GW-7433D Utility

User’s Manual

Warranty

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

Warning

ICP DAS assumes no liability for damages 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, or for any infringements of patents or other rights of third parties resulting from its use.

Copyright

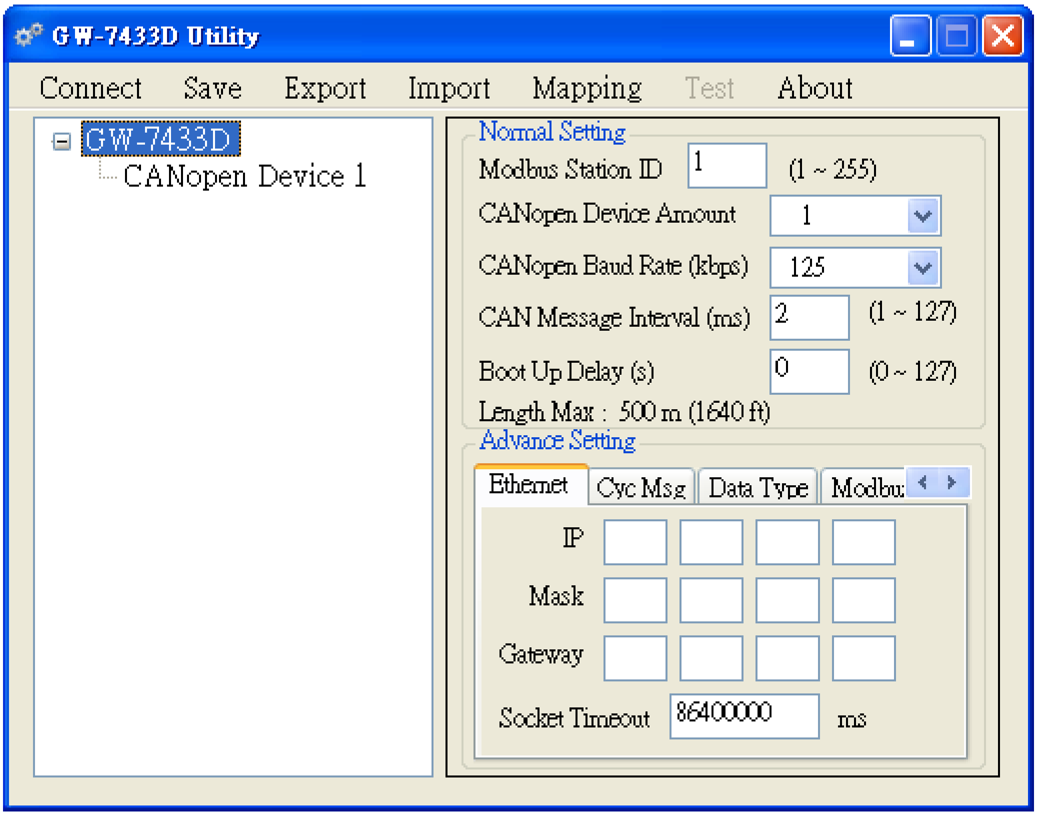
Trademark

The names used for identification only may be registered trademarks of their respective companies.

Table of Contents

1. GW-7433D Utility Introduction

GW-7433D utility is a configure tool of the GW-7433D CANopen Master to Modbus Slave gateway module. Before applying the GW-7433D to the network, users need to use the utility for configuration of the GW-7433D according to the application requirements, such as the IP address, Modbus ID, CANopen device numbers, CANopen baud, mapping of the PDO and SDO messages, and so on. The following figure is the GW-7433D utility version 2.040.



Modbus Station ID 1 (1 ~ 255)

CANopen Device Amount 1

CANopen Baud Rate (kbps) 125

CAN Message Interval (ms) 2 (1 ~ 127)

Boot Up Delay (s) 0 (0 ~ 127)

Length Max : 500 m (1640 ft)

Socket Timeout 86400000 ms

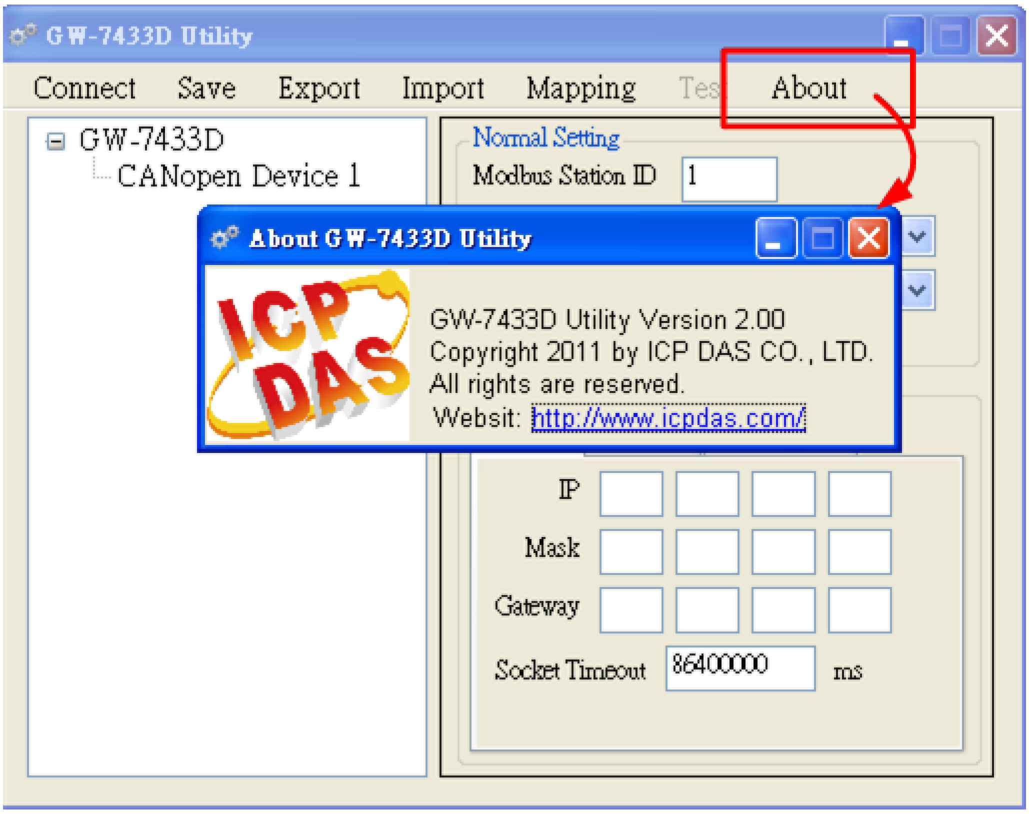
Figure 1.1 CANopen Master Utility

2. About GW-7433D Utility

The new version GW-7433D utility (v2.00 or later) is easier, more stable, and supports more functions than version 1.xx. Not like utility v1.xx, the new one just needs to execute the .exe file and doesn’t need to be installed (but the .NET Framework 2.0 is required). Because supporting more functions than v1.xx, the new one doesn’t support v1.xx firmware of the GW-7433D.

