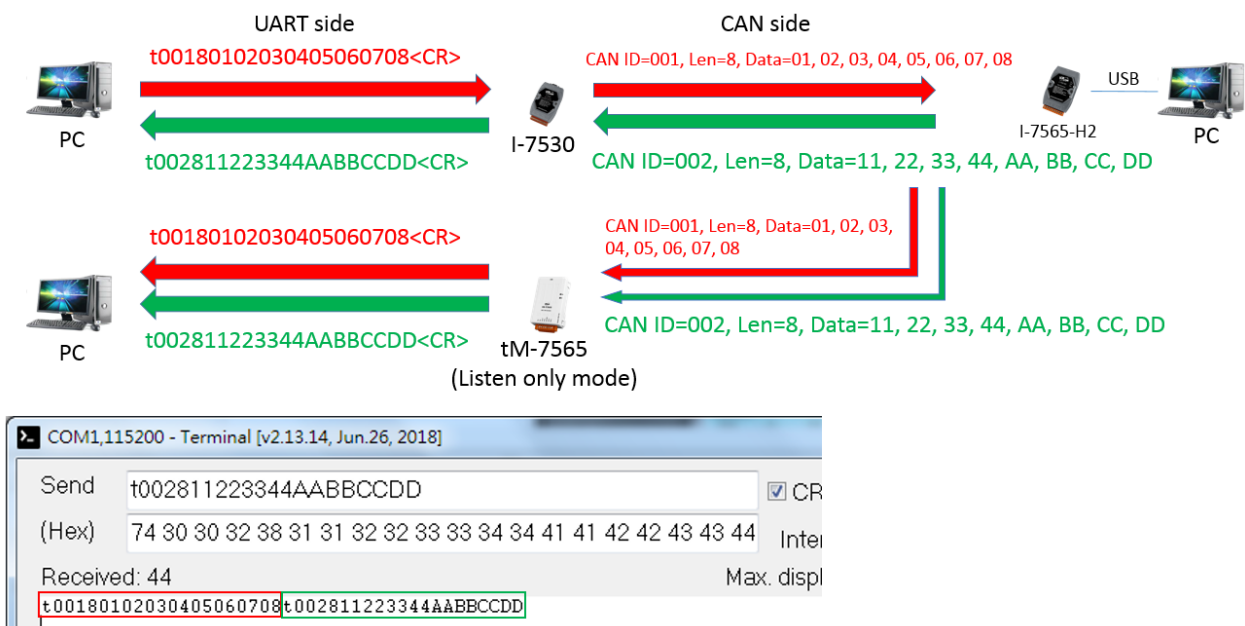


Figure 1.2.3.1. Using COM port testing software to monitor the UART command string

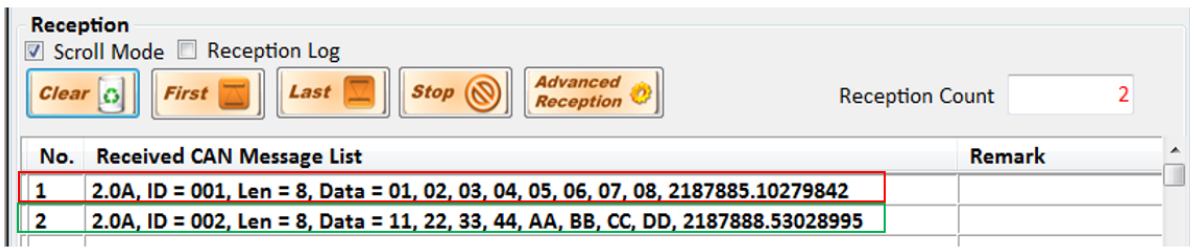


Figure 1.2.3.2. Using VxCAN utility to monitor the CAN message



## 2. Hardware information of CAN converter

### 2.1 Specification Table

	I-7530	I-7530A	I-7565	tM-7530	tM-7530A	tM-7565	I-7530-FT
<b>CAN interface</b>							
Channels	1						
Connector	9-pin male D-Sub		3-pin spring type	7-pin screwed terminal block		9-pin male D-Sub	
Baud Rate (bps)	10 k ~ 1 M						10k ~ 125k
Isolation	DC-DC: 3000 VDC photocoupler: 2500 Vrms						N/A
Terminator Resistor	Built-In 120Ω (Jumper-selected)		N/A	Built-In 120Ω (Wiring-selected)	Built-In 120Ω (Jumper-selected)	Built-In 1kΩ	
Specification	ISO 11898-2						ISO 11898-3
<b>UART/USB interface</b>							
COM	RS-232	RS-232 RS-485 RS-422	virtual COM	RS-232	RS-232 RS-485 RS-422	virtual COM	RS-232
Connector	9-pin female D-Sub	14-pin screwed terminal block	USB Type B	9-pin female D-Sub	10-pin screwed terminal block	USB Type B	9-pin female D-Sub
Baud Rate (bps)	110 ~ 115200		110 ~ 921600	110 ~ 230400		110~230400 (*3)	110 ~ 115200
Data bit	5, 6, 7, 8						
Stop bit	1, 2						
Parity bit	None, Odd, Even						
<b>LED</b>							
indicator light	ON LED ERR LED			Run LED Err LED		ON LED ERR LED	
<b>Power</b>							
Protection	over reverse polarity protection, Over-			over reverse polarity protection, Over-voltage brown-out			over reverse polarity

	I-7530	I-7530A	I-7565	tM-7530	tM-7530A	tM-7565	I-7530-FT
	voltage brown-out protection			protection			protection, Over-voltage brown-out protection
Power Consumption	1W						
<b>Mechanism</b>							
Installation	DIN-Rail						
Dimensions (W x L x H)	72mm x 118mm x 33mm		52mm x 86mm x 32mm	52mm x 93mm x 27mm	52mm x 87mm x 27mm	72mm x 118mm x 33mm	
<b>Environment</b>							
Operating Temp.	-25 ~ 75 °C						
Storage Temp.	-30 ~ 80 °C						
Humidity	10 ~ 90% RH, Non-condensing						

(\*1) The RS-232 interface of the CAN converter only supports half-duplex mode, not full-duplex mode.

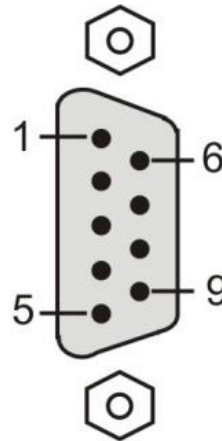
(\*2) When using the I-7565 or tM-7565 for the first time, please refer to section 5.2 or section 5.3 to install the USB driver first.

(\*3) The firmware version is after v3.90 can support 110~921600 bps ◦

## 2.2 Pin Assignment

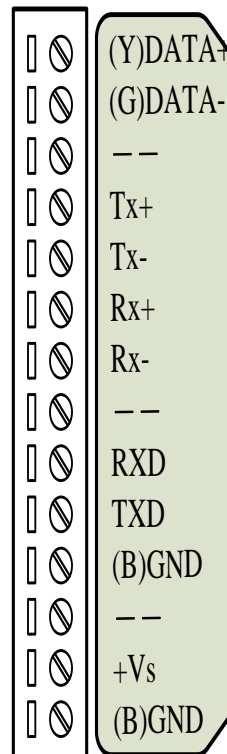
RS-232 terminal of I-7530/tM-7530/I-7530-FT:

Pin	Description
1	N/A
2	TXD
3	RXD
4	N/A
5	GND
6	N/A
7	N/A
8	N/A
9	N/A



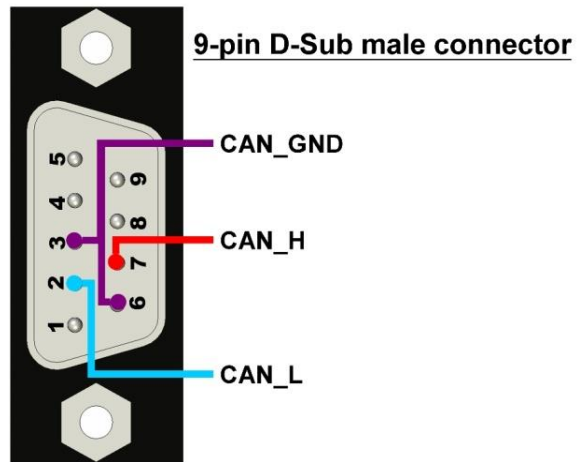
RS-232/485/422 terminal of I-7530A:

Pin	Description
1	(Y)DATA+ (RS-485)
2	(G)DATA- (RS-485)
3	N/A
4	Tx+ (RS-422)
5	Tx- (RS-422)
6	Rx+ (RS-422)
7	Rx- (RS-422)
8	N/A
9	RXD (RS-232)
10	TXD (RS-232)
11	(B)GND (RS-232)
12	N/A
13	+Vs (10 ~ 30 VDC Power)
14	(B)GND (Power ground)



**CAN bus terminal (DB9 male) of I-7530 / I-7530A / I-7565 / I-7530-FT:**

Pin	Description
1	N/A
2	CAN Low
3	CAN GND
4	N/A
5	CAN GND
6	N/A
7	CAN High
8	N/A
9	N/A

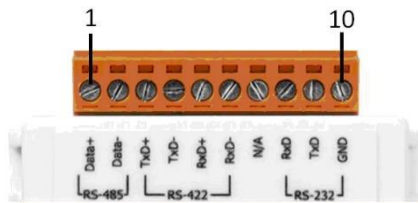


**Power/CAN terminal of tM-7530:**

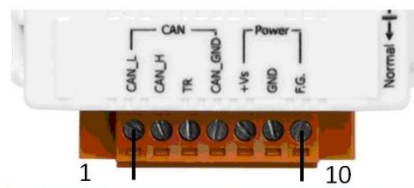
Pin	Description
1	F.G.
2	GND (Power ground)
3	+Vs (10 ~ 30 VDC Power)
Pin	Description
1	CAN_GND
2	CAN_H
3	CAN_L



### UART/CAN terminal of tM-7530A:



Pin	UART Description
1	RS-485 D+
2	RS-485 D-
3	RS-422 TxD+
4	RS-422 TxD-
5	RS-422 RxD+
6	RS-422 RxD-
7	N/A
8	RS-232 RxD
9	RS-232 TxD
10	RS-232 GND



Pin	Power, CAN Bus Description
1	CAN Low (CAN_L)
2	CAN High (CAN_H)
3	Terminator Resistor Enabled Switch(TR)
4	CAN Ground(CAN_GND)
5	+10 to +30 V <sub>DC</sub> (+Vs)
6	Power Ground (GND)
7	Frame Ground (F.G.)

### CAN terminal of tM-7565:



Pin	Description
1	CAN Low(CAN_L)
2	CAN High(CAN_H)
3	CAN Ground(CAN_GND)
4	CAN Low(CAN_L, bypass)
5	CAN High(CAN_H, bypass)
6	N/A
7	Frame Ground(F.G.)

## 2.3 Block Diagram

I-7530 and tM-7530 provide the communication interface of CAN and RS-232, and the CAN side has isolator.

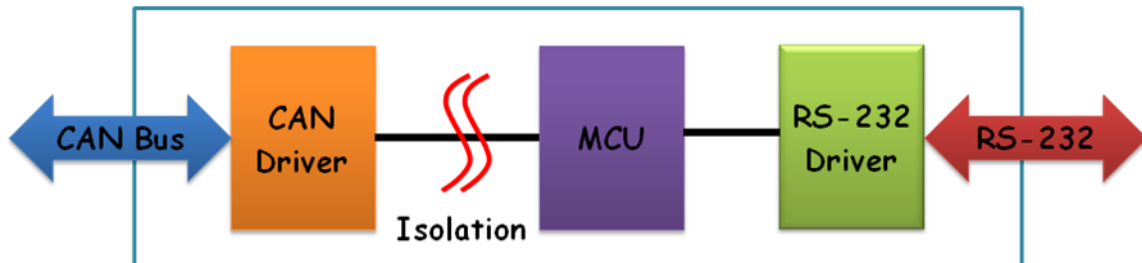


Figure: Hardware block diagram of I-7530 and tM-7530

The hardware appearance of the I-7530-FT is same as the I-7530, and the main difference between the I-7530-FT and the I-7530 is compliance with different CAN specifications. Among them, the I-7530-FT is a CAN device that complies with the ISO 11898-3 specification and has the characteristics of low-speed fault tolerance (LSFT), but there is no isolator on the CAN side, while the I-7530 is a CAN device that complies with the ISO 11898-2 specification, and there is an isolator on the CAN side.

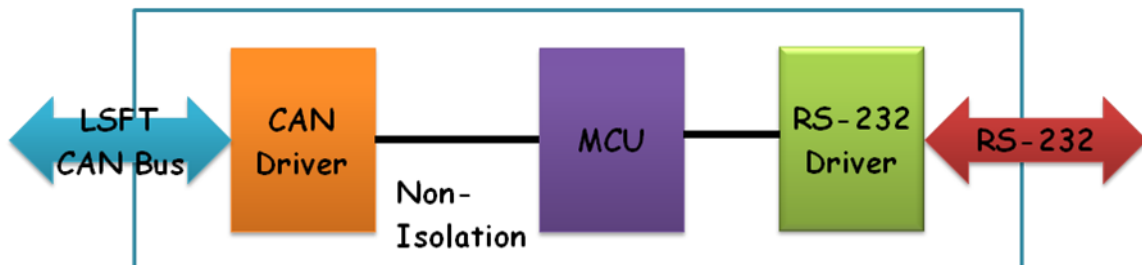


Figure: Hardware block diagram of I-7530-FT

I-7530A and tM-7530A provide four communication interfaces with CAN, RS-232, RS-485 and RS-422. Among them, RS-232, RS-485 and RS-422 are the same UART communication port, so they cannot be used at the same time. In addition, the I-7530A and tM-7530A have isolators on the CAN side.

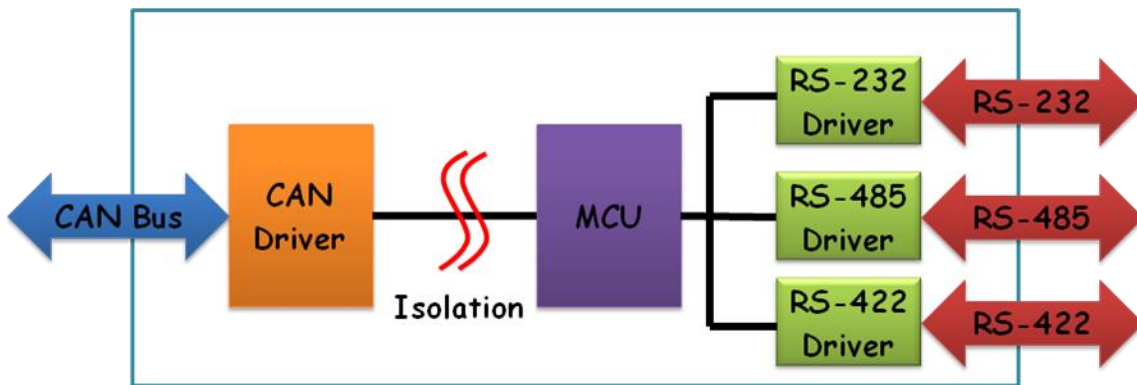


Figure: Hardware block diagram of I-7530A and tM-7530A

I-7565, tM-7565 provide CAN and USB communication interface, and provide virtual COM function on USB interface, that is to say, if the device only has USB interface, you can use I-7565, tM-7565 instead of I-7530. In addition, the I-7565 and tM-7565 are powered through the USB interface, so no external power supply is required, and I-7565 and tM-7565 have isolators on the CAN side.

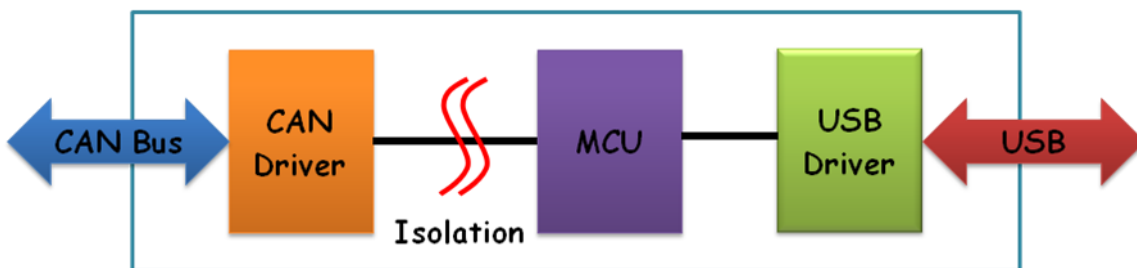


Figure: Hardware block diagram of I-7565 and tM-7565

## 2.4 Wiring method

The I-7530, I-7530-FT and tM-7530 are RS-232 data communication modules, and the RS-232 interface uses a D-Sub 9-pin female terminal. The below picture will describe the wiring between the PC and the module.

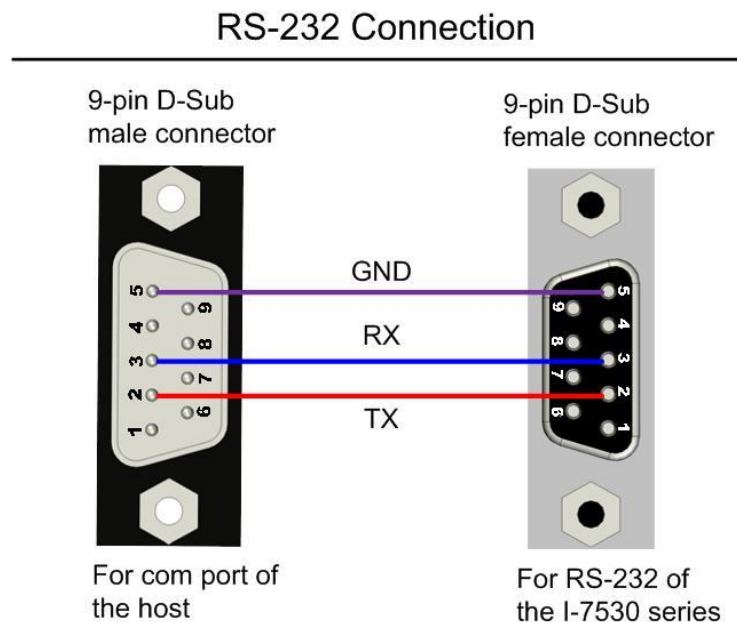


Figure: Connecting the I-7530, I-7530-FT or tM-7530 to a PC

The UART communication port of the I-7530A has three interfaces with RS-232, RS-485 and RS-422, and the three interfaces cannot be used at the same time. The below figure shows the wiring method for connecting the I-7530A to other UART devices. When using the RS-485 or RS-422 interface, we recommend using twisted pair cables as the communication medium.