The new version GW-7433D utility (v2.00 or later) is different from the old version (v1.xx). This document is only for the new version utility. About the old one, please refer to the web: http://ftp.icpdas.com/pub/cd/fieldbus_cd/canopen/gateway/gw-7433d/oldver/.

The version information of the GW-7433D utility can be checked in the about dialog as the figure 2.1.



Modbus Station ID 1

GW-7433D Utility Version 2.00
Copyright 2011 by ICP DAS CO., LTD.
All rights are reserved.
Websit: http://www.icpdas.com/

Socket Timeout 86400000 ms

Figure 2.1 About GW-7433D Utility

3. Utility Features

1. Read back configuration of the GW-7433D.
2. Show the max CAN bus length to be the reference of configuring the CANopen baud rate.
3. Support 4 kinds of data type, 8-bit Boolean, Byte, Word, 4-byte, for the mapping of the CANopen messages.
4. Provide TxPDO, RxPDO, TxSDO, and RxSDO CANopen messages.
5. Support Guarding and Heartbeat protocol.
6. Show Modbus address mapping table at any time.
7. Provide .xml file format to export or import the configuration.
8. Provide the test mode to test the GW-7433D after finishing the configuration.
9. Reset the GW-7433D automatically after “Save” and “Close” utility.
10. Support the new version firmware of the GW-7433D (v2.00 or later).

4. Start GW-7433D Utility

Before using a new GW-7433D first time, users must execute the GW-7433D utility to configure it. Connect the user's PC with the GW-7433D by Ethernet, connect the GW-7433D and all CANopen slaves by CAN bus, and then execute the GW-7433D utility.

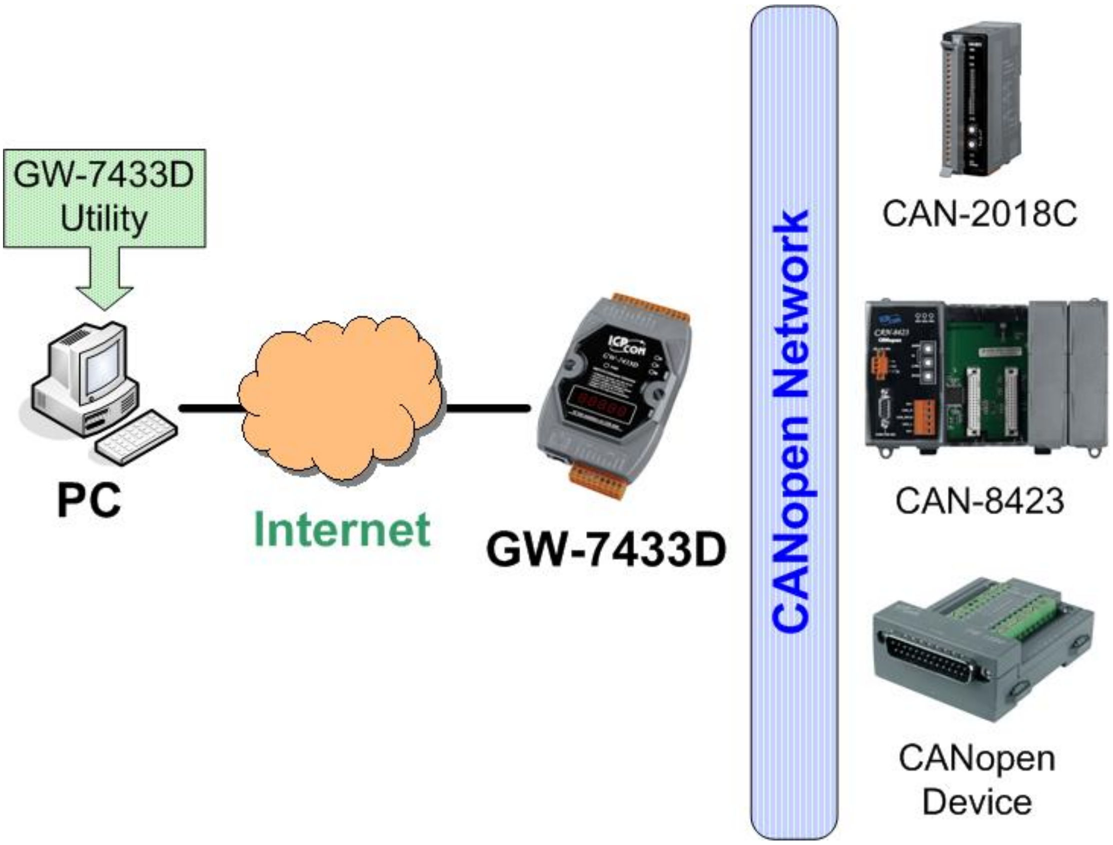
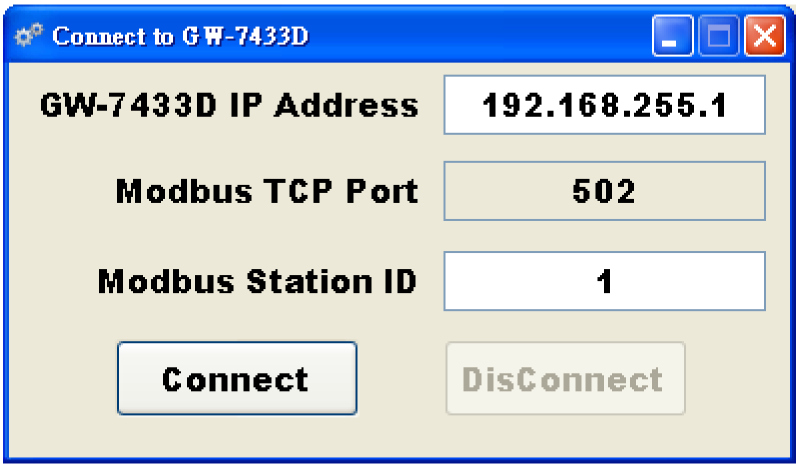


Figure 4.1 GW-7433D Connection

4.1 Connect To GW-7433D

After running the GW-7433D utility, the "Connect to GW-7433D" dialog is popped up as follows.