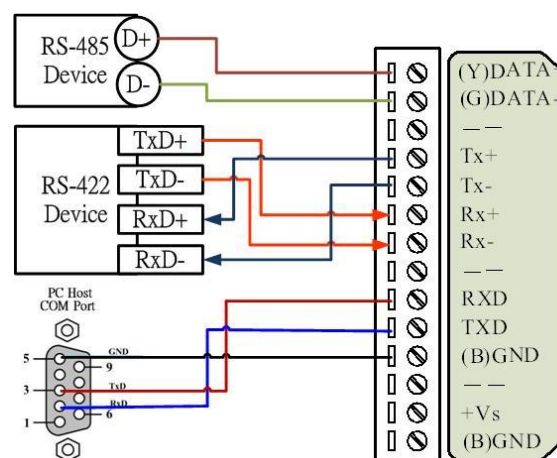


Figure: The wiring method of I-7530A's UART side



I-7530, I-7530A, I-7565 and I-7530-FT in CAN communication port (DB9 male) pin assignment is based on CANopen DS102 introduction and the DeviceNet specification. The wiring method is as follows:

### CAN Devices Wire Connection

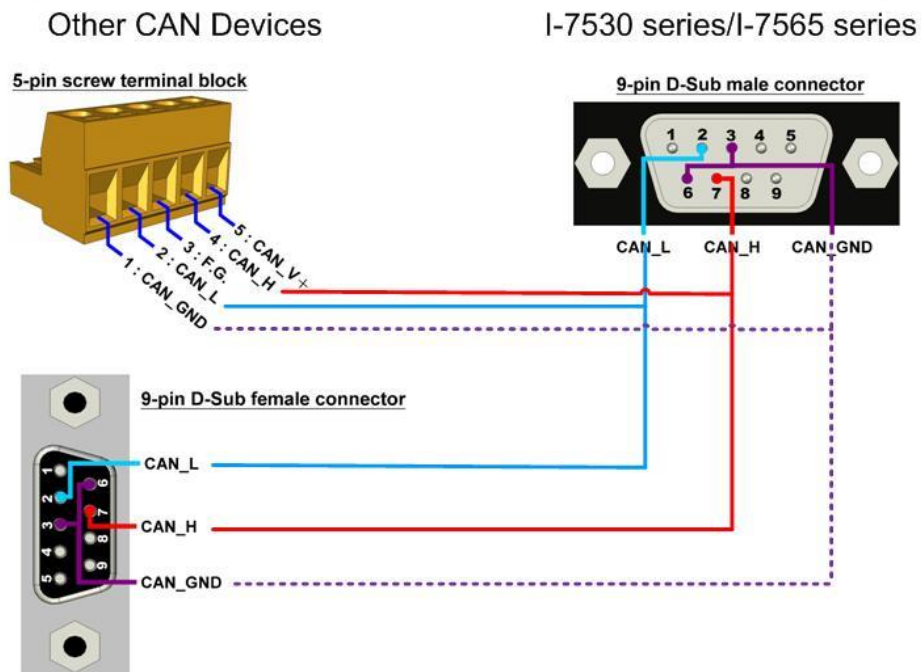


Figure: I-7530 series CAN bus wiring method

The CAN interface of the tM-7530 uses spring-loaded terminals, and the wiring methods as follows:



Figure: tM-7530 CAN bus wiring method

tM-7530A has three interfaces, namely RS-232, RS-485 and RS-422, and the three interfaces cannot be used at the same time. The wiring method uses screw terminals as follows:

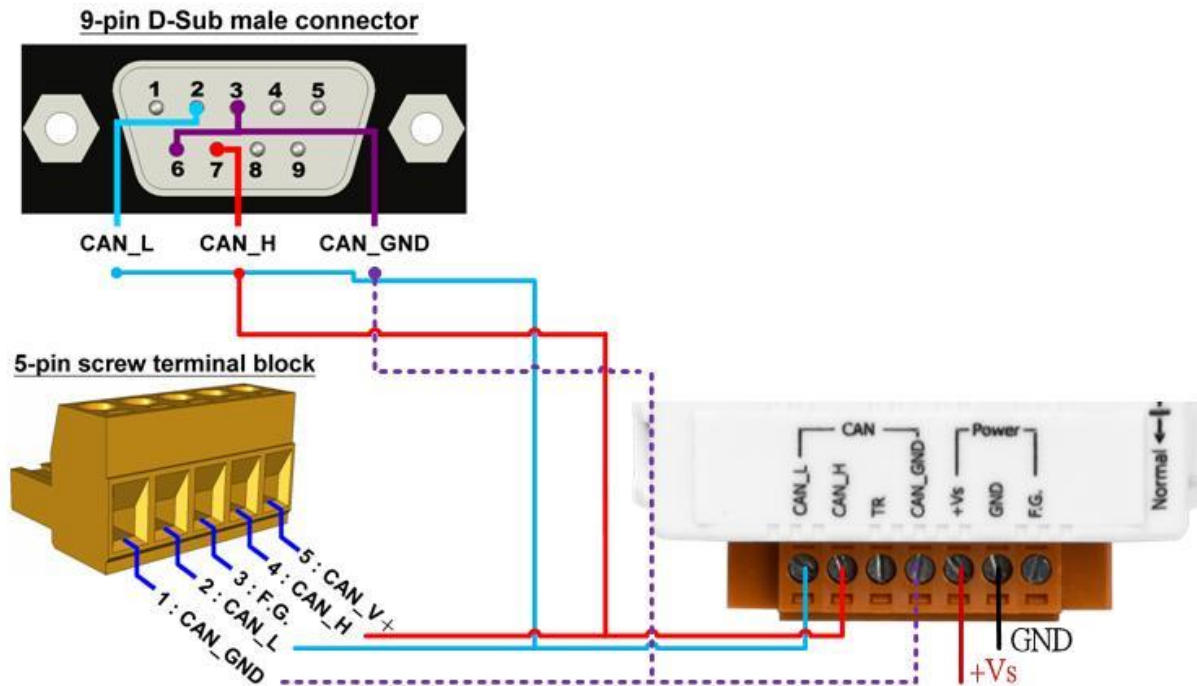


Figure: tM-7530A power supply terminal and CAN bus terminal wiring

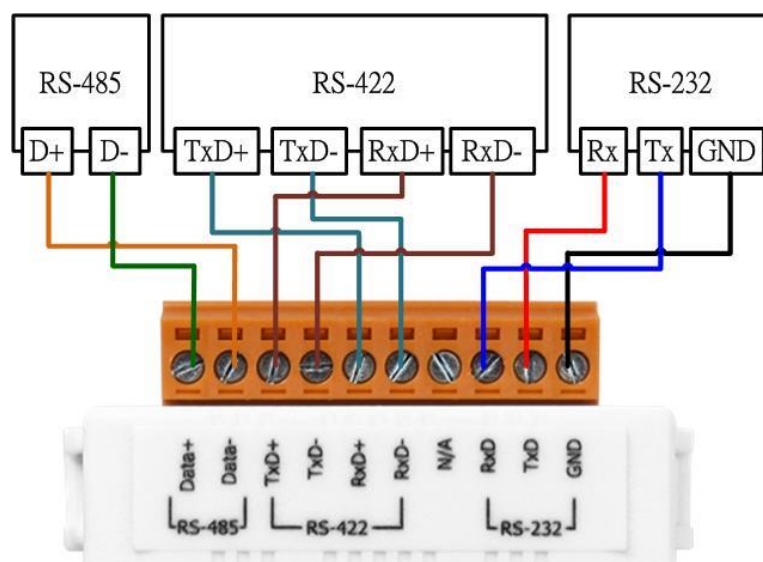


Figure: tM-7530A UART terminal wiring

The tM-7565 uses screw terminals, which are wired as follows:

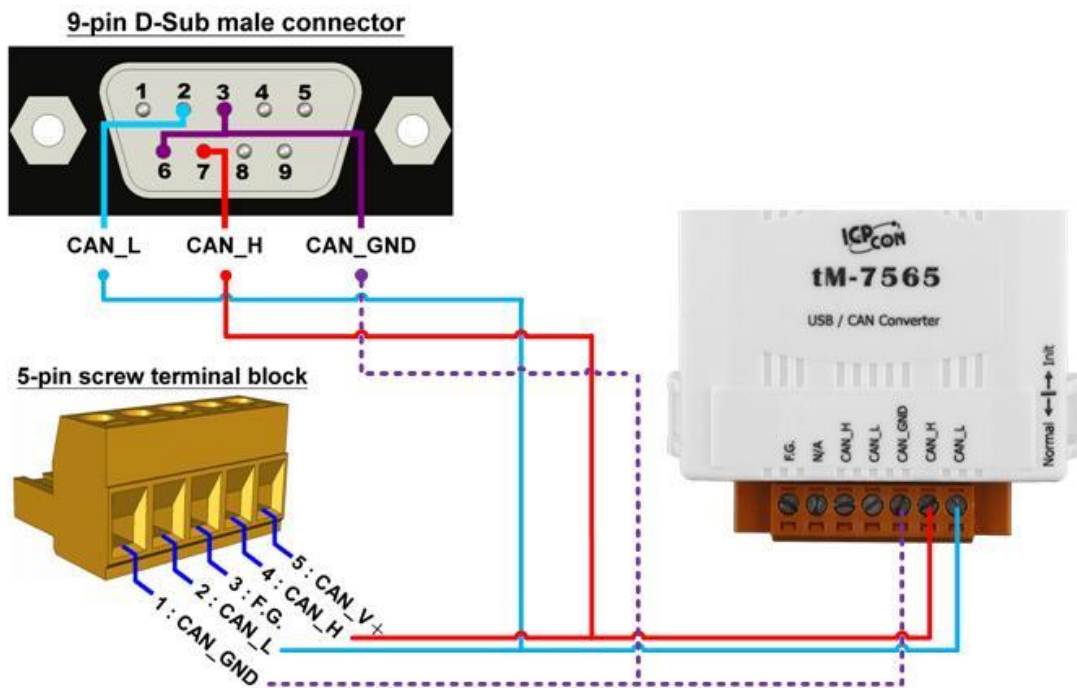


Figure: tM-7565 CAN bus terminal wiring

## 2.5 Terminal resistor

According to the ISO 11898 specification, in order to operate correctly the ISO 11898-2 CAN bus network, it is necessary to set terminal resistors at both ends, which is 120Ω as below figure. If there is no terminal resistor or only one terminal resistor exists on the CAN bus, it may affect the signal stability of the CAN network. Therefore, when the signal stability is not good, it is recommended to use a terminal resistor.

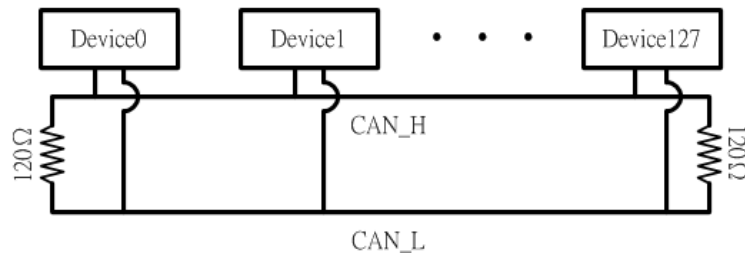
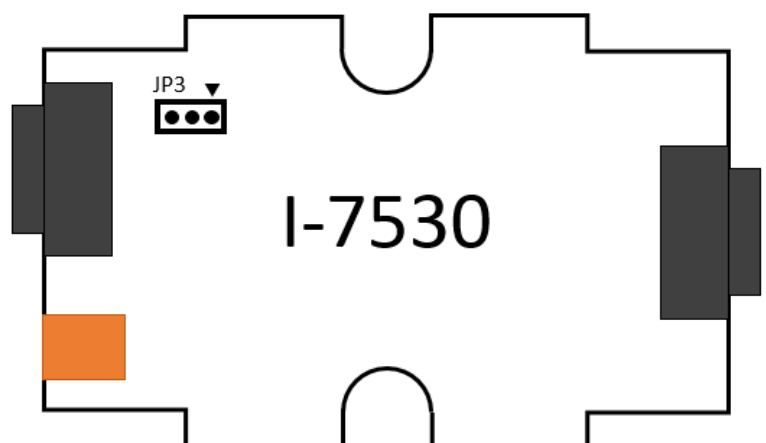


Figure: Terminal resistor for ISO 11898-2 CAN network

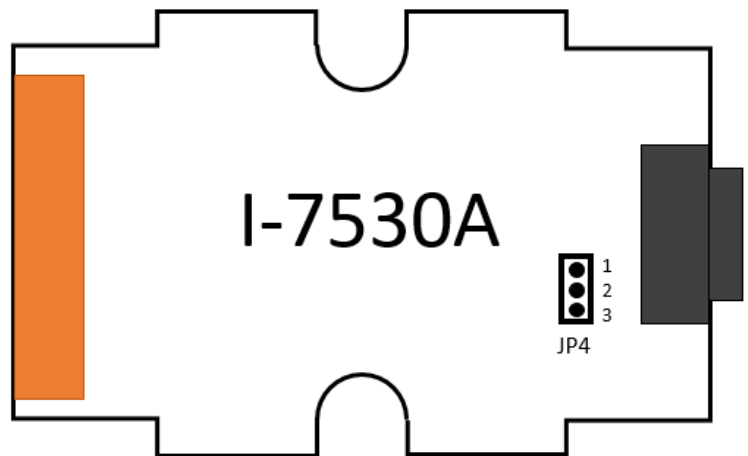
In addition, according to another ISO 11898-3 specification, the terminal resistor must be enabled directly, so the I-7530-FT compliant with this specification already built-in and enabled terminal resistors, while other CAN converters are based on the ISO 11898-2 specification, so they can be choosed whether to enable the terminal resistor by means of jumper, wiring or external terminal resistor by the user.

I-7530, I-7530A, I-7565 already built-in jumpers to select whether to enable or disable the terminal resistor. If the user want to enable or disable the terminal resistor of the CAN converter, please open the cover of the CAN converter and use the jumper to switch as below figure.

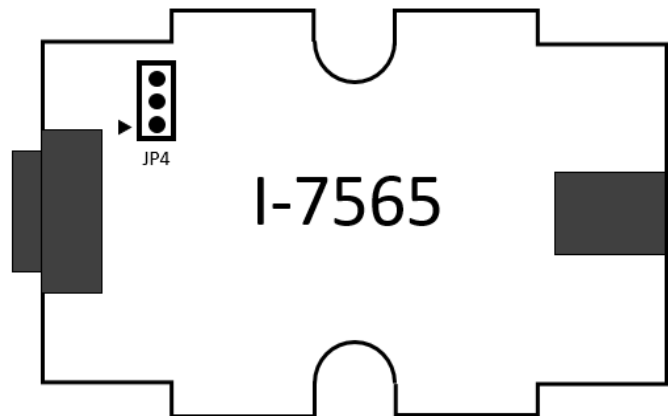
Status	Jumper position (JP3)
Enable (Default)	<p>The diagram shows a 3-pin header labeled 'JP3'. The middle pin is connected to the other two pins, forming a closed loop. This loop is highlighted with a red rectangle.</p>
Disable	<p>The diagram shows a 3-pin header labeled 'JP3'. The middle pin is not connected to the other two pins. This configuration is highlighted with a red rectangle.</p>



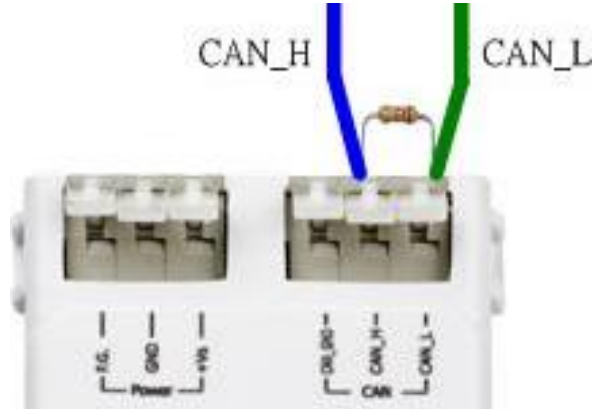
Status	Jumper position (JP4)
Enable (Default)	<p>1 2 3 JP4</p>
Disable	<p>1 2 3 JP4</p>



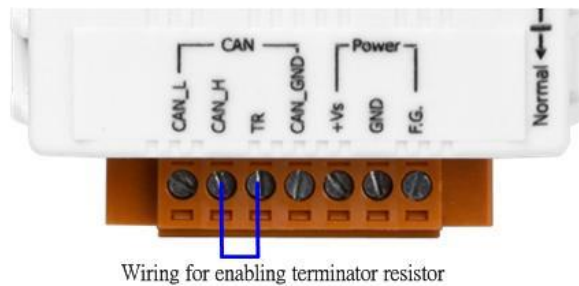
Status	Jumper position (JP4)
Enable (Default)	<p>JP4</p>
Disable	<p>JP4</p>



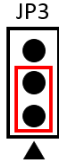

tM-7530 does not have a built-in terminal resistor. If the user needs to use a terminal resistor, please connect a 120 ohm terminal resistor in parallel between CAN\_H and CAN\_L of the CAN bus as below figure:

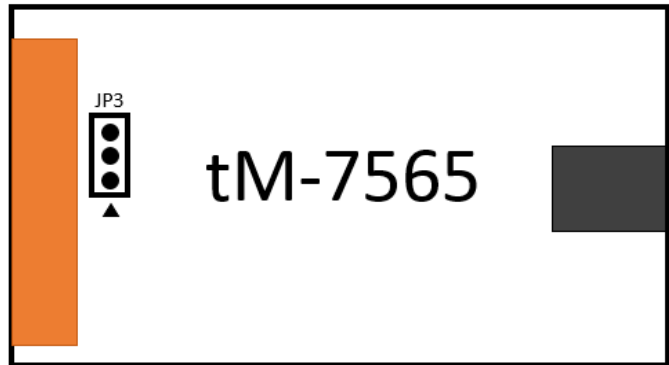


tM-7530A has a built-in terminal resistor. If the user needs to enable the terminal resistor function, please short the CAN\_H pin and the TR pin as below figure below:



The tM-7565 has built-in terminal resistors. If the user needs to enable the terminal resistor, please adjust the jumper to the 1 and 2 positions as below figure

Status	Jumper position (JP3)
Enable (Default)	 <p>The diagram shows a vertical rectangular component labeled JP3 with three circular terminals. A red rectangle highlights the top two terminals, and a black triangle points to the bottom terminal.</p>
Disable	 <p>The diagram shows a vertical rectangular component labeled JP3 with three circular terminals. A red rectangle highlights the bottom two terminals, and a black triangle points to the top terminal.</p>



## 2.6 Init / Normal DIP switch

The Init / Normal DIP switch is used to determine the working mode of the module after it is turned on. The working mode can be divided into "Initialization Mode" or "Normal Operation Mode", which are used to set the module parameters or use it normally.

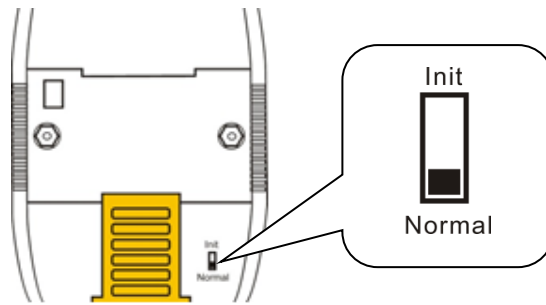
When the user uses the module for the first time, it is necessary to switch the DIP switch to Init and then power on, the module can enter the initialization mode to set the parameters, and after the parameter setting is completed, turn the DIP switch to Normal and repower on again, the module will enter the normal operation mode according to the set parameters.

The differences in the working modes are explained as follows:

- Initialization mode: This mode is used to set the communication parameters and communication modes of UART or CAN, such as: baud rate, data format, CAN ID filter, normal mode, etc. After completing the setting, all parameters will be stored to the EEPROM of module.
- Normal operation mode: This mode is used to convert UART data and CAN messages. In addition, in normal operation mode, if the "communication mode" of the module is set to "normal mode", you can obtain module information or change some UART and CAN port settings through UART command strings such as: P2 and P3, but the changed settings will not be stored to the EEPROM of the module. Please refer to the command list in chapter 4 for the UART command string.

The position of DIP switch will different by module. The DIP switch of the I-7530, I-7530A, I-7565 and I-7530-FT are located on the back of the module, while the tM-7530, tM-7530A and tM-7565 are located on the side of the module. About the position of the modules as below figure:





tM-7530	tM-7530A	tM-7565
<p>A line drawing of the tM-7530 device. A red box highlights a switch on the right side of the front panel, labeled 'Normal ↔ Init'.</p>	<p>A photograph of the tM-7530A device. A red box highlights a switch on the front panel, labeled 'Normal ← → Init'.</p>	<p>A photograph of the tM-7565 device. A red box highlights a switch on the front panel, labeled 'Normal ← → Init'.</p>

## 2.7 LED indicator

The CAN converter provides two LED indicators to show the module in which working mode, and whether an error has occurred as below described.

The I-7530, I-7530A, I-7565 and I-7530-FT provide ON LED and ERR LED, while tM-7530, tM-7530A and tM-7565 provide Run LED and Err LED. Among them, ON LED and Run LED have the same function, and ERR LED and Err LED have the same function. The following descriptions focus on ON LED and ERR LED.

The ON LED shows the module is working in which working mode. When the ON LED indicator flashes once per second means the module is in "Initialization Mode" and the DIP switch is in "Init", and when the ON LED indicator is always on means that the module is in "Normal Operation Mode" and the DIP switch is in "Normal".

The ERR LED shows whether an error has occurred in the module. If there is no error, the ERR LED will turn off. On the contrary, if an error has occurred, the ERR LED will be turned on or flashing, at this time the user can use the UART command string "S[CHK]<CR>" in chapter 4 to get the detailed error information. Among them, if the cause of the error is "CAN or RS-232/RS-485/RS-422/USB overload warning", you can use the UART command string "C[CHK]<CR>" to clear the error, or reboot the module to restore the state.

For the position and status display of the LED indicators in the I-7530, I-7530A, I-7565 and I-7530-FT, please refer to the pictures and tables as below.



LED	Status of DIP switch	Description
ON	Initialization mode (Init)	Flashing once per second
	Normal operation mode (Normal)	Turn on when no data needs to be send to CAN bus
Flash once when sending a CAN message		
ERR	some error occurred	Always turned on

For the position and status display of the tM-7530, tM-7530A and tM-7565 LED indicators, please refer to the pictures and tables as below.



LED	Status of DIP switch	Description
Run	Initialization mode (Init)	Flashing once per second
	Normal operation mode (Normal)	Turn on when no data needs to be send to CAN bus
		Flash once when sending a CAN message
Err	FIFO overflow CAN Bus Off CAN Error Passive	Always turned on
	Some errors occurred	Flashing

### 3. Software for setting and testing (VxCAN Utility)

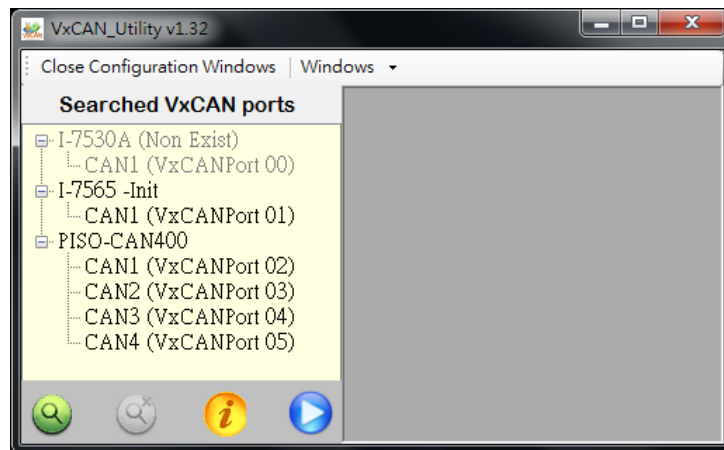
This chapter will describe how to use VxCAN Utility to set up CAN converters and test. The user can download VxCAN Utility from the download center in the product webpage of each CAN converter, or search and download VxCAN Utility on ICP DAS' homepage as below figure.




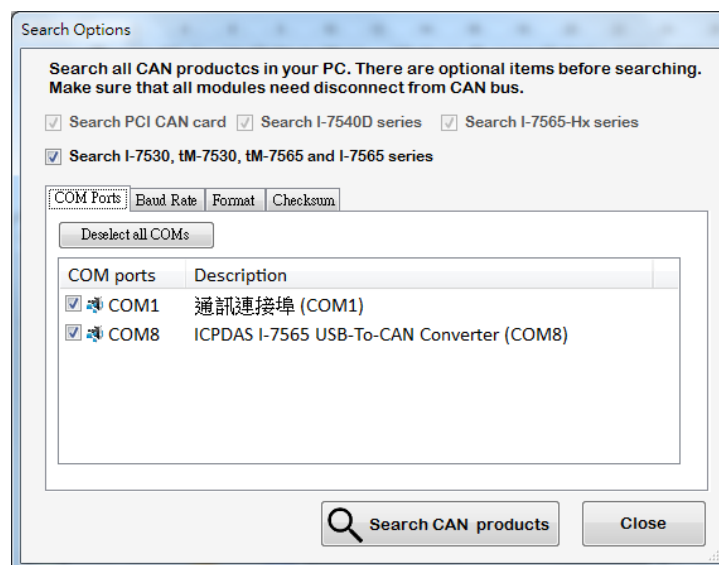
VxCAN Utility is a software developed based on the Virtual CAN technology developed by ICP DAS, it can integrate ICP DAS CAN converter products, and provides users to configure CAN converters, test CAN devices, monitor CAN network and analyze CAN messages through this software. For more information about VxCAN Utility, please refer to the above webpage description or VxCAN Utility manual.

## 3.1 Search and setting interface

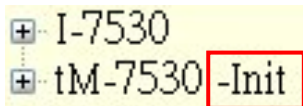
- Step 1. Before powering on the module, please switch the Init/Normal DIP switch of the module to Init, and then power on the module, and confirm that the ON LED or Run LED indicator is flashing every second once.
- Step 2. Connect the module to the COM port of the PC, and open the VxCAN Utility.



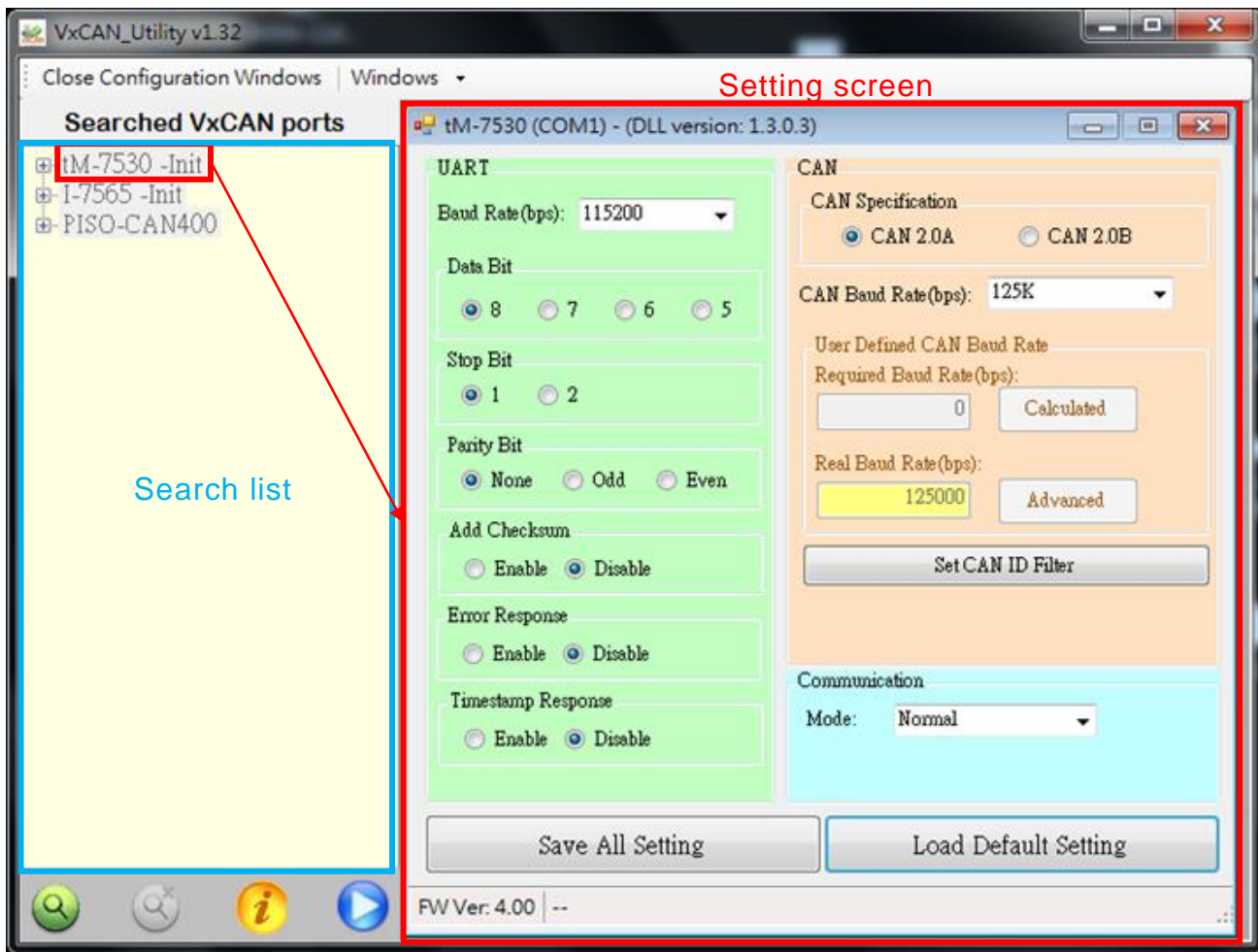
- Step 3. After clicking the Search button , select the module type to be searched in the Search Options screen, and click the Search CAN products button to search.



- Step 4. After the search is completed, the modules connected to the PC will be listed on the left screen of VxCAN Utility, such as I-7530, tM-7530. Among them, when the module in Init mode, and VxCAN Utility will add "-Init" information after the module name and display it in the list.

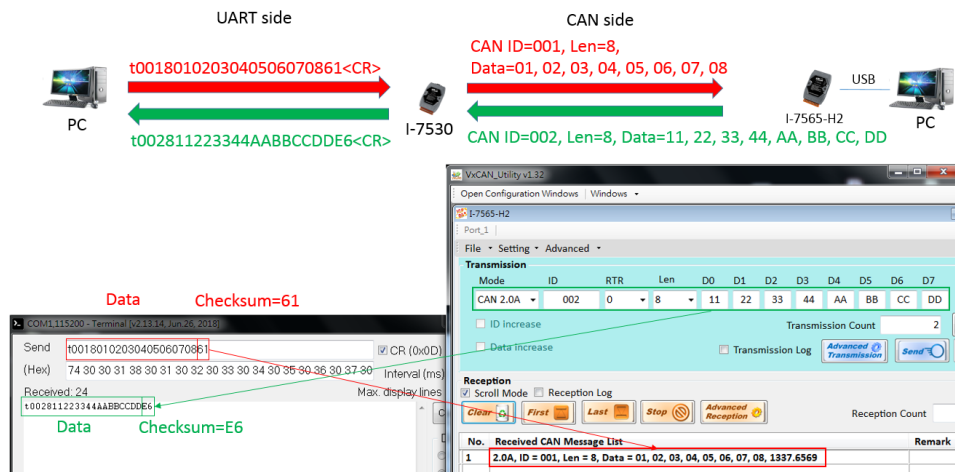


- Step 5. Click the node of "tM-7530 -Init" in the search list to display the setting screen. The setting screen is divided into three areas with different colors with UART setting, CAN setting and communication mode setting area. All parameters will be written to the EEPROM of the module after clicking the "Save All Setting" button.



## 3.2 UART setting

- \* **Baud Rate:** Set the UART baud rate. Different CAN converters have different upper baud rate limits.
- \* **Data Bit:** Set the UART data bit, the CAN converters support four data bit settings.
- \* **Stop Bit:** Set the UART stop bit, the CAN converters support two data bit settings.
- \* **Parity Bit:** Set the UART parity bit check, the CAN converter supports three data bit settings.
- \* **Add Checksum:** Set the CAN converter to enable the checksum for the UART command string, and only supports the communication mode is the normal mode. If the function be set to enable, when the UART command string is transceived or received need to add two check characters to be checksum after the string as below figure below. In addition, please refer to section 4.1 for the calculation method of the checksum.



- \* **Error Response:** Set the CAN converter to enable the error response for the UART command string which in chapter 4, and the function only supports the communication mode is the normal mode. If the function be set to enable, the module will respond with an error code when the user enters an incorrect UART command string. In addition, please refer to section 4.13 for error codes.



\* **Timestamp Response:** Set the CAN converter to enable the timestamp response for the UART command string which in chapter 4. The function only supports modules such as tM-7530, tM-7530A and tM-7565, and only supports the communication modes are normal mode and listen only mode. If the function be set to enable, when the CAN converter converts the CAN message to a UART command string will add a timestamp of 9 ASCII characters (unit is 10us) after the string. In the other word, if the function be set to disable, the UART will not add timestamp after command string.

In addition, because the timestamp reponse is enabled, the module will add timestamp after the UART command string, so it is suitable for users to use the COM port software to parse the data after receiving data as below figure 3.1.2.1.

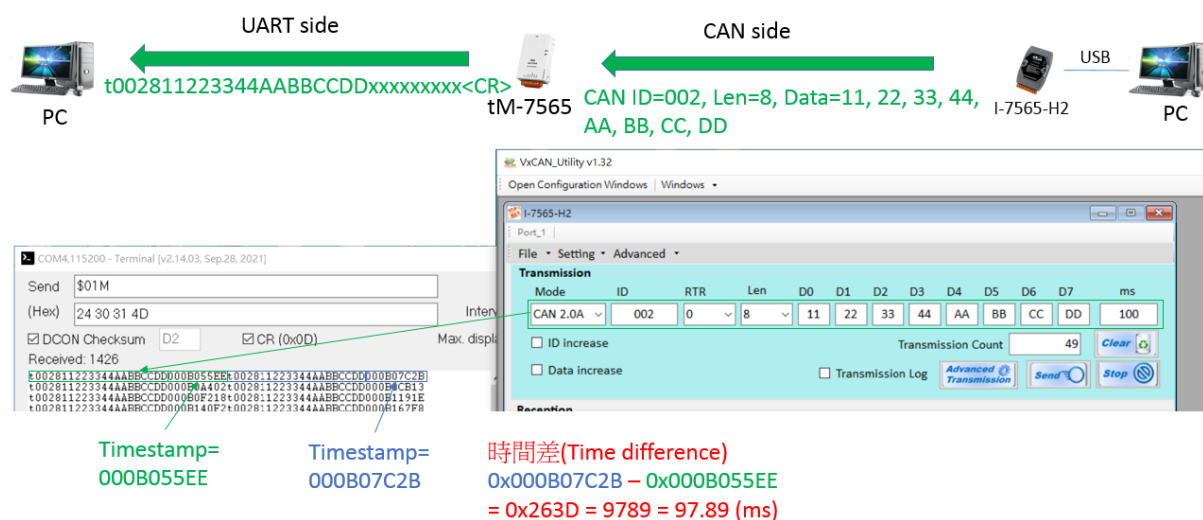
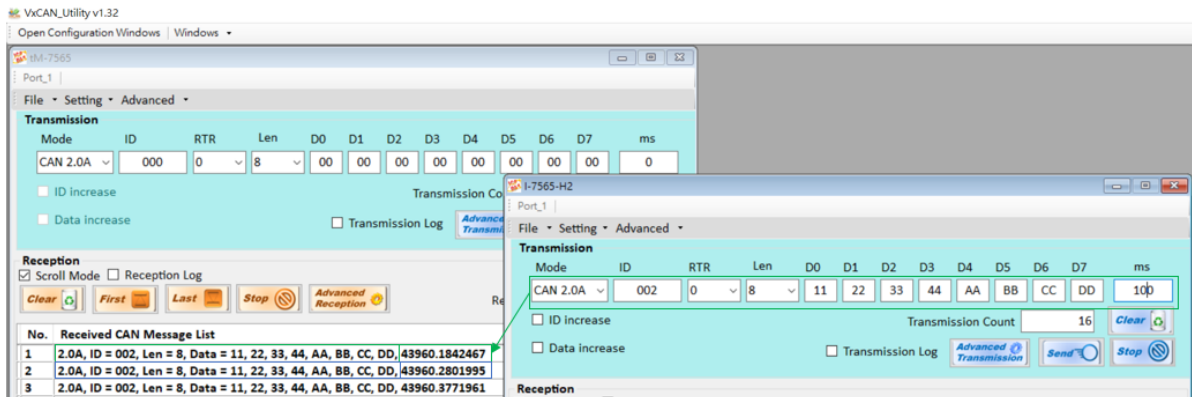


Figure 3.1.2.1 the test screen of the COM port software

However, if the user only need to monitor CAN messages with time stamps can use VxCAN Utility to receive data from the UART, because VxCAN Utility will add timestamp regardless of whether the module supports timestamp response, which means that, in addition to tM-7530, tM-7530A and tM-7565, it can also be applied to modules such as I-7530, I-7530A, I-7565 and I-7565-FT as below Figure 3.1.2.2.

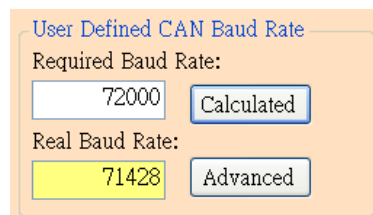


Timestamp = 43960.1842467 時間差(Time difference)  
 Timestamp = 43960.2801995 43960.2801995 - 43960.1842467  
 = 0.0959528 (sec) = 95.9528 (ms)

Figure 3.1.2.2. the test screen of VxCAN Utility

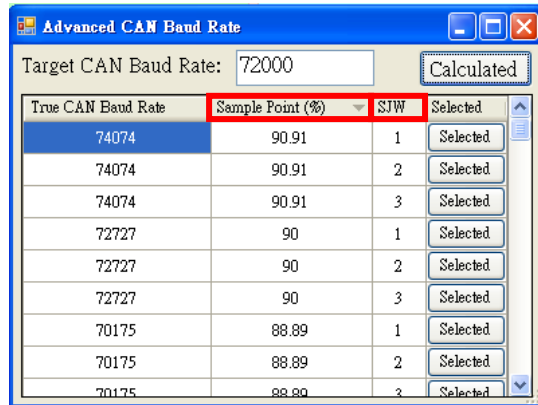
### 3.3 CAN setting

- \* **CAN specification:** Use the CAN specification CAN 2.0A (11-bit CAN ID) or CAN 2.0B (29-bit CAN ID).
- \* **CAN baud rate:** Set the CAN baud rate, which can support 10K, 20K, 50K, 125K, 250K, 500K, 800K, 1000K and 83.3K bps. If these baud rates cannot meet the user's needs, and the special baud rates can be set through the "User-defined CAN baud rate" function.
- \* **User Defined CAN Baud Rate:** When using the user-defined CAN baud rate, the user needs to fill in the value of the CAN baud rate in the "Required Baud Rate" field, and click the "Calculated" button to calculate the "Real Baud Rate". When the calculated real baud rate and the required baud rate are different, it means that the required baud rate cannot be calculated due to hardware limitations, so the "Real Baud Rate" field will display the actual closest and can support The CAN baud rate. As below figure, the required baud rate is 72000bps, but the calculated real baud rate that can be supported is 71428bps.

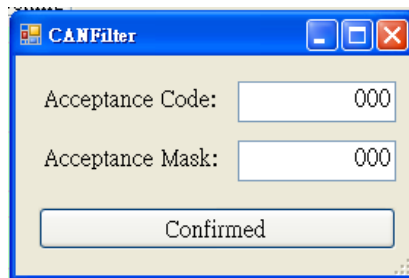


The screenshot shows a dialog box titled "User Defined CAN Baud Rate". It contains two rows of input fields and buttons. The first row is labeled "Required Baud Rate:" and has a text box containing "72000" and a button labeled "Calculated". The second row is labeled "Real Baud Rate:" and has a text box containing "71428" and a button labeled "Advanced".

If the user want to get more information about the real baud rate, please click the "Advanced" button and select the appropriate real baud rate to suit the application as beolw figure. Among them, the field of Sample Point refers to in the CAN message, when the time width of a bit is defined as 0~100% from the start point to the end point, the sample point is which position within this time width. For example: 87.5% means that the sample point is in the 87.5% position of 0~100%. In addition, SJW is the setting for adjusting the amplitude of the sample point, which is usually used to solve the problem of oscillator phase offset between different CAN devices, and ensure the CAN device can receive correct CAN messages. In general, the commonly used sample point and SJW are 87.5% and 1.