Connect to GW-7433D

GW-7433D IP Address: 192.168.255.1

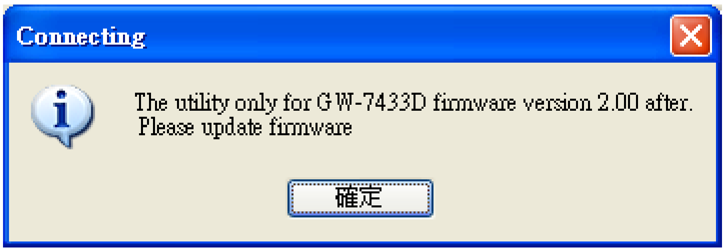
Modbus TCP Port: 502

Modbus Station ID: 1

Figure 4.2 Connect to GW-7433D dialog

The IP address and the Modbus station ID of a new GW-7433D module (GW-7433D factory default) are "192.168.255.1" and "1" and the default Mask is 255.255.0.0. Users can also see the 5-digital 7-segment LED of the GW-7433D to get the correct IP address and Modbus station ID (refer to the section 2.2.3 of the GW-7433D user manual). Input the correct IP address and Modbus station ID and then click "Connect" button to connect the GW-7433D. If the utility shows a dialog as below after clicking the "Connect" button, it means the firmware version of the GW-7433D is v1.xx, please upgrade the firmware. Users can find the firmware from the website:

ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/canopen/gateway/gw-7433d/



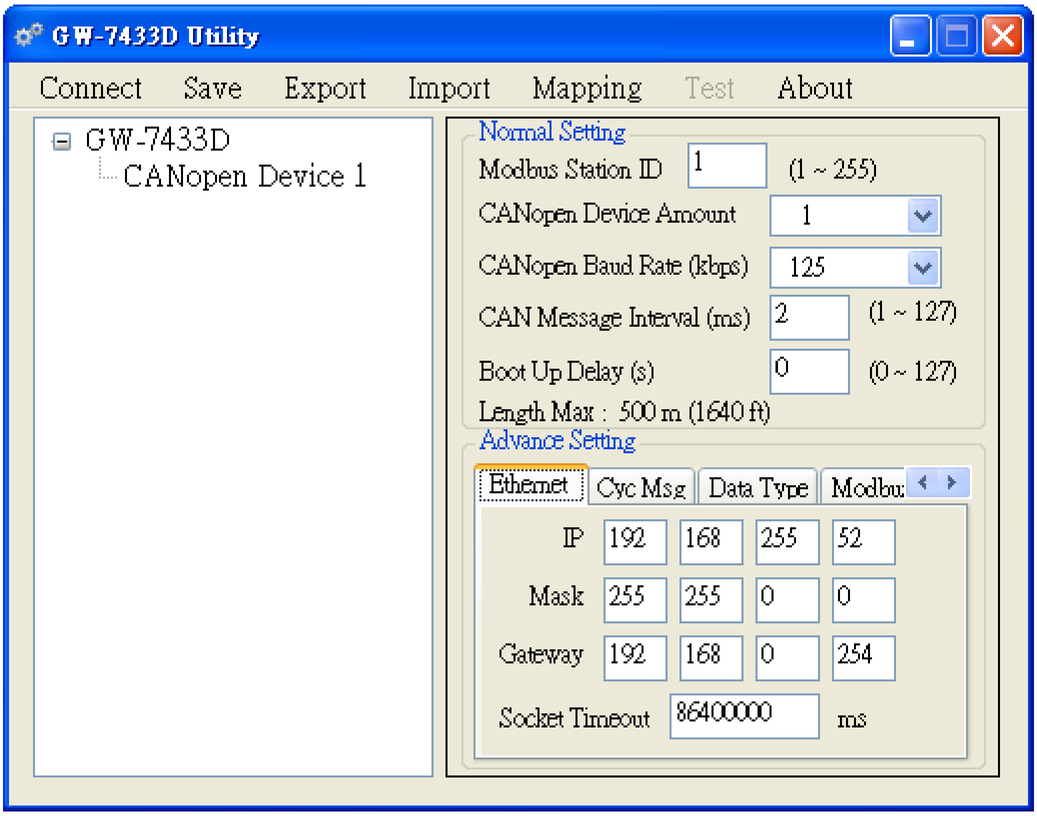
Connecting

The utility only for GW-7433D firmware version 2.00 after.
Please update firmware

Figure 4.3 Firmware version warning

4.2 Main Dialog

After connecting to the GW-7433D successfully, the utility shows the main dialog below. If the GW-7433D has been configured before, the utility will show the previous configuration on the utility.



Modbus Station ID 1 (1 ~ 255)
CANopen Device Amount 1
CANopen Baud Rate (kbps) 125
CAN Message Interval (ms) 2 (1 ~ 127)
Boot Up Delay (s) 0 (0 ~ 127)
Length Max : 500 m (1640 ft)

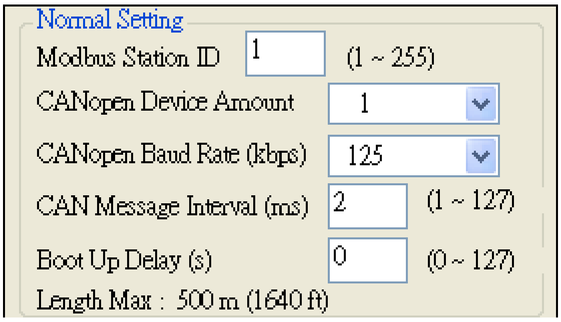
IP 192 168 255 52
Mask 255 255 0 0
Gateway 192 168 0 254
Socket Timeout 86400000 ms

Figure 4.4 Main dialog

4.2.1 GW-7433D Setting Parameter

There are two parts, “Normal Setting” and “Advance Setting”, on the “GW-7433D” parameter setting. These parameters are about the Modbus setting, Ethernet setting, data format, communication baud rate, and so on.