\* **CAN ID Filter:** Clicking this button will open the setting window of the CAN ID filter as below figure, the user can limited the CAN converter only receives which CAN ID messages.

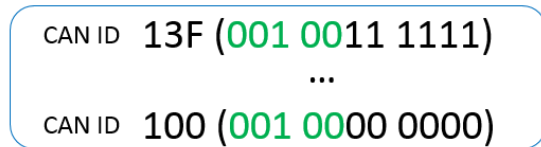


When the CAN converter receives the CAN message, the CAN ID filter will decide whether to check the bit value in the same position with the CAN ID according to the bit value in a certain position of acceptance mask (ACM). If the bit value of a position in the ACM is 1, the bit value in the same position of the CAN ID will be checked; if the bit value of a position in the ACM is 0, the bit value in the same position of the CAN ID will not be checked, for example: the ACM is 7C0 (111 1100 0000 bit) which means CAN ID only bit 10 to 6 will be checked, and bit 5 to 0 are not checked.

Next, after determining the ACM will check which bits of the CAN ID, the CAN converter will only receive the same CAN ID as the acceptance code (ACC) in these bits, for example: when the ACM is 7C0, The CAN ID filter will only check the bit 10 to 6 of the CAN ID, and the ACC is 100 (001 0000 0000 bit), which means that the CAN converter will only receive CAN messages with CAN IDs from 100 to 13F, as below figure.

In other words, if the ACM is set to 000, it means that all CAN ID messages are received without checking the CAN ID, and the content of the ACC can be ignored at this time. Please refer to section 5.5 for details on how the ACC and ACM are calculated.

CAN converter will receive these CAN ID's message



These bits same with acceptance code

Acceptance Code 100 (001 0000 0000)  
 Acceptance Mask 7C0 (111 1100 0000)

bit 10 9 8 7 6 5 4 3 2 1 0

Check these bits Don't check these bits

## 3.4 Communcation mode setting

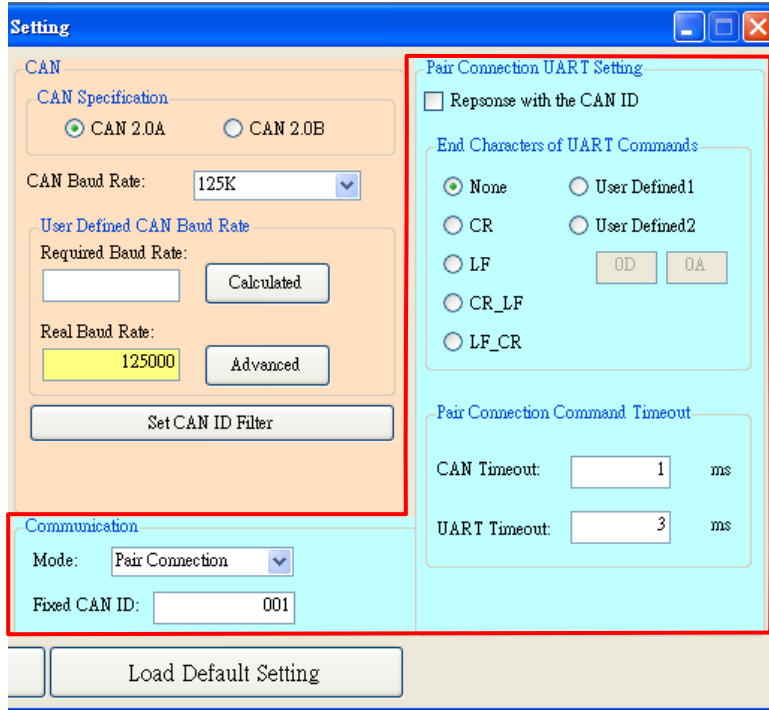
This function is used to set the communication mode of the CAN converter. In normal mode, the UART interface of the CAN converter can only receive the command string as described in chapter 4. In the other word, when the CAN converter receives the CAN messages and convert to the UART interface will also displayed the command string. By the way, any commands input UART interface don't follow the command string format will be regarded as an error message.

In addition, in pair connection mode, the user needs to set a fixed CAN ID in the VxCAN Utility. When the CAN converter receives the UART data will fill in the UART data into the data field of the CAN message, and use this CAN ID to send the CAN message to the CAN side. Among them, the fixed CAN ID is helpful for the application of transparent transmission and pair connection, not only to know the CAN message comes from which CAN converter, but also to avoid to appear the same CAN ID in the CAN network.

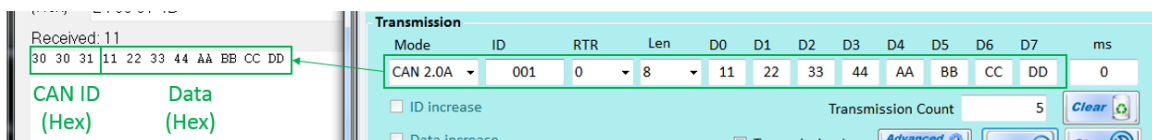
tM-7530, tM-7530A and tM-7565 have an additional communication mode: Listen Only. When using this mode, the tM-7530, tM-7530A and tM-7565 will only receive messages and will not transmit any CAN signals to the CAN network, including Error Frame and ACK signal.

Among them, pair connection mode can according to the application select parameter settings, so the following is a further description of the pair connection mode, including "Fixed CAN ID", "Response with CAN ID", "End Characters of UART Commands" and "Pair Connection Command Timeout".

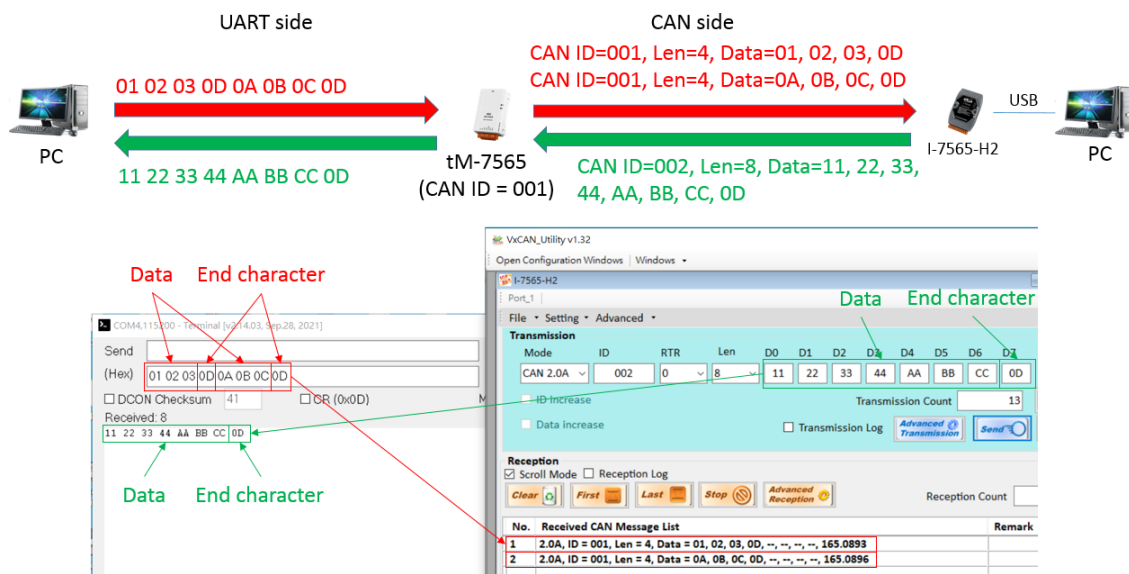
- ◆ **Pair Connection Mode:** When the pair connection mode is selected, the setting fields on the right will expand as below figure, and the order will be explained as follows.



- \* **Fixed CAN ID:** Set a fixed CAN ID. Because in the CAN network, each CAN converter must have a different CAN ID to avoid multiple converters with the same CAN ID to send CAN messages to the CAN network at the same time, and the messages of the same CAN ID to collide and cause errors. If the CAN specification selected by the user is CAN 2.0A, the CAN ID is 11-bit ID, and the range is 000~7FF. If the CAN specification selected by the user is CAN 2.0B, the CAN ID is 29-bit ID, and the range is 00000000~1FFFFFFF.
- \* **Response with CAN ID:** The user can set whether to add the CAN ID before the data and send to the UART side. The CAN ID is ASCII format, for example: CAN ID is 001, when it is sent to the UART side and will display 0x30 0x30 0x31 as below below.



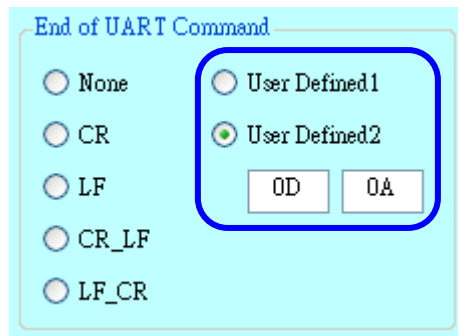
\* **End Characters of UART Commands:** The CAN converter will according to the end character to judge whether the data reception is completed. If the reception is completed, the CAN converter will add the end character to the received data and send it to the other side as below figure. For example: if the end character is CR (0x0D), as long as the CAN converter receives CR, it will be regarded as the completion of data reception, and the received data will be added to the end character, and then sent to the other side.



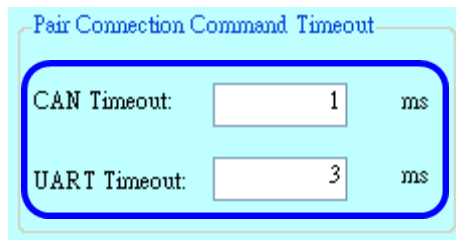
- ▶ **None:** The end character is not used, so the CAN converter will transmit the received data to the other side until a "Pair Connection Command Timeout" occurs, or the firmware buffer is full.
- ▶ **CR:** Set the end character as CR, and the hexadecimal value of ASCII code is "0x0D".
- ▶ **LF:** Set the end character as LF, and the hexadecimal value of ASCII code is "0x0A".
- ▶ **CR\_LF:** Set the end character as CR and LF, and the hexadecimal values of ASCII code are "0x0D" and "0x0A".
- ▶ **LF\_CR:** Set the end character as LF and CR, and the hexadecimal values of ASCII code are "0x0A" and "0x0D".



- ▶ **User-defined:** This function allows the user to define the ending character, and only support tM-7530, tM-7530A and tM-7565. Among them, "User Defined1" and "User Defined2" can set 1 end character and 2 end characters respectively as beolw figure, when "User Defined2" is selected and the end characters are set to 0x0D and 0x0A, when the CAN converter receive 0x0D and 0x0A, and the received data will be added to the end character and then sent to the other side.



- \* **Pair Connection Command Timeout:** Set the waiting time when receiving data, and only support tM-7530, tM-7530A and tM-7565. When the CAN converter receives a byte data, it will wait for this time to confirm whether new data has been received. If no new data is received after this time will determined that the data is received completely, and will send the data to the other side.



"CAN Timeout" can set the timeout time for converting CAN data to UART data, and "UART Timeout" can set the timeout time for converting UART data to CAN data. If the user wants to send the data immediately, the value can be set to 0.

Among them, the function of "UART Timeout" will only work when "End Characters of UART Commands" is set to "None". In addition, about the application of the pair connection mode, please refer to section 5.6.

### 3.5 Saving and default value




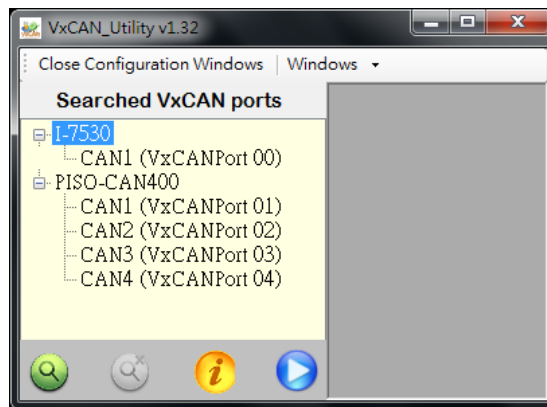
After the user selects the parameters in the "setting screen", and click the "Save All Setting" button to write the parameters into the EEPROM of the CAN converter, but if the user wants to restore the parameters to the factory default values, please click the "Load Default Setting" button to restore all the parameters of the "Setting Screen" to the default values, and then click the "Save All Settings" button to write the parameters into the EEPROM of the CAN converter. Finally, set the Init/Normal of DIP switch to Normal, and power on the CAN converter again to start using these parameters.

The default values of CAN converter as below table:

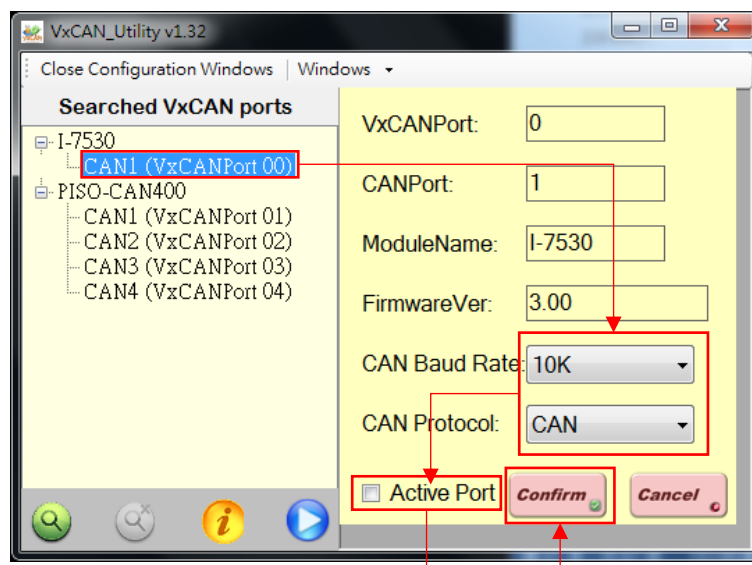
UART side	
Baud Rate	<ul style="list-style-type: none"> <li>● 115200 : I-7530、I-7530A、tM-7530、tM-7530A、tM-7565 and I-7530-FT</li> <li>● 921600 : I-7565</li> </ul>
Data Bit	8
Stop Bit	1
Parity Bit	None
Add Checksum	Disable
Error Response	Disable
Timestamp Response	Disable
CAN side	
CAN specification	2.0A
CAN baud rate	125k
acceptance code (ACC)	000
acceptance mask (ACM)	000
Communication mode	
mode	Normal mode


## 3.6 Testing interface

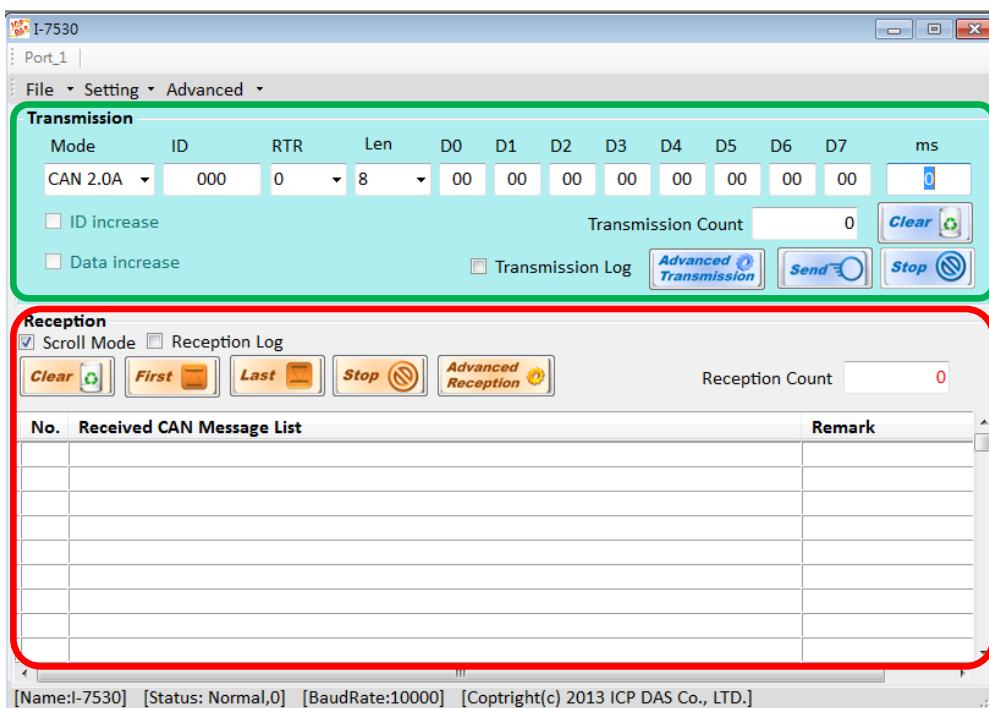
- Step 1. Set the Init/Normal of DIP switch in the CAN converter to Normal mode, and power on the CAN converter again.
- Step 2. Connect the COM port of the PC to the CAN converter and execute the VxCAN Utility.
- Step 3. Click the search button  to search for all CAN converters connected to the PC.
- Step 4. The search list on the left will list the CAN modules that have been searched by the PC.



- Step 5. Click the CAN communication port of the CAN converter and select the parameters be used of the CAN converter, and then check the “Active Port” and click the “Confirm” button.



- Step 6. Click the “Start” button  to open the testing interface. Among them, the testing interface is divided into two areas, which are the transmission area and the reception area of CAN messages. If the user wants to send CAN messages, please fill in the data in the fields in the transmission area and click the "Send" button. In addition, when the CAN converter receives the CAN message, it will automatically display the CAN message in the reception area. For more detailed operation of VxCAN Utility, please refer to the user manual of VxCAN Utility.



Transmission area

Reception area

## 4. UART command string

When the DIP switch is in Normal and the communication mode is in normal mode, the module supports the following command strings in ASCII format that allow the user to send and receive CAN messages on the UART side through these command formats, or set module parameters. The command format as below:

Command format: <Command>[CHK]<CR>

<Command> : RS-232/RS-485/RS-422/USB commands for CAN converter

[CHK] : 2-character checksum value. The checksum will only be activated when this function is enabled using the CAN converter tool software. About the algorithm for checksum, please refer to section 4.1.

<CR> : The commands of the CAN converter on the UART side must be added with the character "<CR>" (ASCII decimal value is 13, hexadecimal value is 0x0D) at the end

The command strings list as below, and the details are described in the following subsections.

Table: List of command strings

Command	Description
tIIILDD...[CHK]<CR>	Transmit or receive standard data frame
TIIIL[CHK]<CR>	Transmit or receive standard remote frame
eIIIIIIILDD...[CHK]<CR>	Transmit or receive expanded data frame
EIIIIIIIL[CHK]<CR>	Transmit or receive expanded remote frame
S[CHK]<CR>	Read the CAN baud rate and error message of the CAN converter.
C[CHK]<CR>	Clear error flag or overload in register of CAN converter
P0BBDSPA[E][CHK]<CR>	Set the parameter configuration and storage of the UART side
P1B[CHK]<CR>	Set the parameter configuration and storage of the CAN side

P2BBDSPAЕ[CHK]<CR>	Set the parameter configuration of the UART side, but do not store it
P3SBCCCCCCCCMMMMMMMM[CHK]<CR>	Set the parameter configuration of the CAN side, but do not store it
RA[CHK]<CR>	Restart the CAN converter module

## 4.1 Checksum algorithm

The checksum algorithm is that convert all the characters of the command string (the first character to the characters before [CHK]) into ASCII hexadecimal values, and take out the lowest 2 digit value after adding up, and then convert this value back to an ASCII string, for example as below.

Example:

Command: Restart CAN converter module, RA[CHK]<CR>.

1. String sum = 'R' + 'A' = 52h + 41h = 93h.
2. Therefore, the checksum is 93h, then [CHK] = "93".
3. The command string contains the checksum is "RA93<CR>".

## 4.2 tIII LDD...[CHK]<CR>

➤ **Description:** Transmit or receive standard data frame.

➤ **Syntax:** tIII LDD...[CHK]<CR>

t	Represent a standard (2.0A) data frame.
III	11 bits Identifier (000~7FF)
L	Data length (0~8)
DD...	Input data frame value according to the data length (00~FF)

➤ **Response:**

Valid command: No response

Invalid command: ?<Error Code><CR>

➤ **Note:**

In order to communicate with the computer to receive error information, the user must enable the "Error Response" function in the CAN converter software.

➤ **Example:**

Command: t03F6112233445566<CR>

Send a CAN message with a standard data frame. ID=03F, DLC=6, data1=11, data2=22, data3=33, data4=44, data5=55 and data6=66.



## 4.3 TIIIL[CHK]<CR>

➤ **Description:** Transmit or receive standard remote frame.

➤ **Syntax:** TIIIL[CHK]<CR>

T Represents a standard (2.0A) remote frame.

III 11 bits Identifier (000~7FF)

L Data length (0~8)

➤ **Response:**

Valid command: No response

Invalid command: ?<Error Code><CR>

➤ **Note:**

In order to communicate with the computer to receive error information, the user must enable the "Error Response" function in the CAN converter software.

➤ **Example:**

Command: T2E88<CR>

Send a CAN message with a standard remote frame.

ID=2E8, DLC=8.

## 4.4 eIIIIIIILDD...[CHK]<CR>

➤ **Description:** Transmit or receive expanded data frame

➤ **Syntax:** eIIIIIIILDD...[CHK]<CR>

e	Stands for the extended (2.0B) data frame.
IIIIIIII	29 bits Identifier (00000000~1FFFFFFF)
L	Data length (0~8)
DD...	Input data frame value according to the data length (00~FF)

➤ **Response:**

Valid command: No response

Invalid command: ?<Error Code><CR>

➤ **Note:**

In order to communicate with the computer to receive error information, the user must enable the "Error Response" function in the CAN converter software.

➤ **Example:**

Command: e1234567851122334455<CR>

Send a CAN message with an extended data frame. ID=12345678, DLC=5, data1=11, data2=22, data3=33, data4=44 and data5=55.

## 4.5 EIIIIIIIL[CHK]<CR>

➤ **Description:** Transmit or receive expanded remote frame.

➤ **Syntax:** EIIIIIIIL[CHK]<CR>

E Stands for the extended (2.0B) CAN remote frame.

IIIIIIII 29 bits Identifier (00000000~1FFFFFFF)

L Data length (0~8)

➤ **Response:**

Valid command: No response

Invalid command: ?<Error Code><CR>

➤ **Note:**

In order to communicate with the computer to receive error information, the user must enable the "Error Response" function in the CAN converter software.

➤ **Example:**

Command: E010156786<CR>

Send a CAN message with an extended remote frame.

ID=01015678, DLC=6.

## 4.6 S[CHK]<CR>

➤ **Description:** Read the CAN baud rate and error message of the CAN converter.

➤ **Syntax:** S[CHK]<CR>

**S** Command character.

➤ **Response:**

Valid Command: !CFFTTRRO[CHK]<CR>

Invalid command: ?<Error Code>[CHK]<CR>

**!** Delimiter for valid command

**C** current baud rate setting of CAN

**FF** CAN control register, expressed as bits.

**TT** CAN transmit error counter, expressed as a value

**RR** CAN receive error counter, expressed as a value

**O** CAN or RS-232/RS-485/RS-422/USB overload warning, expressed as bits

➤ **Note :**

In order to communicate with the computer to receive error information, the user must enable the "Error Response" function in the CAN converter software. In addition, all returned results are in ASCII format, so the user must convert ASCII to hexadecimal as below table.

Table: CAN baud rate

AsciiToHex(C)	Description
0	10K CAN baud rate
1	20K CAN baud rate
2	50K CAN baud rate
3	100K CAN baud rate
4	125K CAN baud rate
5	250K CAN baud rate, I-7530-FT not supported
6	500K CAN baud rate, I-7530-FT not supported
7	800K CAN baud rate, I-7530-FT not supported

8	1000K CAN baud rate, I-7530-FT not supported
9	83.3K CAN baud rate, I-7530-FT not supported
A	User-defined CAN baud rate, I-7530-FT not supported

Table: CAN control register

AsciiToHex(FF)	Description
Bit 7	Bus Off Mode
Bit 6	Error Passive Mode
Bit 5	Reserved
Bit 4	I-7530, I-7530A, I-7565 and I-7530-FT: Overrun Buffer tM-7530, tM-7530A and tM-7565: Reserved
Bit 3	produces Stuff error
Bit 2	produces CRC error
Bit 1	produces Form error
Bit 0	produces Acknowledgment error

Table: CAN Converter FIFO overflow flag

AsciiToHex(O)	Description
Bit 3	Reserved
Bit 2	Reserved
Bit 1	RS-232/RS-485/RS-422/USB FIFO Overflow
Bit 0	CAN FIFO Overflow

➤ **Example:**

Command: S<CR>

Receive: !50000000<CR>

Obtain some current information from the CAN converter module, and the response will show the following results: CAN baud rate=250K, CAN register= normal, CAN transmit error counter=0, CAN receive error counter=0 and CAN/RS232 FIFO= normal.

## 4.7 C[CHK]<CR>

➤ **Description:** Clear error flag or overload in register of CAN converter.

➤ **Syntax:** C[CHK]<CR>

C                    Command character

➤ **Response:**

Valid Command: No response, and will turn the ERR LED off after the ERR LED was turned on as a result of the CAN Converter FIFO overflow error flag.

Invalid command: ?<Error Code>[CHK]<CR>

➤ **Note:**

In order to communicate with the computer to receive error information, the user must enable the "Error Response" function in the CAN converter software.

➤ **Example:**

Command: C<CR>

## 4.8 P0BBDSPAE[CHK]<CR>

➤ **Description:** Set the parameter configuration and storage of the UART side.

➤ **Syntax:** P0BBDSPCE[CHK]<CR>

<b>P0</b>	Command character
<b>BB</b>	RS-232/RS-485/RS-422/USB Baud rate
<b>D</b>	Data bit 0 = 5 bits Data formation 1 = 6 bits Data formation 2 = 7 bits Data formation 3 = 8 bits Data formation
<b>S</b>	Stop bit (0=1 stop bit, 1=2 stop bits)
<b>P</b>	Parity (0=None, 1=Odd, 2=Even)
<b>C</b>	Checksum (0=No, 1=Yes)
<b>E</b>	Error Response (0=No, 1=Yes);

Table: RS-232/RS-485/RS-422/USB's UART baud rate

Code BB	Description
00	UART baud rate 110 bps
01	UART baud rate 150 bps
02	UART baud rate 300 bps
03	UART baud rate 600 bps
04	UART baud rate 1200 bps
05	UART baud rate 2400 bps
06	UART baud rate 4800 bps
07	UART baud rate 9600 bps
08	UART baud rate 19200 bps
09	UART baud rate 38400 bps
0A	UART baud rate 57600 bps
0B	UART baud rate 115200 bps
0C	UART baud rate 230400 bps

Code BB	Description
	(Only supports I-7565, tM-7530, tM-7530A and tM-7565)
0D	UART baud rate 460800 bps (Only supports I-7565 and tM-7565 after firmware version v3.9)
0E	UART baud rate 921600 (Only supports I-7565 and tM-7565 after firmware version v3.9)

➤ **Response:**

Valid command: No response, and will automatically restart after writing the configuration parameters of the UART

Invalid command: ?<Error Code><CR>

➤ **Note:**

In order to communicate with the computer to receive error information, the user must enable the "Error Response" function in the CAN converter software. In addition, in firmware 2.X or earlier, the I-7565 only supports a baud rate of 921600 bps.

➤ **Example:**

Command: P00B30000<CR>

Set RS-232/RS-485/RS-422/USB baud rate=115.2K, data bit=8, stop bit=1, none parity, none checksum to the CAN converter and no error return, the CAN converter will reboot and use these parameters



## 4.9 P1B [CHK]<CR>

➤ **Description:** Change the CAN configuration of CAN Converter module. Module will be rebooted after saving these parameters into EEPROM successfully.

➤ **Syntax: P1B[CHK]<CR>**

**P1** Command character

**B** CAN Baud rate

Code B	Description
0	CAN baud rate 10K
1	CAN baud rate 20K
2	CAN baud rate 50K
3	CAN baud rate 100K
4	CAN baud rate 125K
5	CAN baud rate 250K, I-7530-FT not supported
6	CAN baud rate 500K, I-7530-FT not supported
7	CAN baud rate 800K, I-7530-FT not supported
8	CAN baud rate 1000K, I-7530-FT not supported
9	CAN baud rate 83.3K, I-7530-FT not supported
A	User-defined CAN baud rate, I-7530-FT not supported

➤ **Response:**

Valid command: No response, and will write CAN configuration parameters to the module and restart automatically

Invalid command: ?<Error Code><CR>

➤ **Note:**

In order to communicate with the computer to receive error information, the user must enable the "Error Response" function in the CAN converter software. This command does not support user-defined CAN baud rate.

➤ **Example:**

Command: P14<CR>

Set the CAN baud rate=125K, and then reboot the module.

## 4.10 P2BBDSPAE[CHK]<CR>

➤ **Description:** Set the parameter configuration of the UART side such as RS-232/RS-485/RS-422/USB, and these parameters will not be saved. When the module is restarted, these parameters will be restored to the previous settings.

➤ **Syntax: P2BBDSPCE[CHK]<CR>**

<b>P2</b>	Command character
<b>BB</b>	RS-232/RS-485/RS-422/USB Baud rate
<b>D</b>	Data bit 0 = 5 bits Data formation 1 = 6 bits Data formation 2 = 7 bits Data formation 3 = 8 bits Data formation
<b>S</b>	Stop bit (0=1 stop bit, 1=2 stop bits)
<b>P</b>	Parity (0=None, 1=Odd, 2=Even)
<b>C</b>	Checksum (0=No, 1=Yes)
<b>E</b>	Error Response (0=No, 1=Yes)

Table: RS-232/RS-485/RS-422/USB Baud Rate

Code BB	Description
00	UART baud rate 110 bps
01	UART baud rate 150 bps
02	UART baud rate 300 bps
03	UART baud rate 600 bps
04	UART baud rate 1200 bps
05	UART baud rate 2400 bps
06	UART baud rate 4800 bps
07	UART baud rate 9600 bps
08	UART baud rate 19200 bps
09	UART baud rate 38400 bps
0A	UART baud rate 57600 bps

Code BB	Description
0B	UART baud rate 115200 bps
0C	UART baud rate 230400 bps (Only supports I-7565, tM-7530, tM-7530A and tM-7565)
0D	UART baud rate 460800 bps (Only supports I-7565 and tM-7565 after firmware version v3.9)
0E	UART baud rate 921600 (Only supports I-7565 and tM-7565 after firmware version v3.9)

➤ **Response:**

Valid command: No response, and will change and not save the configuration parameters on the UART side

Invalid command: ?<Error Code><CR>

➤ **Note:**

In order to communicate with the computer to receive error information, the user must enable the "Error Response" function in the CAN converter software.

➤ **Example:**

Command: P20B30000<CR>

Set RS-232/RS-485/RS-422/USB baud rate=115.2K, data bit=8, stop bit=1, none parity, none checksum to the CAN converter module and no error return. When the module is rebooted, the parameters are restored to previous settings.

## 4.11 P3SBCCCCCCCCMMMMMMMM [CHK]<CR>

➤ **Description:** Set the parameter configuration of the CAN side, but do not store it

➤ **Syntax:** P3SBCCCCCCCCMMMMMMMM[CHK]<CR>

**P3**                    Command character  
**S**                    CAN Specification 2.0A/2.0B  
**B**                    CAN Baud rate  
**CCCCCCCC**        CAN ID filter acceptance code (ACC)  
**MMMMMMMM**        CAN ID filter acceptance mask (ACM)

Table: CAN baud rate

Code B	Description
0	CAN baud rate 10K
1	CAN baud rate 20K
2	CAN baud rate 50K
3	CAN baud rate 100K
4	CAN baud rate 125K
5	CAN baud rate 250K, I-7530-FT not supported
6	CAN baud rate 500K, I-7530-FT not supported
7	CAN baud rate 800K, I-7530-FT not supported
8	CAN baud rate 1000K, I-7530-FT not supported
9	CAN baud rate 83.3K, I-7530-FT not supported

➤ **Response:**

Valid command: No response, and the CAN bus configuration parameters will be changed.

Invalid command: ?<Error Code><CR>

➤ **Note :**

In order to communicate with the computer to receive error information, the user must enable the "Error Response" function in the CAN converter software. For the CAN ID filter setting, please refer to "Set CAN ID Filter" in Section 3.3.

➤ **Example:**

Command: P30400000000000000000<CR>

Change CAN baud rate=125K, accept code=0x00000000, accept mask=0x00000000. When the module is rebooted, the parameters are restored to previous settings.

## 4.12 RA[CHK]<CR>

➤ **Description:** Restart the CAN converter module.

➤ **Syntax:** RA[CHK]<CR>

RA Command character

➤ **Response:**

Valid command: No response, and will reboot the CAN converter module.

Invalid command: ?<Error Code><CR>

➤ **Note:**

In order to communicate with the computer to receive error information, the user must enable the "Error Response" function in the CAN converter software.

➤ **Example:**

Command: RA<CR>

When the CAN converter module receives this command and will reboot.

## 4.13 Error Code

If the CAN converter has enabled the "Error Response" function of the UART setting in section 3.2, the CAN converter will reply error messages to the device or computer via RS-232/RS-485/RS-422/USB. The meaning of the error code message as below table:

Table: Error code

Error code	Description	Possible causes & solutions
1	Invalid header	RS-232/RS-485/RS-422/USB command character headers are not the following command characters: "t","T","e","E","S","C","P0 ","P1","P2","P3" or "RA"
2	Invalid length	The data byte of the CAN message does not conform to the CAN data length, for example: Error: t001512345<CR> Correct: t00150102030405<CR>
3	Invalid checksum	The checksum of the RS-232/RS-485/RS-422/USB command string is error, for example: Error: t0012112209<CR> Correct: t00121122FD<CR>
4	Reserved	None.
5	Timeout	The ASCII command string was not sent completely, for example, the <CR> ending character was missing. Error: T0018 Correct: T0018<CR>

When the CAN converter wants to transfer the data from the UART side to the CAN side, if the baud rate of the CAN is different from the baud rate of the CAN network, the ON LED of the CAN converter will flash at a constant frequency, and the CAN converter cannot send the any CAN message to the CAN network.

In addition, the user can use the command "S[CHK]<CR>" to check the status of the CAN converter to confirm what is wrong. Generally speaking, the common problems

are as follows:

1. CAN connection problems.
2. terminal resistance problems.
3. problems with different CAN baud rates.



## 5. Appendix

This chapter attaches common problems and solutions to help users quickly to check problems.

## 5.1 Cable selection

The CAN bus is a balanced (differential) 2-wire interface running over either a Shielded Twisted Pair (STP), Un-shielded Twisted Pair (UTP), or Ribbon cable. The CAN-L and CAN-H wire start on one end of the total CAN network that a terminator resistor of 120 Ohm is connected between CAN-L and CAN-H. The cable is connected from CAN node to CAN node, normally without or with short T connections. On the other end of the cable again a 120Ω (Ohm) terminator resistor is connected between the CAN lines.



**Figure: Un-shielded Twisted Pair (UTP)**

How to decide a cable type, cable length, and terminal resistor depends on the baud rate in the CAN BUS network, please refer to the following table.

CAN baud rate	Cable type	Cable resistance / meter	terminator resistor	Bus Length
50k bit/s , 1000m	0.75~0.8mm <sup>2</sup> 18AWG	70 mOhm	150~300 Ohm	600~1000m
100k bit/s , 500m	0.5~0.6 mm <sup>2</sup> 20AWG	< 60 mOhm	150~300 Ohm	300~600m
500k bit/s , 100m	0.34~0.6mm <sup>2</sup> 22AWG, 20AWG	< 40 mOhm	127 Ohm	40~300m
1000k bit/s , 40m	0.25~0.34mm <sup>2</sup> 23AWG, 22AWG	< 40 mOhm	124 Ohm	0~40m

## 5.2 I-7565 Driver Installation Instructions

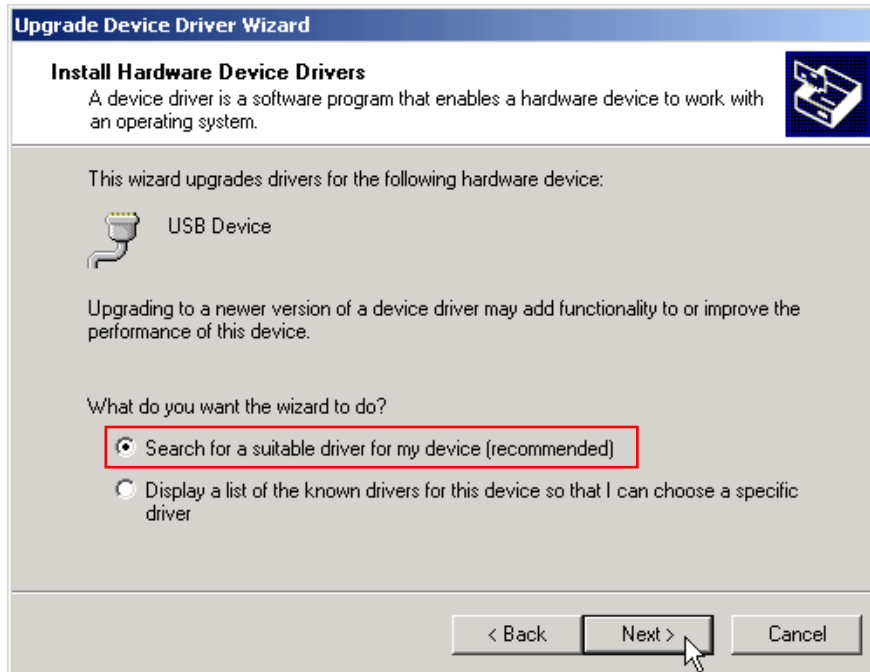
This chapter will describe how to install the driver for the I-7565 converter on Windows XP, 2000, ME, 98 (NT not yet supported) operating platforms. Among them, the driver can be downloaded from the Download Center of the I-7565 product webpage. The product webpage as beolw: <https://www.icpdas.com/en/product/I-7565-G>

Next explains how to install the driver in Windows 2000, and this installation process is applicable to other Windows operating systems, and the installation steps are as below.

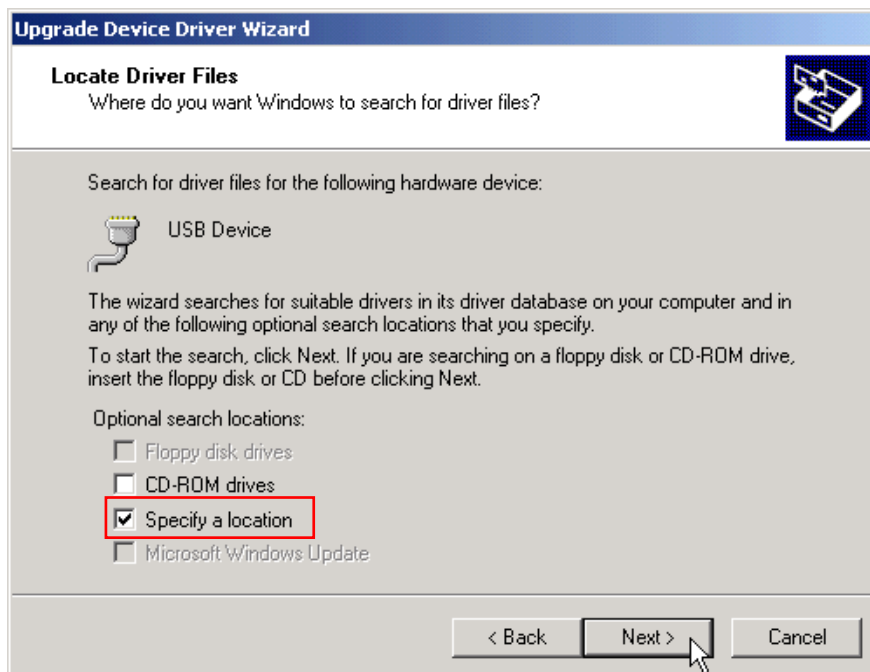
1. Please turn on your computer to the Windows operating platform first.
2. Connect the USB port of the I-7565 to the USB port of the computer.
3. When Windows detects the USB device will appear a screen to promote you to install the device driver, and then click Next to continue the installation.