Normal Setting



Normal Setting

Modbus Station ID: 1 (1 ~ 255)

CANopen Device Amount: 1

CANopen Baud Rate (kbps): 125

CAN Message Interval (ms): 2 (1 ~ 127)

Boot Up Delay (s): 0 (0 ~ 127)

Length Max : 500 m (1640 ft)

Figure 4.5 GW-7433D Normal Setting

●Modbus Station ID:

It shows the GW-7433D Modbus station ID after connecting to the GW-7433D successfully. If users want to change the station ID of the GW-7433D, input a new one in this field. The range is 1 ~ 255.

●CANopen Device Amount:

Select the total CANopen slave numbers which the GW-7433D connects to. The “CANopen Device n” item showed in the left hand side will increase or decrease following this selection. The max amount is 10.

●CANopen Baud Rate:

The GW-7433D utility supports 8 kinds of baud such as, 10, 20, 50, 125, 250, 500, 800, and 1000 kbps. The “Length Max” will show the maximum CAN bus working distance depend on the baud selection.

●CAN Message Interval:

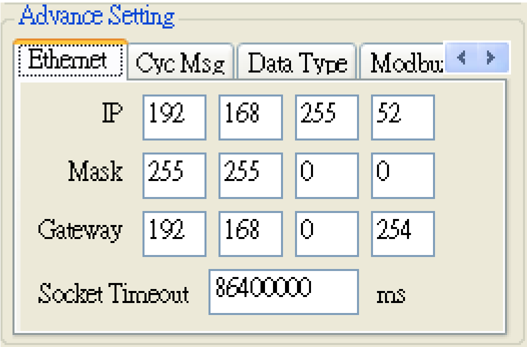
User can slow the polling speed of the GW-7433D through the parameter to avoid the CANopen network too busy.

●Boot Up Delay:

This parameter let user decide when the GW-7433D will really boot-up after power-on.

Advance Setting

Ethernet



IP 192 168 255 52

Mask 255 255 0 0

Gateway 192 168 0 254

Socket Timeout 86400000 ms

Figure 4.6 GW-7433D Advance Setting - Ethernet

- IP:

  It shows the GW-7433D IP address after connect to the GW-7433D successfully. Users can input a new IP address for configuration.

- Mask:

  It shows the GW-7433D IP mask after connecting to the GW-7433D successfully. Users can input a new mask for configuration.

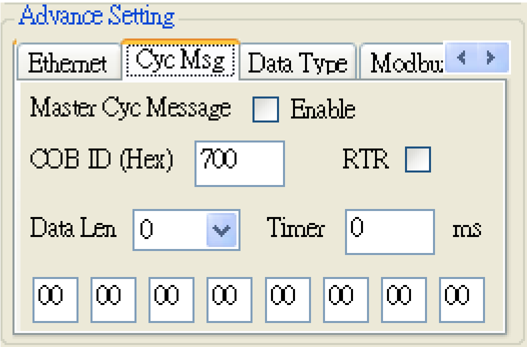
- Gateway:

  It shows the GW-7433D gateway address after connecting to the GW-7433D successfully. Users can input a new address for configuration.

- Socket Timeout:

  It is Ethernet socket timeout value. After users use a Modbus TCP client (such as PC, HMI, PAC or PLS) to connect to the GW-7433D, the GW-7433D will monitor the connection status after building a Modbus TCP connection successfully. If the GW-7433D doesn’t receive any Modbus TCP message from the Modbus TCP client during the Socket Timeout, the socket of the Modbus TCP client will be dropped. The default socket timeout value is 86400000 ms (24 hours) and socket timeout 0 means never timeout.

Cyc Msg



Master Cyc Message □ Enable

COB ID (Hex) 700 RTR □

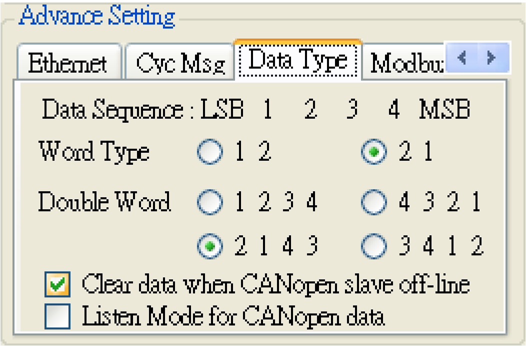
Data Len 0 Timer 0 ms

00 00 00 00 00 00 00 00

Figure 4.7 GW-7433D Advance Setting – Cyc Msg

The "Cyc Msg" function let GW-7433D can send a CAN message cyclically such as a CANopen Master Heartbeat Producer. User can decide COB ID, RTR, Data Len, Data and Timer of the cyclic CAN message and then checks the enable check box. After completed the GW-7433D setting, the GW-7433D will send the user-defined CAN message cyclically with the timer.

Data Type



Data Sequence : LSB 1 2 3 4 MSB

Word Type ○ 1 2 ● 2 1

Double Word ○ 1 2 3 4 ○ 4 3 2 1

● 2 1 4 3 ○ 3 4 1 2

☑ Clear data when CANopen slave off-line

□ Listen Mode for CANopen data

Figure 4.8 GW-7433D Advance Setting - Data Type

The data defined in the CANopen spec is little-endian, but Modbus use big-endian for the data format. To convert these two different data format, The GW-7433D utility provides some data type for transformation. Assume that the GW-7433D get the 4-byte data from the CANopen network, there are 1h, 2h, 3h, and 4h. The data type may have the following configuration.

●Word Type:

Take a look at the first two byte data, 1h and 2h. If users select “1 2”, the GW-7433D will move the received CAN data to the Modbus address directly. So, the data in the Modbus address will be filled with the data 0102h. If select “2 1”, the GW-7433D will swap the CAN data first, and save it into Modbus address. So the Modbus data will be 0201h.