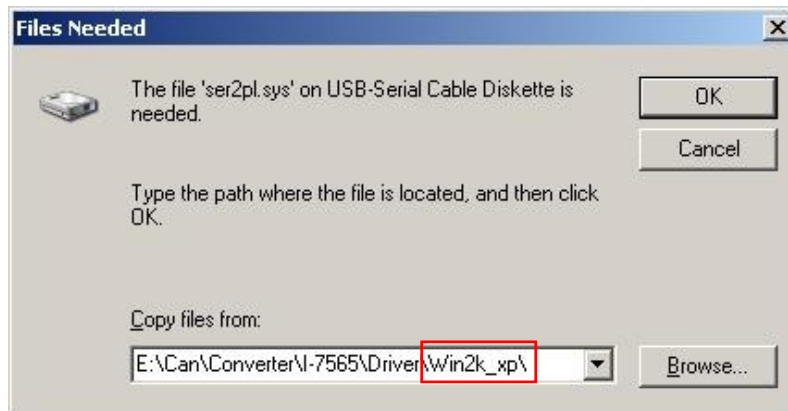
4. The "Install Hardware Device Drivers" window will appear on the screen. After selecting "Search for a suitable driver for my device [recommended]", click "Next" to search for the most suitable driver for the new device.



5. After selecting the "Specify a location" option, click "Next" to start to search.



6. For the file path, please select this folder win2k\_xp, and this folder will be in the folder of the driver you downloaded, and then click "OK" to start to search.



7. After Windows has found the correct driver, click "Next" to start to install the driver.



8. Windows will install the USB-to-Serial driver, and notify that the installation is complete after the installation is complete. At this time, please click "Finish" and re-plug the USB cable.

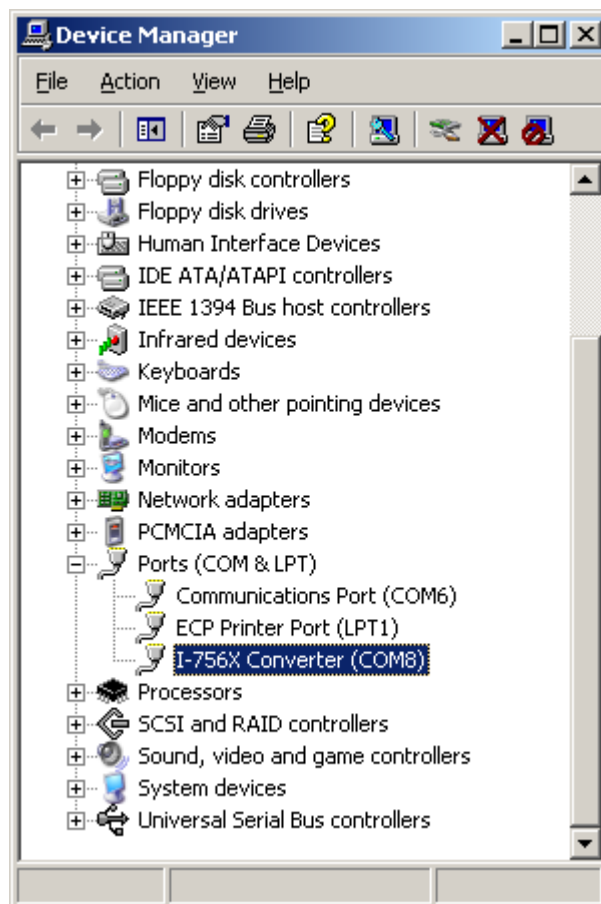


## 5.2.1 Verifying the installation

This chapter explains how to verify whether the I-7565 has been installed and confirm the virtual COM port number assigned to the I-7565 by Windows.

Please connect the USB port of the I-7565 to the USB port of the computer, and then perform the following steps:

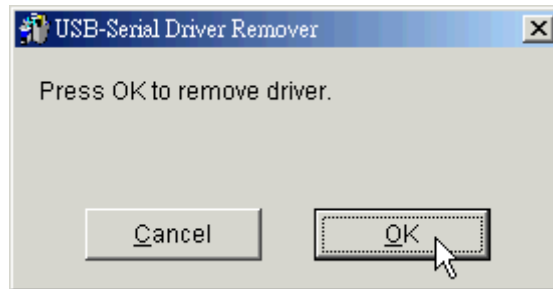
1. Click “Start” on the Windows toolbar → “Settings” → “Control Panel” → “System” and after the system content window is displayed, click “Hardware” → “Device Manager”.
2. Double-click “Ports (COM & LPT)”, if the device has been installed, the information of “I-756X Converter (COM8)” will be displayed in the device list. Among them, COM 8 is the virtual COM port number assigned by the computer to the I-7565, and this number may be different on each computer, please use the number assigned by the computer as the main one.



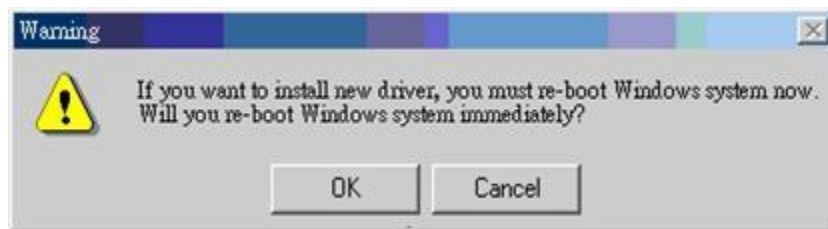
## 5.2.2 Remove the driver

The steps to remove the driver for the I-7565 as below:

1. Execute the removal program dremover98\_2K.exe in the driver folder.
2. When you want to remove the utility, and the Uninstaller will prompt you click "OK" to continue.



3. After the software has been removed, the program will prompt you to restart Windows, please click "OK" to continue.



4. After successfully removing the driver software, Windows will display a dialog window of "Remove driver successfully", please restart the computer.



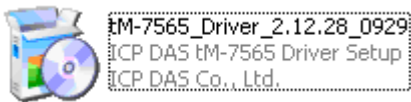


## 5.3 tM-7565 Driver Installation Instructions

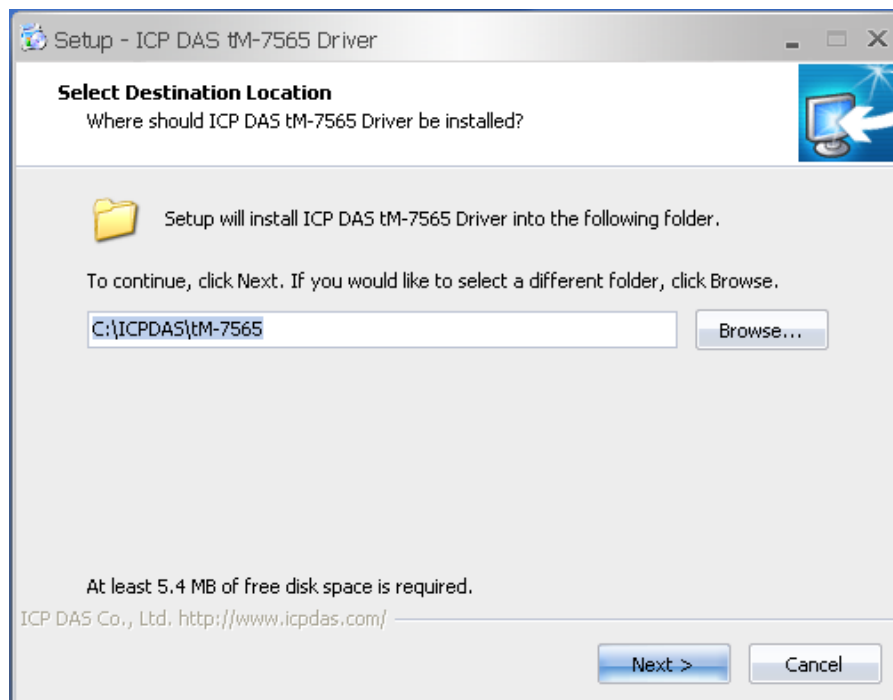
This chapter will describe how to install the driver (x32, x64) of tM-7565 on Windows XP, Windows 7, Windows 10 operating system. Among them, the driver can be downloaded from the Download Center of the tM-7565 product webpage. The product webpage as below: <https://www.icpdas.com/en/product/tM-7565>

Next will explain how to install the driver in Windows XP. The installation process is applicable to other Windows operating systems, and the installation steps as below.

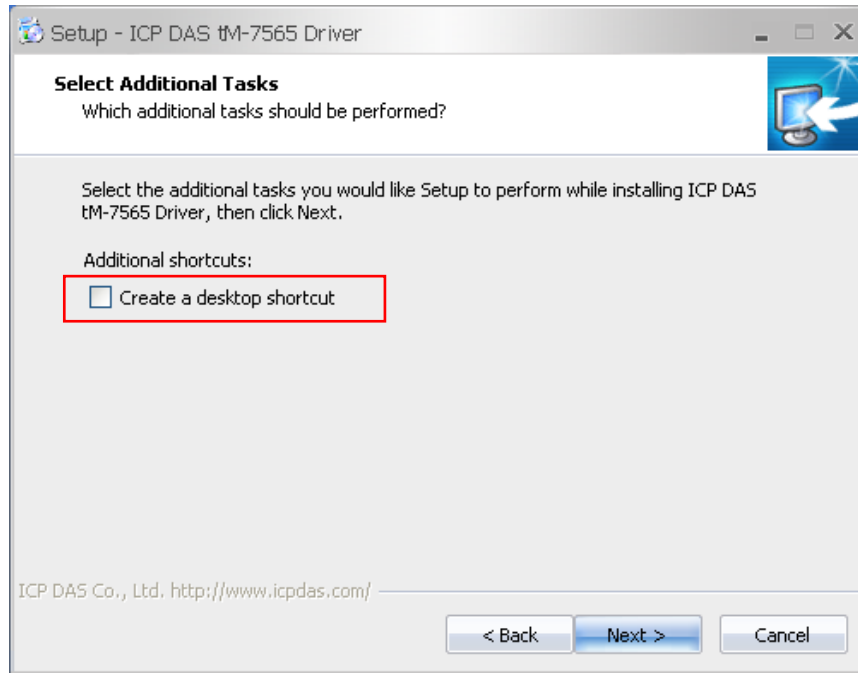
1. Please turn on your computer to the Windows operating platform first.
2. Click on the driver file and install the tM-7565 driver.



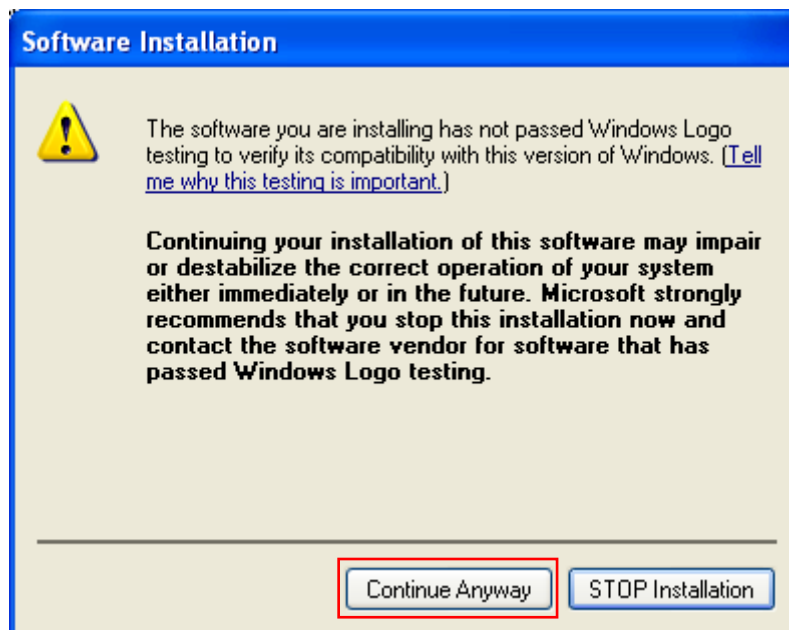
3. After clicking, the installation wizard will ask for the driver installation location. After confirming the installation path, and click the "Next" button to proceed to the next step.



4. In this step, you can choose whether to create a desktop shortcut. If you need to create a desktop shortcut, please check the "Create a desktop shortcut" option, and then click the "Next" button, if you do not need to create a desktop shortcut, please click the "Next" button directly.



5. During the installation process, the following warning message will be displayed. Please click the "Continue Anyway" button.



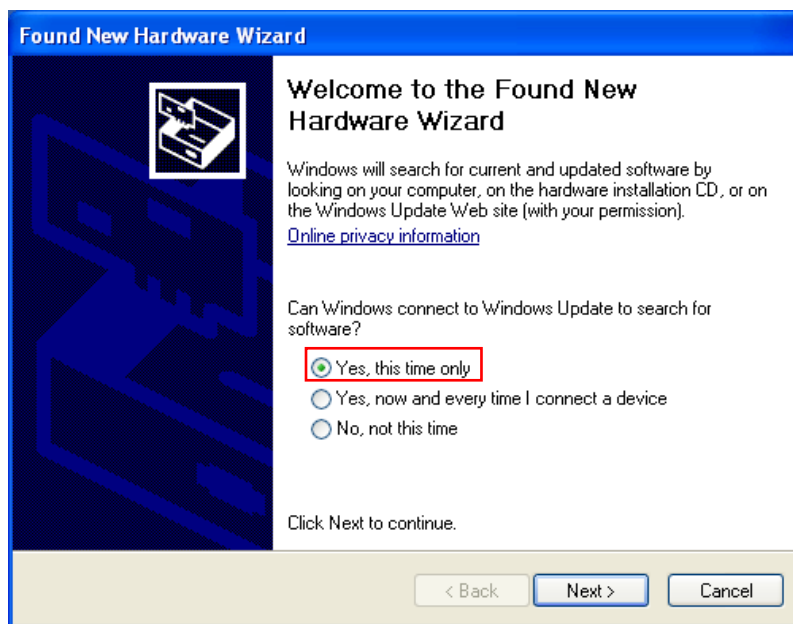
6. After the driver installation is complete, please click the "finish" button.



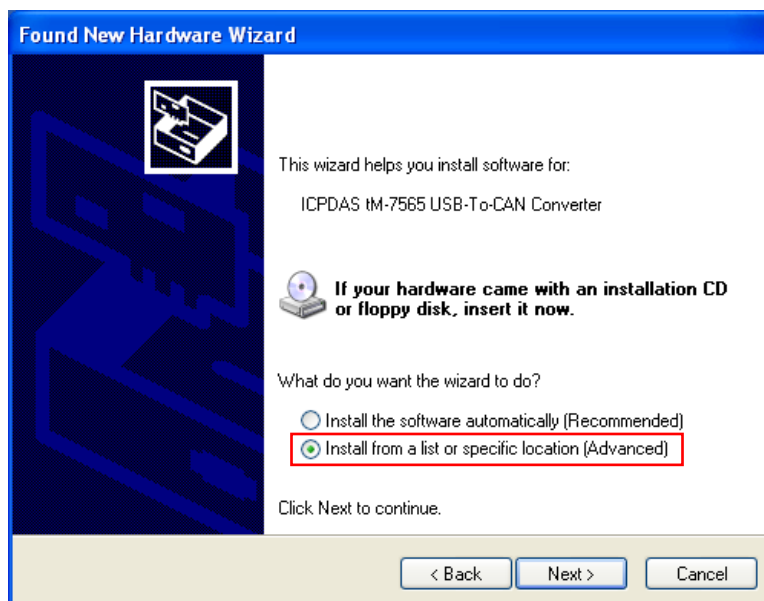
### 5.3.1 Installing the hardware

The hardware installation will install twice, once for the hardware driver and once for the virtual COM port driver. The detailed steps as below:

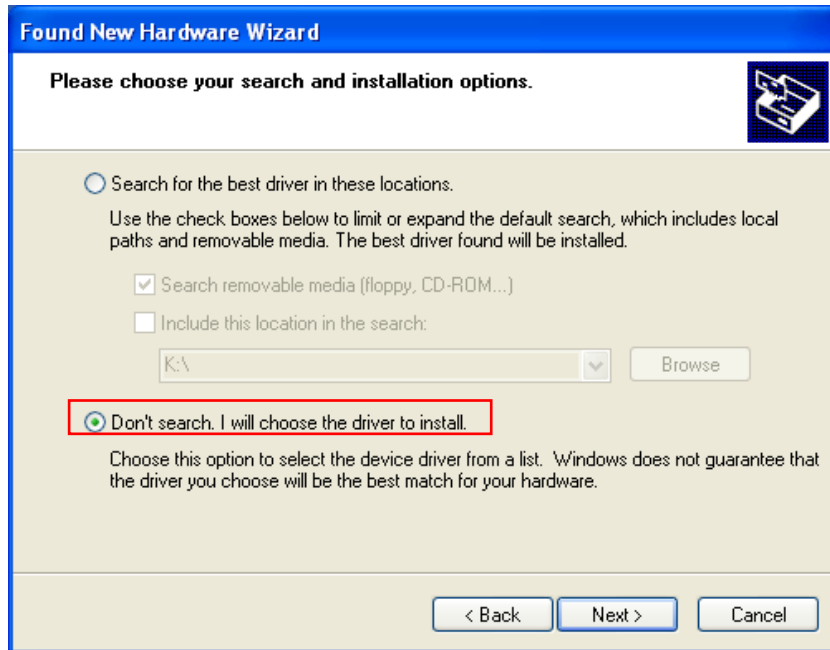
1. Connect the USB port of the tM-7565 to the USB port of the computer.
2. After Windows detects the new USB device will appear a screen to promote you to install the device driver. Please select "Yes, this time only" and click the "Next" button.



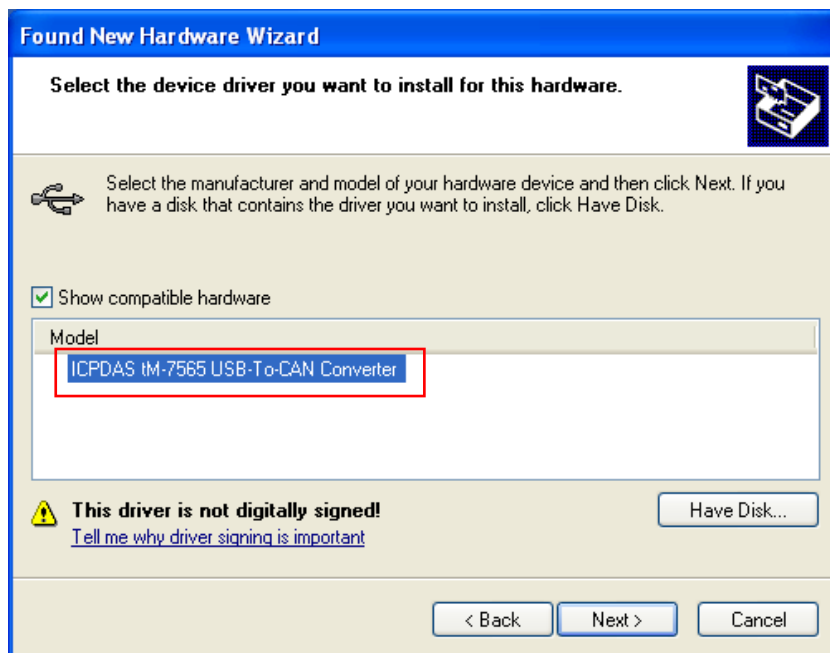
3. Select "Install from a list or specific location [Advanced]" and click the "Next" button to search for the most suitable driver for the new device.