●Double Word:

There are four types for double word data (4-bytes), “1 2 3 4”, “4 3 2 1”, “2 1 4 3”, and “3 4 1 2”. The situation is similar with the one of the Word Type. In the “1 2 3 4” type, the Modbus value is 01020304h. In the “4 3 2 1” type, the Modbus value is 04030201h. In the “2 1 4 3” and “3 4 1 2” types, the Modbus value is 02010403h and 03040102h respectively.

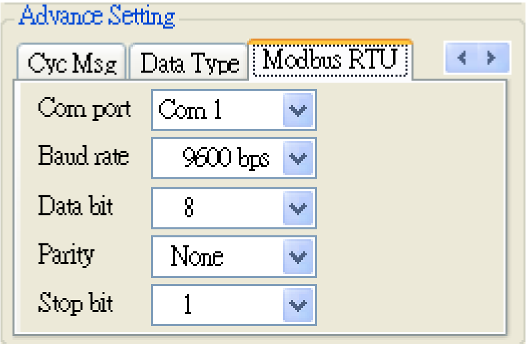
●Clear data when CANopen slave off-line:

If the function has been checked, the GW-7433D will set 0 to all the data corresponding to the special CANopen salve which loses the connection with the GW-7433D. If this item is not checked, the GW-7433D will keep the last received data while the connection of the corresponding CANopen slave is fail.

●Listen Mode for CANopen data:

Enable the function will let GW-7433D module into listen mode on CANopen side. In this case, the GW-7433D only monitors the data of CANopen network and never sends CAN messages. The GW-7433D can still monitor the Guarding and Heartbeat messages to judge if the CANopen slave is disconnect.

Modbus RTU



Com port: Com 1
Baud rate: 9600 bps
Data bit: 8
Parity: None
Stop bit: 1

Figure 4.9 GW-7433D Advance Setting - Modbus RTU

These parameters are only for the Modbus commands sent from the RS485 or RS232 port of GW-7433D. If users only use the Modbus TCP for communication, please ignore this part of instruments.

●Com port:

There are two ports to be selected, Com1 and Com2. The Com1 is RS232 port and the Com2 is RS485.

●Baud rate:

There are 8 kinds of the bauds. The selectable range is from 1200 bps to 115200 bps.

●Data bit:

The GW-7433D supports 7-bit and 8-bit data formats for Modbus RTU.

●Parity:

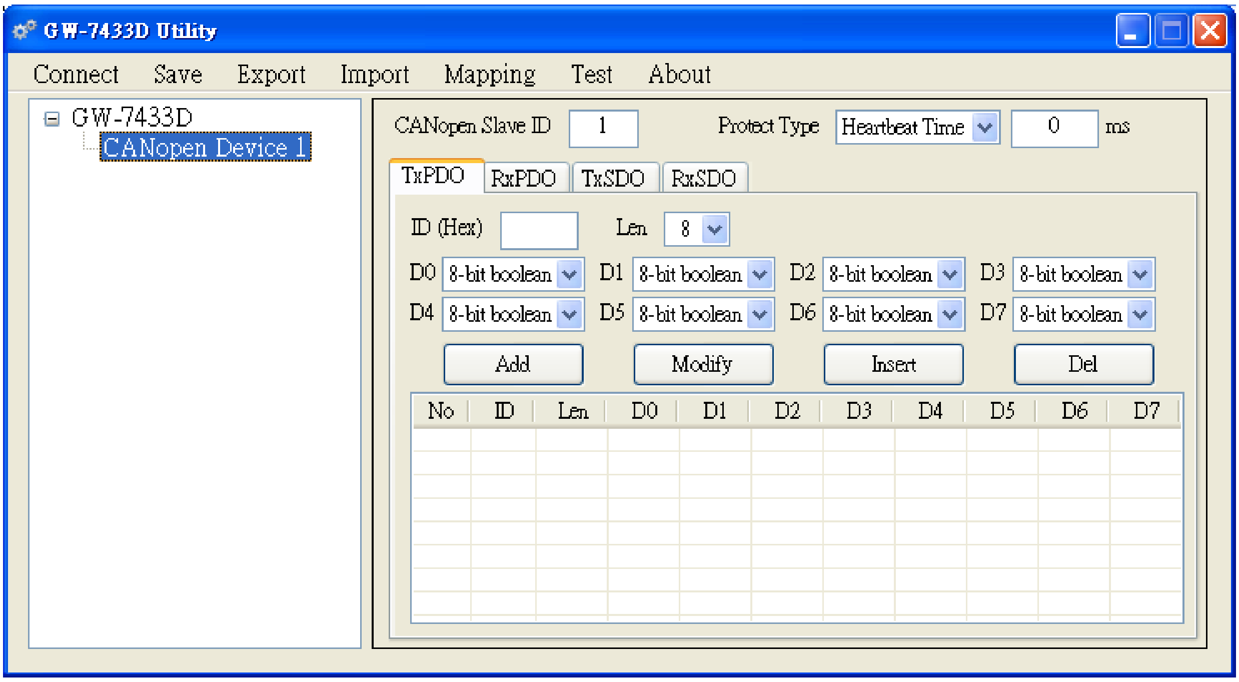
The GW-7433D supports 3 kinds of parities, None, Even, and Odd.

●Stop bit:

The GW-7433D supports 1 stop bit and 2 stop bits.

4.2.2 CANopen Device Parameter Configurations

Select a “CANopen Device” item, then the CANopen device setting dialog is shown in the right hand side. Users can set CANopen slave ID, CANopen protect function, TxPDO, RxPDO, TxSDO, RxSDO, and so on.