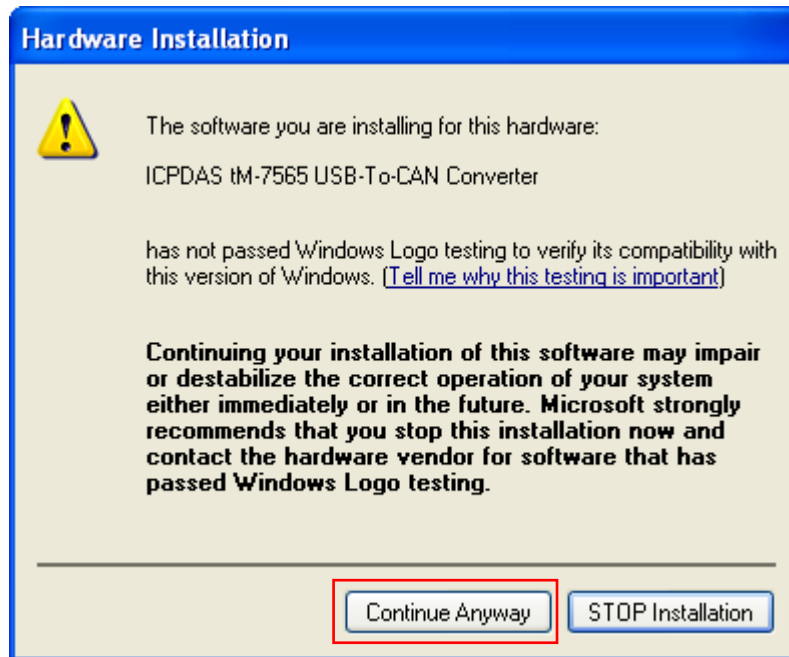
4. Select "Don't search. I will choose the driver to install" and click the "Next" button.



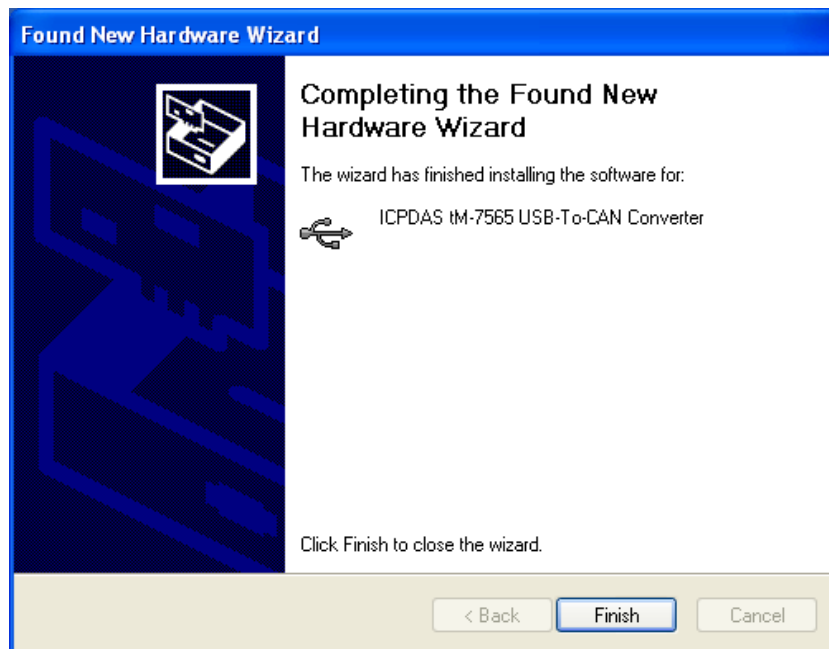
5. Select "ICPDAS tM-7565 USB-To-CAN Converter" and click the "Next" button.



6. During the installation process will appear the following warning message, please click the "Continue Anyway" button.



7. After the installation is complete, please click the "finish" button.

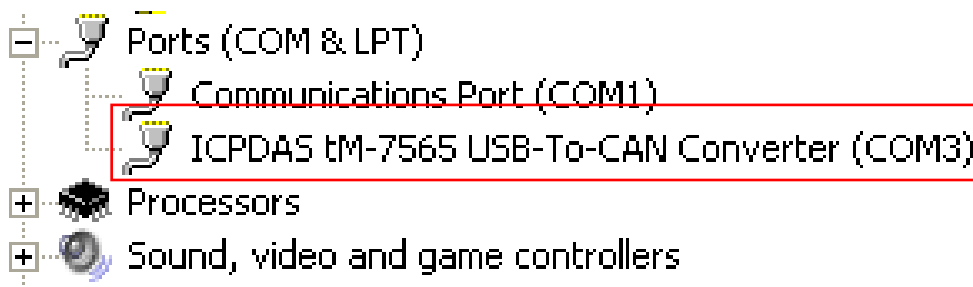


### 5.3.2 Verifying the installation

This chapter explains how to verify whether the tM-7565 has been installed and confirm the virtual COM port number assigned to the tM-7565 by Windows.

Please connect the USB port of the tM-7565 directly to the USB port of the computer, and then perform the following steps:

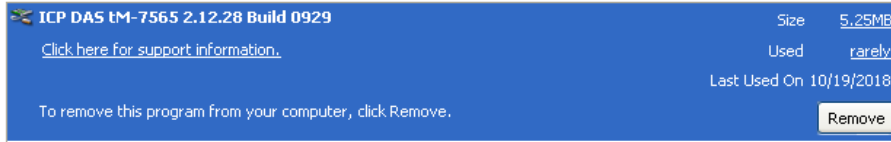
1. Click “Start” on the Windows toolbar → “Settings” → “Control Panel” → “System, and after the system content window is displayed, click “Hardware” → “Device Manager”.
2. Double-click “Ports (COM & LPT)”, if the device has been installed, the information of “ICPDAS tM-7565 USB-To-CAN Converter (COM3)” will be displayed in the device list. Among them, COM 3 is the virtual COM port number assigned by the computer to the I-7565, and this number may be different on each computer, please use the number assigned by the computer.



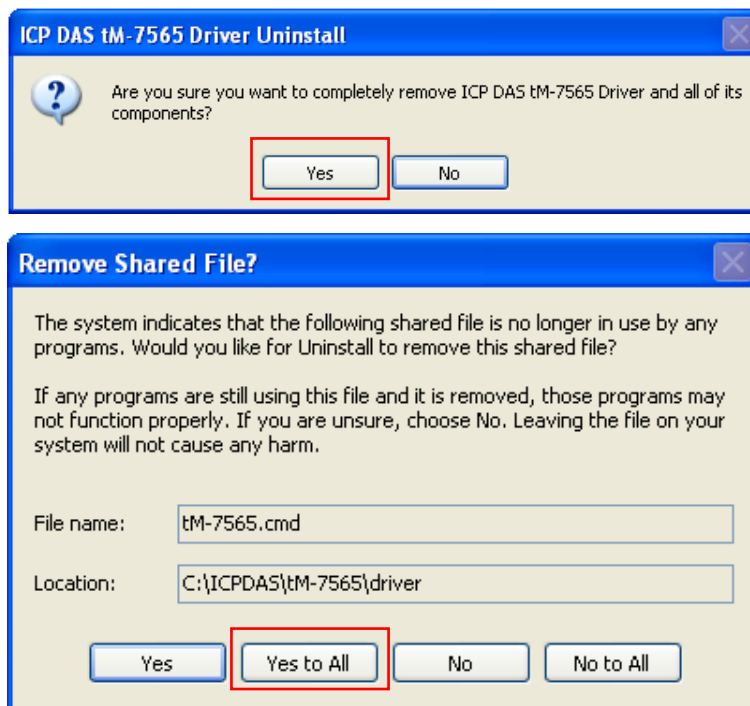
### 5.3.3 Remove the driver

The steps to remove the driver for the tM-7565 as below:

1. "Control Panel" → "Add Remover", and then click the "Remove" button.



2. After uninstalling the installer will appear a prompt window, and please click the "Yes" and "Yes to All" buttons to remove the tM-7565 driver.



3. Click "OK" to complete the uninstallation process.





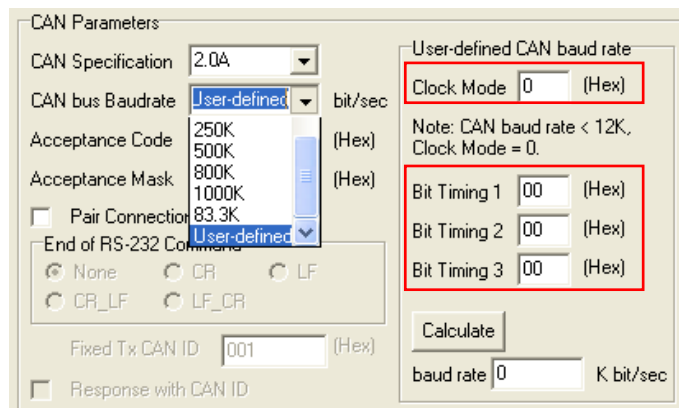
## 5.4 Calculation of user-defined CAN baud rate

This chapter explains how to calculate the "user-defined CAN baud rate" when the user uses an old version of the tool software (CAN Converter Utility) for setting, and the calculation only support I-7530, 7530A and I-7565, but not for I-7530 (v2 version), I-7530A (v2 version), I-7565 (v2 version), tM-7530, tM-7530A, tM-7565, I-7530-FT and I-7530-FT (v2 version).

In addition, if the user uses the VxCAN Utility to set the user-defined CAN baud rate, please refer to the description in "User-defined CAN baud rate" in section 3.3, and the VxCAN Utility can automatically calculate the CAN baud rate for the user, and then the user clicks "Save All Setting" to write the CAN baud rate into the module.

### 5.4.1 Setting parameters of CAN Converter Utility

The function needs to set 4 parameters, and namely Clock Mode, Bit Timing 1, Bit Timing 2 and Bit Timing 3 as below window:



1. Clock Mode : This parameter is the clock of the MCU. Assuming that the CAN baud rate is less than 12Kbps, the value of this parameter is 0, and the rest are 1.
2. Bit Timing 1 : This parameter is the bit timing register 1 of the MCU, please refer to Section 5.4.2.
3. Bit Timing 2 : This parameter is the bit timing register 2 of the MCU, please refer to Section 5.4.2.
4. Bit Timing 3 : This parameter is the bit timing register 3 of the MCU, please refer to Section 5.4.2.

### 5.4.2 Parameter description

The Bit Timing register of MCU contains 6 kinds of parameters, namely BPR, SJW, PRS, PHS2, PHS1 and SMP as below table:

Table: Bit Timing Register

Bit register \ Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bit Timing 1	-	BRP_5	BRP_4	BRP_3	BRP_2	BRP_1	BRP_0	-
Bit Timing 2	-	SJW_1	SJW_0	-	PRS_2	PRS_1	PRS_0	-
Bit Timing 3	-	PHS2_2	PHS2_1	PHS2_0	PHS1_2	PHS1_1	PHS1_0	SMP

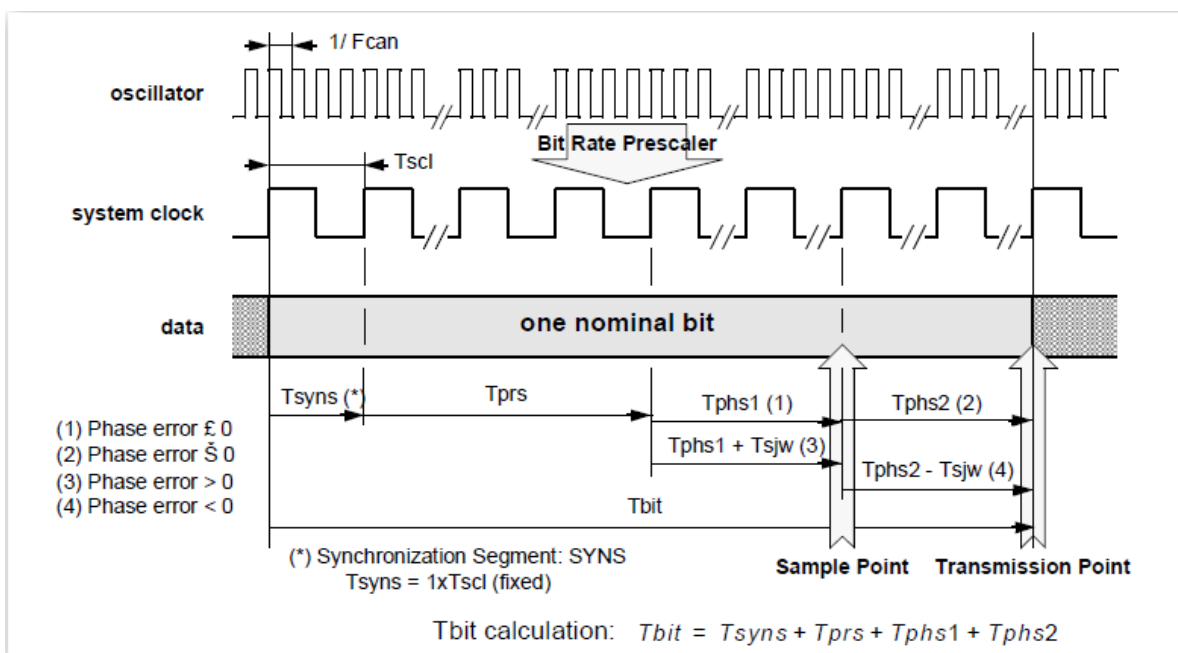


Figure: Structure of Bit period

In general, the SJW0, SJW1 of the register Bit Timing 2 and SMP of register Bit Timing 3 are all 0, and the time width  $T_{bit}$  of one bit is composed of  $T_{syns}$ ,  $T_{prs}$ ,  $T_{phs1}$  and  $T_{phs2}$ . Among them, the time width can be divided to several time slices (TQ) based on the system clock.

The formula as below:

$$\text{Number of Time Slices (TQ)} = \text{System Clock (Hz)} / \text{CAN Baud Rate (bps)}$$

For example: when the system clock is 20MHz and the CAN baud rate is 40K bps, the number of time slices (TQ) is  $20(\text{MHz}) / 40(\text{Kbps}) = 500$ , that is to say that one bit of the time width Tbit is divided into 500 time slices in the CAN converter, and these time slices are allocated to Tsyns, Tprs, Tphs1 and Tphs2. Among them, Tsyns is defined to one time slice, and the time slices of Tprs, Tphs1 and Tphs2 need to refer the number be set of PRS, PHS1 and PHS2 in the Bit Timing register, the calculation method as below.

$$T_{\text{prs}} = (\text{PRS} + 1) * \text{TQ}$$

$$T_{\text{phs1}} = (\text{PHS1} + 1) * \text{TQ}$$

$$T_{\text{phs2}} = (\text{PHS2} + 1) * \text{TQ}$$

Therefore, the number of time slices Tbit of one bit can be expressed as below that can be divided into  $(4 + \text{PRS} + \text{PHS1} + \text{PHS2})$  time slices.

$$\begin{aligned} T_{\text{bit}} &= T_{\text{syns}} + T_{\text{prs}} + T_{\text{phs1}} + T_{\text{phs2}} \\ &= 1 * \text{TQ} + (\text{PRS} + 1) * \text{TQ} + (\text{PHS1} + 1) * \text{TQ} + (\text{PHS2} + 1) * \text{TQ} \\ &= [1 + (\text{PRS} + 1) + (\text{PHS1} + 1) + (\text{PHS2} + 1)] * \text{TQ} \\ &= (4 + \text{PRS} + \text{PHS1} + \text{PHS2}) * \text{TQ} \end{aligned}$$

However, since the maximum value that can be set for PRS, PHS1 and PHS2 is 3 bits (the value is 7), the number of time slices Tbit of one bit can only be divided into 25 time slices at most that is  $4 + 7 + 7 + 7 = 25$ . As mentioned above, when the number of time slices Tbit of one bit needs to be divided into 500 time slices, the 500 time slices need to be divided by a proportional value to be reduced first, and then allocate the value to the PRS, PHS1 and PHS2. Among them, if the proportional value is represented by  $(\text{BRP}+1)$ , the representation of the time width Tbit of one bit will be modified as below.

$$\begin{aligned} T_{\text{bit}} / (\text{BRP} + 1) &= (4 + \text{PRS} + \text{PHS1} + \text{PHS2}) * \text{TQ} \\ \rightarrow T_{\text{bit}} &= (4 + \text{PRS} + \text{PHS1} + \text{PHS2}) * (\text{BRP} + 1) * \text{TQ} \end{aligned}$$

In addition, when Tbit is fixed at 500, the maximum value of (4 + PRS + PHS1 + PHS2) is limited to 25, and the maximum value of BRP in the register Bit Timing 1 is 63, so (4 + PRS + PHS1 + PHS2) will be in the range of 8~25, and (BRP + 1) will be in the range of 20~62, that is to say, the range of BRP will be in the range of 19~61.

For example, when the system clock is 20MHz and the CAN baud rate is 40K bps, the CAN converter is divided into 500 time slices (TQ) in a time width Tbit of one bit. If (4 + PRS + PHS1 + PHS2) is fixed to 20, the BRP value

$$\begin{aligned} \rightarrow T_{bit} &= 500 * TQ = (4 + PRS + PHS1 + PHS2) * (BRP + 1) * TQ \\ \rightarrow T_{bit} &= 500 / 20 = (BRP + 1) = 25 \\ \rightarrow BRP &= 24 \end{aligned}$$

In addition, because (4 + PRS + PHS1 + PHS2) is fixed at 20, the user can know the time width of one bit that contains the number of time slices is Tbit = 4 + PRS + PHS1 + PHS2 = 20, but now can't determine the values of these three parameters, so necessary to calculate the values of these three parameters according to the proportional value of the sample point as below.

The sample point is mean that the position of the proportion in a bit time width Tbit. That is to say, if a bit time width Tbit is 0~100%, and the sampling point is 60 that mean the position of sampling point is the 60% in a bit time width Tbit. The sampling point be calculated as below:

$$\begin{aligned} \text{Sample point} &= (T_{\text{syns}} + T_{\text{prs}} + T_{\text{phs1}}) / T_{\text{bit}} \\ &= (T_{\text{syns}} + T_{\text{prs}} + T_{\text{phs1}}) / (T_{\text{syns}} + T_{\text{prs}} + T_{\text{phs1}} + T_{\text{phs2}}) \\ &= [1 * TQ + (PRS + 1) * TQ + (PHS1 + 1) * TQ] / TQ * (4 + PRS + PHS1 + PHS2) \\ &= TQ * [1 + (PRS + 1) * TQ + (PHS1 + 1)] / TQ * (4 + PRS + PHS1 + PHS2) \\ &= TQ * (3 + PRS + PHS1) / TQ * (4 + PRS + PHS1 + PHS2) \\ &= (3 + PRS + PHS1) / (4 + PRS + PHS1 + PHS2) \end{aligned}$$

If as mentioned above, the time width of one bit is Tbit = 4 + PRS + PHS1 + PHS2, and the sum must be 20, when the sampling point is 60%, then

$$\begin{aligned} 0.6 &= (3 + PRS + PHS1) / (4 + PRS + PHS1 + PHS2) \\ \rightarrow (3 + PRS + PHS1) &= 0.6 * 20 = 12 \end{aligned}$$

Therefore, if PHS1 is 7, then PRS will be 2. Next, if PHS1 and PRS are brought into the formula of  $(4 + PRS + PHS1 + PHS2) = 20$ , then  $PHS2 = 20 - 4 - 2 - 7 = 7$ .

That is to say, the user can calculate the value of BRP from the number of time slices included in the time width of one bit, and calculate the values of PRS, PHS1 and PHS2 from the sampling point.

### 5.4.3 Example description

If the user want to use the user-defined CAN baud rate is 40K bps and the sampling point is 60%, because the CAN baud rate is greater than 12Kbps, the clock mode needs to be set to 1.

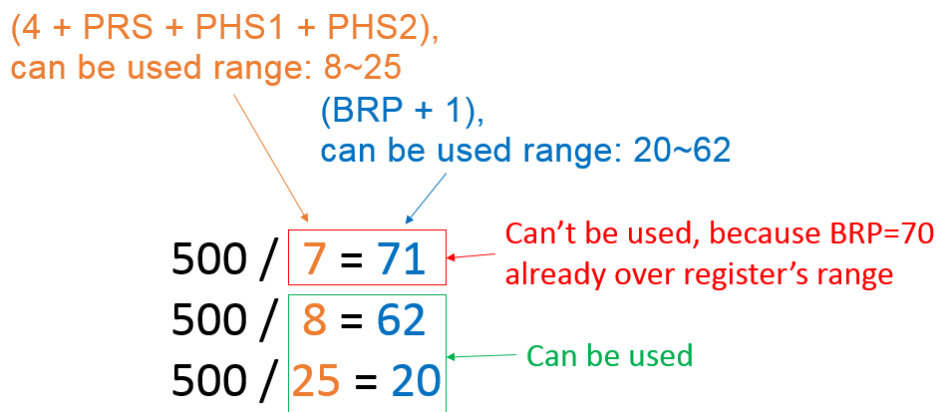
However, when Clock Mode is set to 1, the MCU will use X2 mode, which means the system clock is 20MHz. In addition, if the MCU is in X1 mode, the system clock is 10MHz.

The following steps will guide the user to calculate the parameters of the Bit Timing register.

1. Calculate the number of time slices (TQ) after division:

$$\begin{aligned} \text{Number of time slices (TQ)} &= \text{System clock (Hz)} / \text{CAN baud rate (bps)} \\ &= 20 \text{ M(Hz)} / 40 \text{ k(bps)}, \text{ where Hz is times/second and bps is bits/second} \\ &= 500 \text{ times/bit} \end{aligned}$$

2. Reduce the number of time slices. Because the number of time slices has a limited range in the register, it is necessary to use the BRP value to reduce the scale, as below figure and formula. Among them, as mentioned above, since the number of time slices (4 + PRS + PHS1 + PHS2) of one bit is limited by the maximum setting range of the register, the maximum value of (4 + PRS + PHS1 + PHS2) is 25. At the same time, if considering the setting range of BRP in the register, that is to say, the available range of (4 + PRS + PHS1 + PHS2) is 8~25, so it can be known that the range of the value that can be set by (BRP+1) is 20~62, and the range of BRP values that can be set is 19~61.



If we fix (4 + PRS + PHS1 + PHS2) to 20, the BRP value is:

$$T_{\text{bit}} / (\text{BRP} + 1) = (4 + \text{PRS} + \text{PHS1} + \text{PHS2}) * TQ$$

$$\rightarrow 500 / (\text{BRP} + 1) = 20$$

$$\rightarrow 500 / 20 = (\text{BRP} + 1) = 25$$

$$\rightarrow \text{BRP} = 24$$

3. Find the values of PRS, PHS1 and PHS2 using the sampling points. If it is assumed that the sampling point is 60%, and (4 + PRS + PHS1 + PHS2) in step 2 is fixed to 20, then:

$$\text{取樣點的取樣比例} = (3 + \text{PRS} + \text{PHS1}) / (4 + \text{PRS} + \text{PHS1} + \text{PHS2})$$

$$\rightarrow 60\% = (3 + \text{PRS} + \text{PHS1}) / 20$$

$$\rightarrow 12 = (3 + \text{PRS} + \text{PHS1})$$

Among them, since the maximum value of PRS, PHS1, and PHS2 in the register is 7, if it is assumed that PHS1 is 7, it can be known that PRS is 2, as below formula.

$$12 = (3 + \text{PRS} + \text{PHS1})$$

$$\rightarrow \text{PRS} = 12 - 3 - 7$$

$$\rightarrow \text{PRS} = 2$$

Next, since (4 + PRS + PHS1 + PHS2) is fixed at 20, it can be known that PHS2 is 7, as below formula.

$$(4 + \text{PRS} + \text{PHS1} + \text{PHS2}) = 20$$

$$\rightarrow \text{PHS2} = 20 - 4 - 2 - 7$$

$$\rightarrow \text{PHS2} = 7$$

4. Generally, both SJW and SMP are set to 0. Among them, when SMP is 0, it will be sampled once, and when SMP is 1, it will be sampled three times, and the sampling time interval must be greater than 1/2 Tsc1 period, and the final result is determined by a majority of three values.

Finally, according to the above calculations, the values of the three sets of Bit Timing registers are as below.

The value of Bit Timing 1 register is 0x30

The value of Bit Timing 2 register is 0x04

The value of Bit Timing 3 register is 0x7E

In addition, after the user fills these parameters into the CAN Converter Utility, please click the "Calculate" button to calculate the baud rate, and then click the "Setting" button, the baud rate will be written to the CAN converter.

The screenshot shows the 'CAN Parameters' window. The 'CAN bus Baudrate' is set to 'User-defined' (bit/sec). The 'User-defined CAN baud rate' section includes 'Clock Mode' (1), a note 'Note: CAN baud rate < 12K. Clock Mode = 0.', and three bit timing registers: 'Bit Timing 1' (30), 'Bit Timing 2' (04), and 'Bit Timing 3' (7E). A 'Calculate' button is present, and the resulting 'baud rate' is 40.00 K bit/sec.



## 5.5 CAN ID filter setting

The CAN converter can be set to filter the CAN messages of some CAN IDs, so that these CAN messages will not be transmitted to the RS-232/485/422/USB port. If the user want to enable this function need to adjust parameters such as acceptance code (ACC) and acceptance mask (ACM).

ACC: CAN ID bits to be received.

ACM: CAN ID bits to be filtered.

In the ACC, when the bit is set to 1, it represents the CAN ID bit to be received. In addition, in the ACM, when the bit is set to 1, it represents the CAN ID bit to be filtered.

The example as below:

(1) If want to receive all CAN messages can set to:

In CAN 2.0A : Acceptance Code : Ignorable

Acceptance Mask : 000

(2) If only receiving CAN ID bits of "123" (hexadecimal) and filtering all other messages can set to:

In CAN 2.0A : Acceptance Code : 123 (001 0010 0011 bit)

Acceptance Mask : 7FF (111 1111 1111 bit)

(3) If only get the CAN ID range from "100" to "13F" (hexadecimal), and filter out other CAN messages can set to:

In CAN 2.0A : Acceptance Code : 1XX (001 00XX XXXX bit) (X: Ignorable)

Acceptance Mask : 7C0 (111 1100 0000 bit)

(4) The settings of CAN 2.0A and CAN 2.0B are the same, and the difference between CAN 2.0A and CAN 2.0B is the bit length of CAN ID.

In CAN 2.0A : The CAN ID bit length is 11.

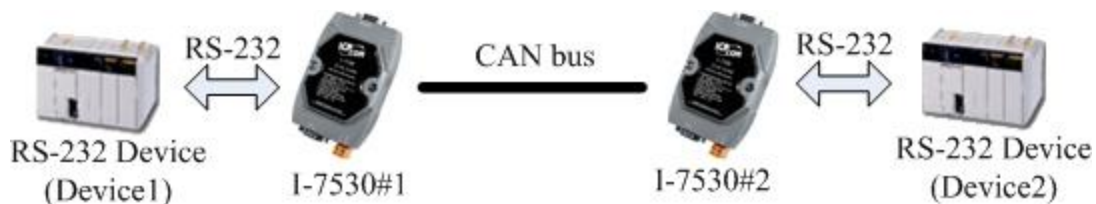
In CAN 2.0B : The CAN ID bit length is 29.

## 5.6 Application of Pair Connection mode

The pair connection mode usually requires two identical sets of CAN converter modules. For example, when two sets of I-7530s are set to pair connection mode, one set of I-7530s will convert RS-232 commands into CAN messages and transmit them, while the other set of I-7530s will transmit CAN messages to RS-232, and the setting of each application will be explained as below.

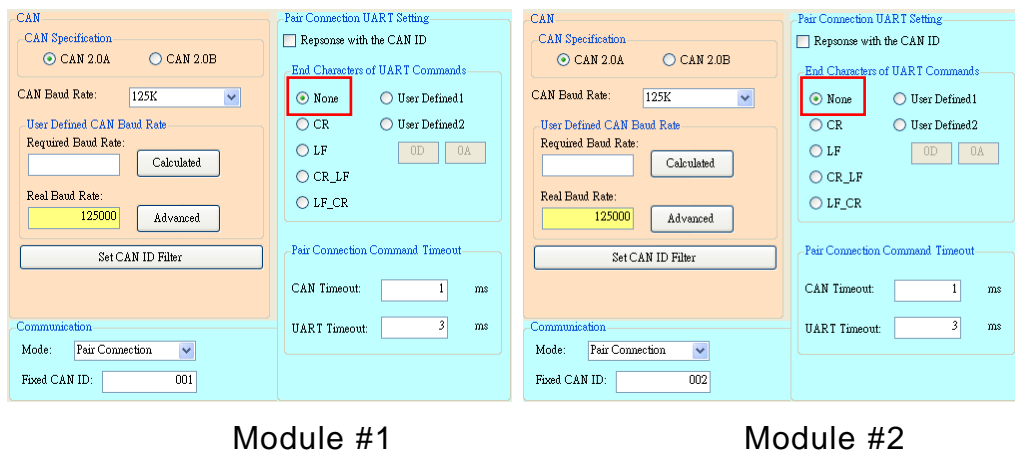
### Application 1 :

This application is commonly used between two sets of devices with RS-232 interface. Because the distance between the devices is too far to use RS-232 to connect directly, so the RS-232 interface is first converted into a CAN interface through a CAN converter, and using the CAN interface to extend the communication distance, and finally switch from the CAN interface to the RS-232 interface.



### Setting:

In this application, the user needs to set up 2 sets of CAN converters, as shown in the above I-7530 module #1 and module #2, and select "None" for the End Characters of UART Commands. In addition, the setting of RS-232 parameters is the same as the RS-232 parameters of the two devices.

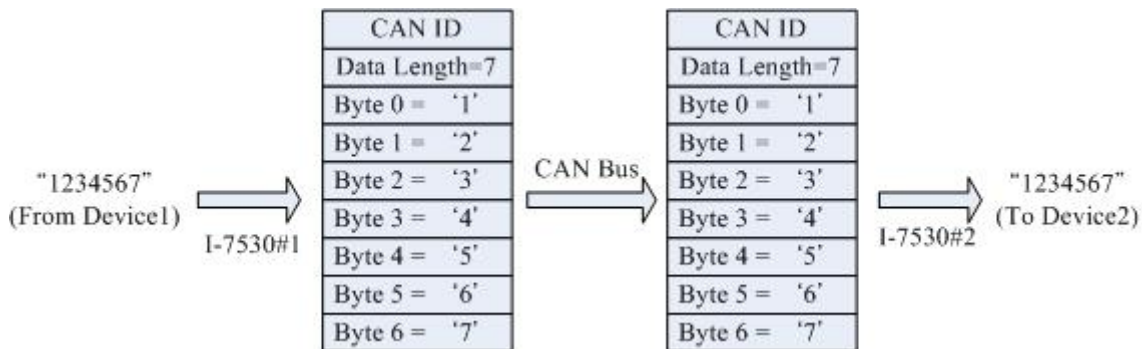


Module #1

Module #2

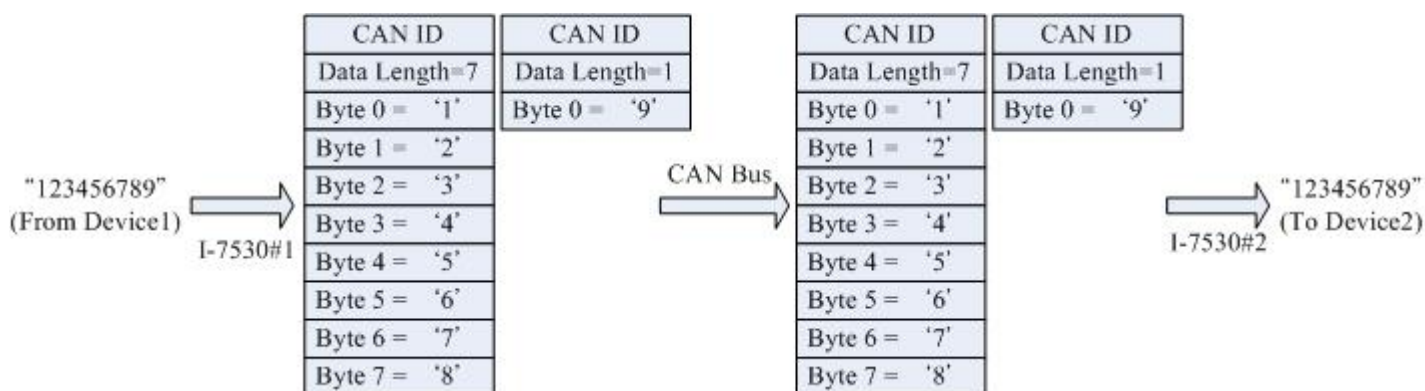
**Communication description:**

For example, if device 1 sends 7 bytes of data "1234567" to I-7530 module #1, I-7530 module #1 then forwards it to I-7530 module #2, and finally I-7530 Module #2 will send to the COM port of Device 2, and Device 2 will receive the data "1234567".



The CAN ID in the above picture needs the user to set a fixed CAN ID, and the "Fixed CAN ID" parameter in the setting screen. If the CAN specification selected by the user is CAN 2.0A and the CAN ID is 11-bit ID, the range is 000~7FF. If the CAN specification selected by the user is CAN 2.0B and the CAN ID is 29-bit ID, the range is 0000000~1FFFFFFF.

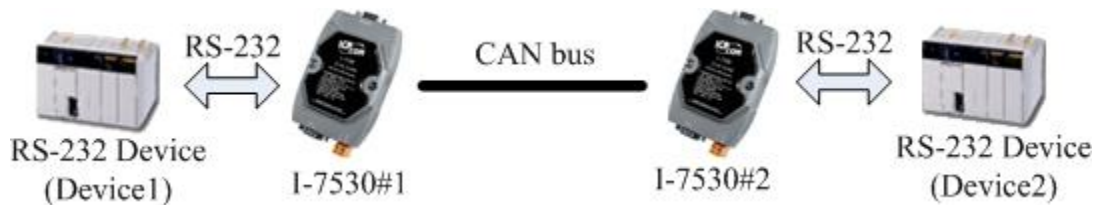
In addition, since a CAN message can only contain 8 bytes of data, if device 1 sends 9 bytes of data "123456789" to I-7530 module #1, I-7530 module #1 will divide the 9 bytes into 2 CAN messages and forwarded to I-7530#2. Finally, I-7530 module #2 merges the 2 CAN messages back to "123456789" and forwards them to the COM port of device 2. Therefore, device 2 will receive the data of "123456789".



Note: If the user uses 115200bps in the CAN RS-232 communication port, it is recommended to configure the baud rate of the CAN side of the CAN converter to be close to the same baud rate, such as 125Kbps. When you are using the pair connection mode, the appropriate baud rate should be below 125Kbps.

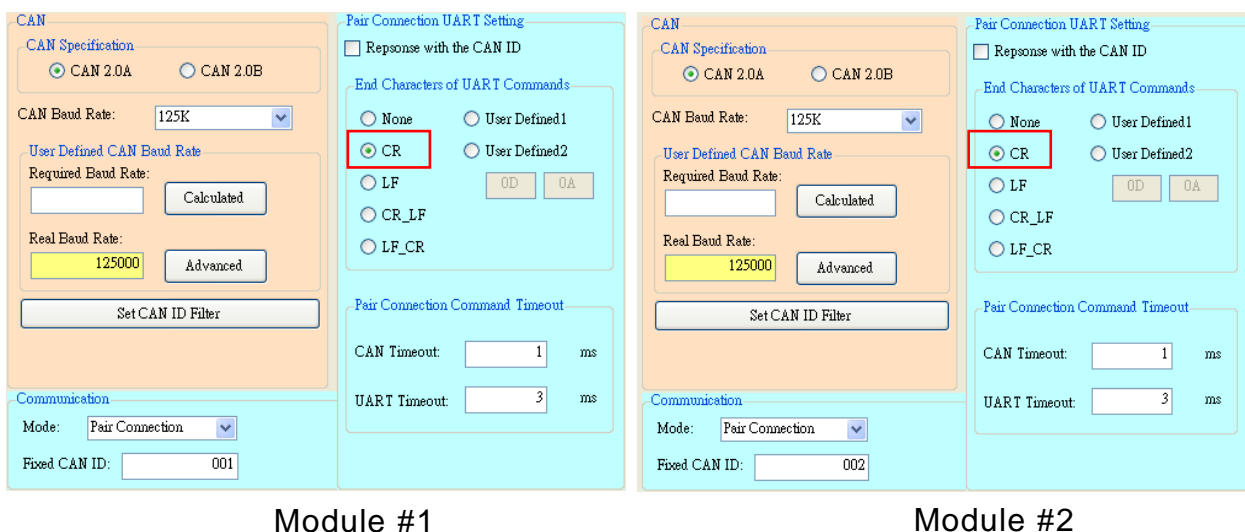
## Application 2 :

The structure of this application is same with Application 1, as shown below figure, the difference is that "CR" is selected for the End Characters of UART Commands.



### Setting:

In this application, the user needs to set up 2 sets of CAN converters, as shown in the above I-7530 module #1 and module #2, and select "CR" as the end character of the UART command. In addition, the setting of RS-232 parameters RS-232 parameters of the same 2 devices.



### Communication description:

In this application, when device 1 sends RS-232 data to module #1, module #1 will confirm whether the end character of the data is "CR", if it is "CR", it will transfer the data to the module #2, and then Module #2 will send the data to Device 2. If it is not "CR", this data will be ignored and not forwarded to Module #2.

## 5.7 COM port software

Users can use the common communication port (COM port) software to transmit and receive the data of the CAN converter on the UART side, or use VxComm Utility of ICP DAS to test. The information of VxComm Utility as below.

ICP DAS

VxComm

PRODUCTS SOLUTIONS NEWS & EVENTS SUPPORT CORPORATE CONTACT US

HOME > PRODUCTS > Software > Utility/Driver > VxComm (Virtual COM)

Introduction VxComm Driver/Utility

### VxComm (Virtual COM) Driver/Utility

The VxComm Driver/Utility can help users to create virtual COM ports in Windows and map them to the serial ports of the tDS/7188E/8000E/DS/PDS/iDS series device servers through the Internet/Ethernet network. User's RS-232/485 client programs need only to change to the virtual COM port to access the serial devices connected to the device servers through the network.

DOWNLOAD

FAQ

VxComm Utility [v2.13.14, Jun.26, 2018]

Configuration Setting

COM Port: TCP/IP Port

COM Port: COM1 Data Bits: 8

Baudrate: 115200 Parity Bit: None

Stop Bits: 1

Open COM

COM1:115200 - Terminal [v2.13.14, Jun.26, 2018]

Send \$01M

(Hex) 24 30 31 4D

Received

Interval (ms) 0

Max. display lines 2000

Clear Received

Display

Hex/Text

Hex

Text

None

Exit