CANopen Slave ID: 1

Protect Type: Heartbeat Time, 0 ms

ID (Hex):

Len: 8

D0: 8-bit boolean, D1: 8-bit boolean, D2: 8-bit boolean, D3: 8-bit boolean

D4: 8-bit boolean, D5: 8-bit boolean, D6: 8-bit boolean, D7: 8-bit boolean

No	ID	Len	D0	D1	D2	D3	D4	D5	D6	D7

Figure 4.10 CANopen Device Setting

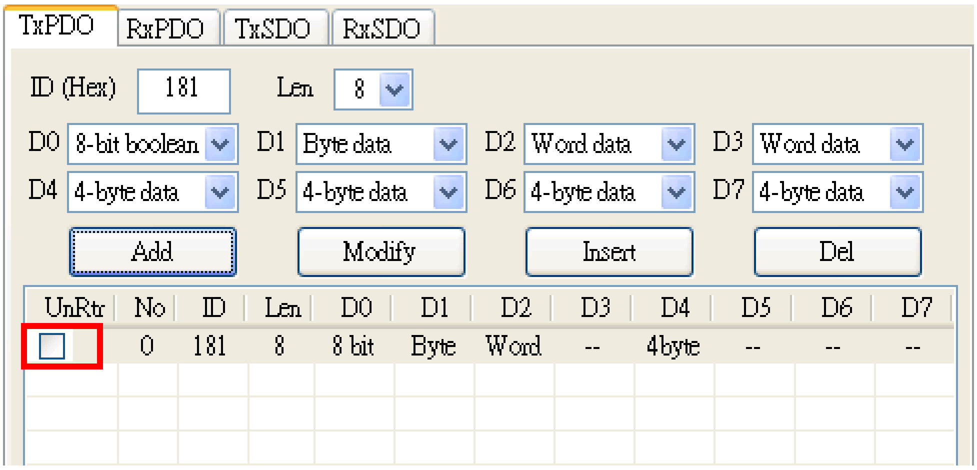
CANopen Slave ID

Use this parameter to decide which slave will be configured. The default value is from 1 ~ 10 and user can change it refer to his real CANopen slave ID. The range of ID is 1 ~ 127.

Protect Type

The GW-7433D provides two protect type, Node Guarding and Heartbeat protocol. If the CANopen slave supports one of these protocols, users can input the value for the life time of the CANopen slave. While the GW-7433D boots up, it will start to detect the connection status of the specific CANopen slave.

TxPDO



ID (Hex) 181 Len 8

D0 8-bit boolean D1 Byte data D2 Word data D3 Word data

D4 4-byte data D5 4-byte data D6 4-byte data D7 4-byte data

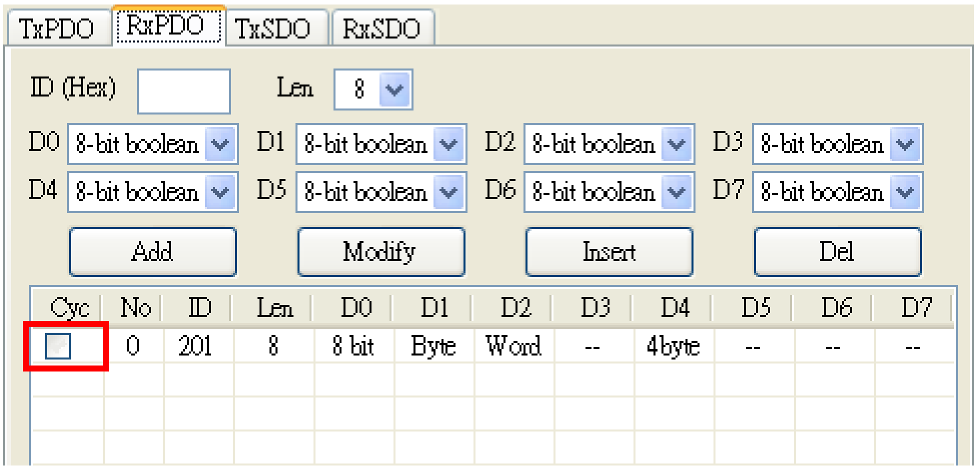
UnRtr	No	ID	Len	D0	D1	D2	D3	D4	D5	D6	D7
□	0	181	8	8 bit	Byte	Word	--	4byte	--	--	--

Figure 4.11 CANopen Device Setting - TxPDO

First, input the TxPDO Cob-ID into “ID (Hex)” text box. The Cob-ID can is used to transmit the slave data which users want to read. Second, click “Len” combo box to set the proper PDO’s data length. Users must refer to the user manual of the CANopen slave device to set the data length. Then select the data type of each byte in the TxPDO and click “Add” to store the PDO configuration into the list. Users can also select the TxPDO information in the list and click the “Modify”, “Insert”, and “Del” button to modify the content of the list. About the “UnRtr” check box, it can decide the TxPDO to be run in the normal Rtr polling mode or the un-polling Rtr (listen) mode. The default setting is normal Rtr polling mode (not check). Check it to enable the un-polling Rtr mode, the GW-7433D will just listen the TxPDO but not sent any Rtr message. About the details of the data type configuration please refer to the GW-7433D user manual section 4.2.1.

Note: If the “Listen Mode for CANopen Data” check box had been enabled (section 4.2.1\Advance Setting\Data Type), the TxPDO would all in un-polling Rtr mode whether the “UnRtr” checked or not.

RxPDO



ID (Hex)

Len 8

D0 8-bit boolean D1 8-bit boolean D2 8-bit boolean D3 8-bit boolean

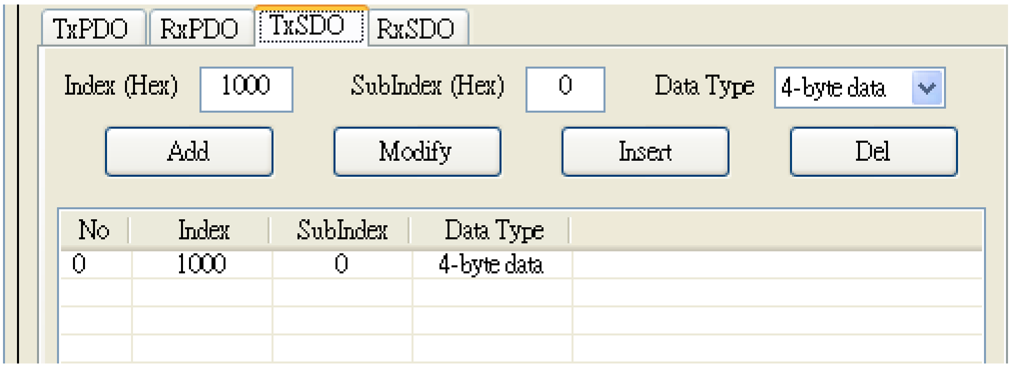
D4 8-bit boolean D5 8-bit boolean D6 8-bit boolean D7 8-bit boolean

Cyc	No	ID	Len	D0	D1	D2	D3	D4	D5	D6	D7
☐	0	201	8	8 bit	Byte	Word	--	4byte	--	--	--

Figure 4.12 CANopen Device Setting - RxPDO

The configuration of the RxPDO is almost the same as TxPDO except “Cyc” check box. The “Cyc” check box can decide the RxPDO to be run in the cyclic mode or the command mode. Check it to enable the cyclic mode, the RxPDO will be sent cyclically by the GW-7433D whether the GW-7433D has received the Modbus commands to output the RxPDO message or not. If the “Cyc” not be checked, the RxPDO will be sent only while the GW-7433D has received a Modbus command to output it.

TxSDO



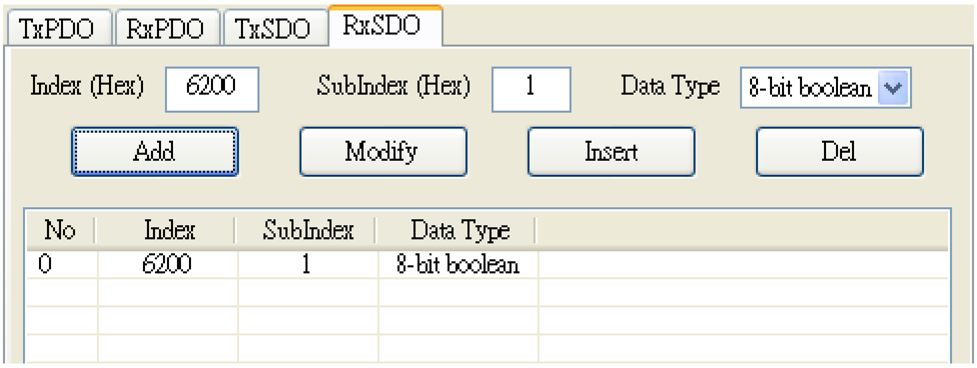
Index (Hex) 1000 SubIndex (Hex) 0 Data Type 4-byte data

No	Index	SubIndex	Data Type
0	1000	0	4-byte data

Figure 4.13 CANopen Device Setting - TxSDO

Input the “Index” and “SubIndex” of the object dictionary of the CANopen slave in these fields if users want to get the data via SDO. The GW-7433D will polling these TxSDO messages, and put the obtained data into the corresponding Modbus memory, and then users can use the Modbus command to read them.

RxSDO



Index (Hex) 6200 SubIndex (Hex) 1 Data Type 8-bit boolean

No	Index	SubIndex	Data Type
0	6200	1	8-bit boolean

Figure 4.14 CANopen Device Setting - RxSDO

Sometimes, users need to use the RxSDO to set some objects of a CANopen slave. Fill the “Index” and “SubIndex” of the object dictionary of the CANopen slave here. While the GW-7433D gets the Modbus commands writing some piece of the specific memory, it will send the corresponding data to the CAN bus via SDO. Then users can use the RxSDO to do some settings to CANopen slaves through the Modbus commands.

4.3 Save Configuration to GW-7433D

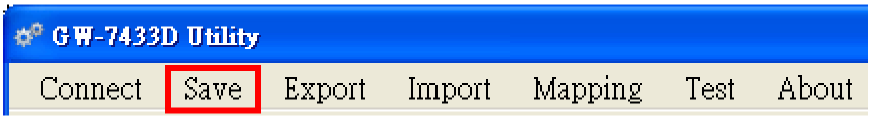
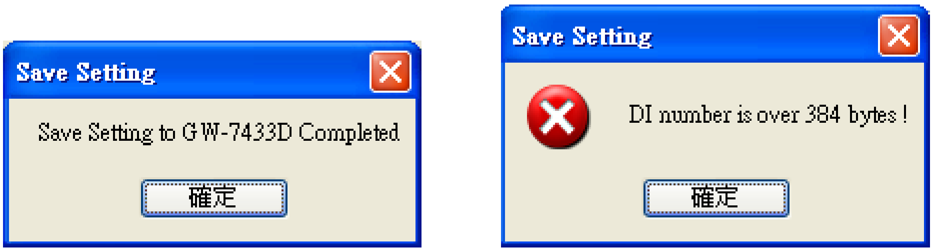


Figure 4.15 Save configuration to GW-7433D

When finishing the configurations of the all parameters of the GW-7433D, users need to click “Save” button to save these configurations. If successful, the utility will pop up a message box for hint, as following left figure. After closing the utility, the GW-7433D will reset and run with the new configurations automatically. But if the setting is over range, it will show the related error message (whether has GW-7433D or not) such as following right figure.


Save Setting

Save Setting to GW-7433D Completed

Save Setting

DI number is over 384 bytes !

Figure 4.16 Save Completed / Setting Over Range

4.4 Export and Import File

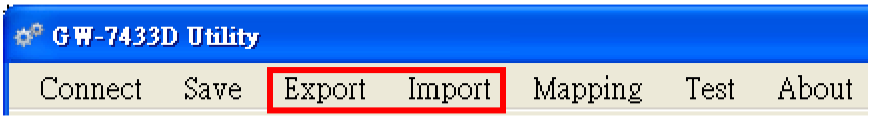


Figure 4.17 Export and Import File

If users want to configure several GW-7433D modules with the same configurations, configure the one of them first, and use the Export and Import function of the utility. After finishing the configurations of all these parameters of one GW-7433D and save the configurations, click “Export” button to export these configurations into an xml file. When users use the GW-7433D utility to connect to another GW-7433D module, click “Import” button to import the xml file for the configurations. Take a note that the “Import” function just import these configurations on the utility but not save to the GW-7433D module yet. So users need click the “Save” button to save these configurations to the GW-7433D.

4.5 Modbus Address Mapping

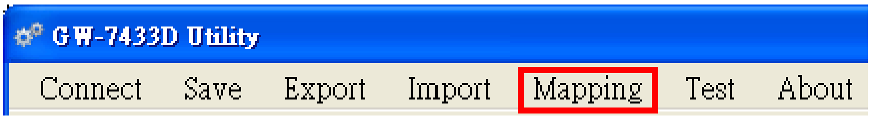
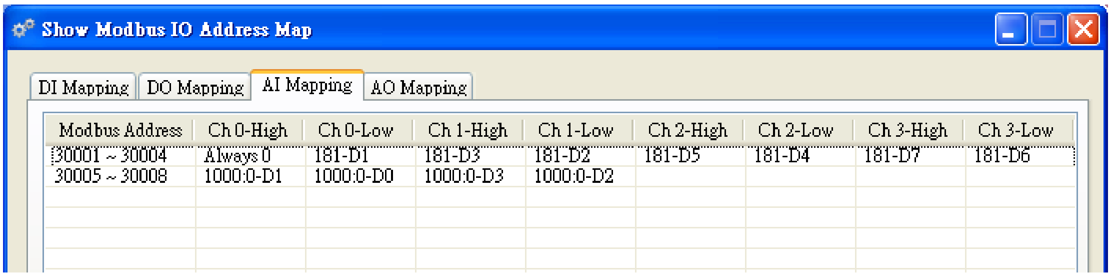


Figure 4.18 Modbus Address Mapping

While configuring the GW-7433D parameters, users may want to know what the mapping table is between the CANopen configurations of the GW-7433Ds and the Modbus address. Click "Mapping" button, users can get the Modbus address mapping at any time. Following figure shows the example of the Modbus address mapping table.



Modbus Address	Ch 0-High	Ch 0-Low	Ch 1-High	Ch 1-Low	Ch 2-High	Ch 2-Low	Ch 3-High	Ch 3-Low
30001 ~ 30004	Always 0	181-D1	181-D3	181-D2	181-D5	181-D4	181-D7	181-D6
30005 ~ 30008	1000:0-D1	1000:0-D0	1000:0-D3	1000:0-D2				

Figure 4.19 Modbus Address Mapping Table

The "Module Address" column shows the memory addresses of the data on the same row. Other columns show the source information of these Modbus data, such as PDO Cob-ID, SDO object address, data byte No., and so on. For example, we have added a TxPDO with Cob-ID 0x181 and 8 byte data to the GW-7433D (see figure 4.10). At the TxPDO D1 byte, we set it to "Byte" data type. In the mapping table, the 181-D1 is mapped to the low byte data in the address 30001. Because the D1 is a byte data, the high byte data in the address 30001 is always 0. While getting data from the Modbus interface, the Modbus TCP command is shown below.

Send: 01 02 00 00 00 06 01 04 00 00 00 01

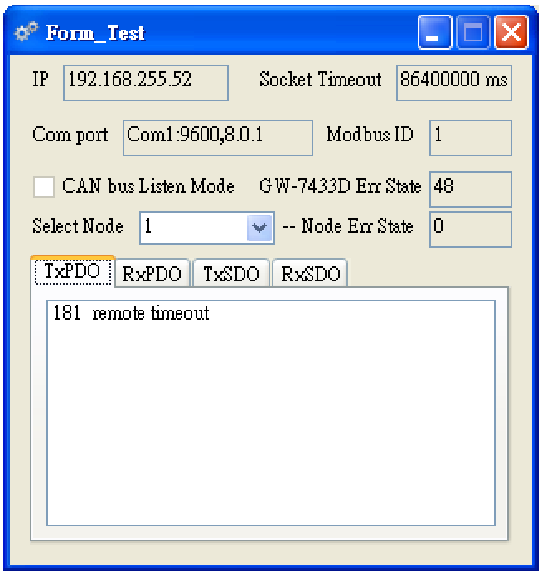
- "01 02 00 00 00 06" is the Modbus TCP header.
- "01 04 00 00 00 01" is the Modbus standard function 4 command for reading AI data.

Response: 01 02 00 00 00 05 01 04 02 00 33

- "01 02 00 00 00 05" is the Modbus TCP header.
- "01 04 02 00 33" is the Modbus standard function 4 response. The "00 33" of the reception is the high byte data 0 and the D1 data 33.

4.6 Test Function

When users finish the configuration of the GW-7433D module and start to run it, sometimes users would get some problems, such as the GW-7433D doesn’t work, can’t read the correct data, get no responses while sending Modbus commands, and so on. The “Test” function can do some simple test with Modbus TCP and CANopen bus to help users solving the problems.


Form_Test

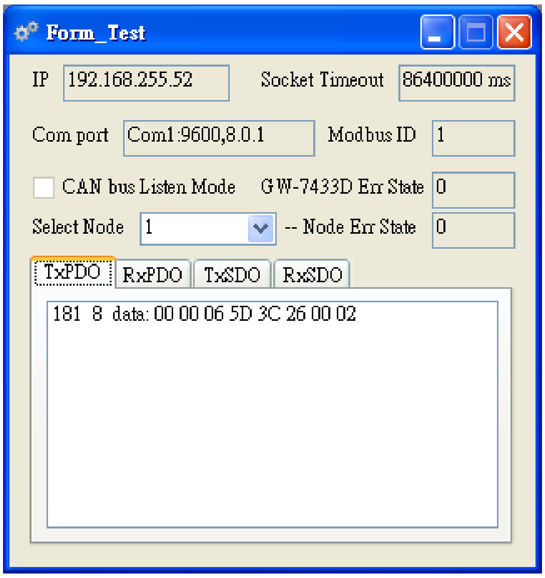
IP 192.168.255.52 Socket Timeout 86400000 ms

Com port Com1:9600,8.0.1 Modbus ID 1

□ CAN bus Listen Mode GW-7433D Err State 48

Select Node 1 -- Node Err State 0

181 remote timeout


Form_Test

IP 192.168.255.52 Socket Timeout 86400000 ms

Com port Com1:9600,8.0.1 Modbus ID 1

□ CAN bus Listen Mode GW-7433D Err State 0

Select Node 1 -- Node Err State 0

181 8 data: 00 00 06 5D 3C 26 00 02

Figure 4.20 Test GW-7433D - Error **Figure 4.21 Test GW-7433D - OK**

Figure 4.19 and 4.20 show the “Test” dialog in fail and successful. From this dialog, users can check if the parameters such as IP address, socket timeout, com port setting, Modbus ID, are correctly. Besides, users can also get the GW-7433D error status, slave node error status, and every PDO and SDO message status. In figure 4.19, the “GW-7433D Err State” is 48 that means “CAN data transmit / receive error” (please refer to GW-7433D user manual table 2-6) and the TxPDO - 181 shows “remote timeout”. So it indicates that the CANopen communication between the GW-7433D and the slave Node 1 has some problems. It may be caused by broken-line or no power. After fixing the problems, the TxPDO - 181 will receive data correctly and the GW-7433D error status is clear as figure 4.20.