GW-7433D
MODBUS TCP Server & RTU Slave to CANopen Master

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. General Information

1.1 CANopen Introduction

The CAN (Controller Area Network) is a serial communication protocol, which efficiently supports distributed real-time control with a very high level of security. It is an especially suited for networking “intelligent” devices as well as sensors and actuators within a system or sub-system. In CAN networks, there is no addressing of subscribers or stations in the conventional sense, but instead, prioritized messages are transmitted. CANopen is one kind of the network protocols based on the CAN bus and mainly used for machine control network, such as textile machinery, printing machines, injection molding machinery, or packaging machines, etc. CANopen is a low level network that provides connections between simple industrial devices (sensors, actuators) and higher-level devices (controllers), as shown in Figure 1.1.

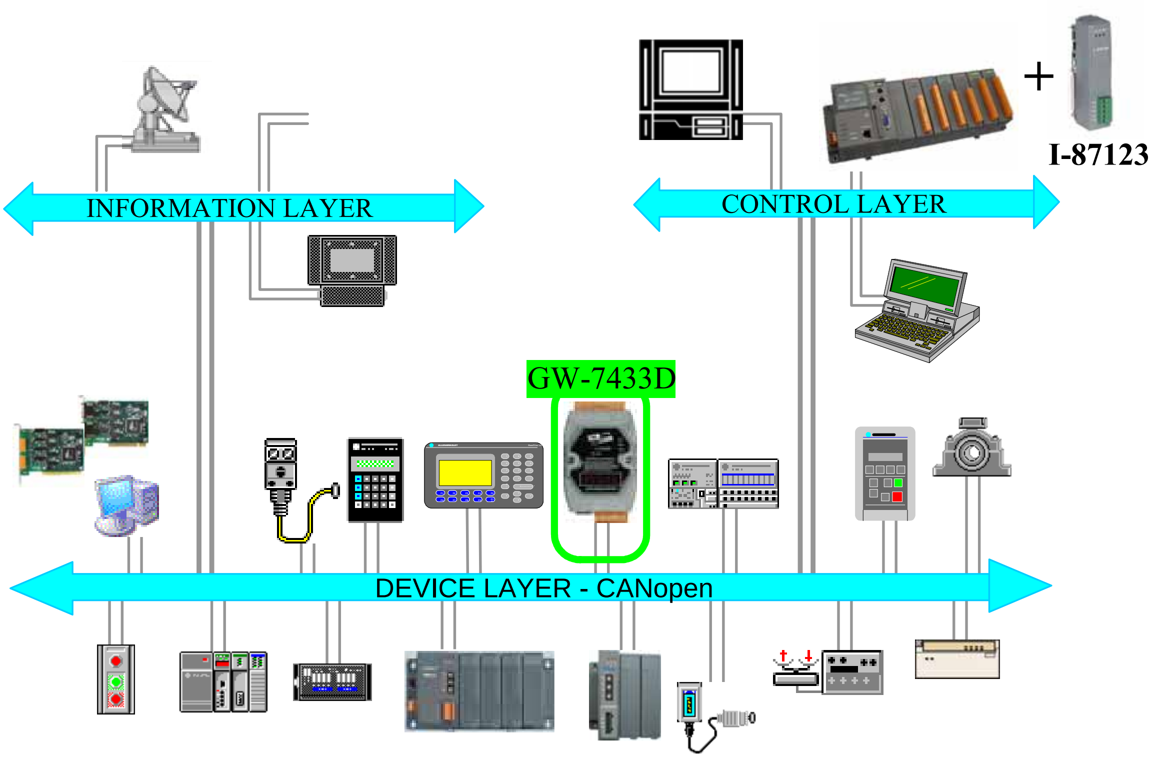


Figure 1.1 Example of the CANopen network

CANopen was developed as a standardized embedded network with highly flexible configuration capabilities. It provides standardized communication objects for real-time data (Process Data Objects, PDO), configuration data (Service Data Objects, SDO), network management data (NMT message, and Error Control), and special functions (Time Stamp, Sync message, and Emergency message). Nowadays, CANopen is used in many various application fields, such as medical equipment, off-road vehicles, maritime electronics, public transportation, building automation and so on.

Baud rate (bit/s)	Max. Bus length (m)
10 K	5000m
20 K	2500m
50 K 125 K	1000m 500m
250 K	250m
500 K	100m
800K 1M	50m 25m

Table 1.1 The Baud rate and the Bus length

1.2 CANopen Applications

CANopen is the standardized network application layer optimized for embedded networks. Its specifications cover the standardized application layer, frameworks for the various applications (e.g. general I/O, motion control system, maritime electronics and so forth) as well as device, interface, and application profiles.

The main CANopen protocol and products are generally applied in the low-volume and mid-volume embedded systems. The following examples show some parts of the CANopen application fields. (For more information, please refer to the web site, http://www.can-cia.org):

- Truck-based superstructure control systems
- Off-highway and off-road vehicles
- Passenger and cargo trains
- Maritime electronics
- Factory automation
- Industrial machine control
- Lifts and escalators
- Building automation
- Medical equipment and devices
- Non-industrial control
- Non-industrial equipment

1.3 Module Characteristics

"Embedded Internet" and "Embedded Ethernet" are hot topics today. Nowadays the Ethernet protocol becomes the de facto standard for local area network. Via Internet, connectivity is occurring everywhere, from home appliances to vending machines to testing equipment to UPS…etc. Using Ethernet for network in industrial area is appealing because the required cabling is already installed.

The GW-7433D from ICP DAS is a solution that provides a communication protocol transfer the Modbus TCP/RTU to CANopen protocol and solves a mission-critical problem, connecting an existing CANopen network to Ethernet-base PLCs and PC-based configuration and monitor system. It enables CANopen networks to be coupled together over the Internet/Ethernet, whereby remote monitoring and control is possible.

The GW-7433D can be a CANopen master device in the CAN bus on the CANopen network. It provides "Predefined Master Connection Set", On the Ethernet network, it acts as a Modbus TCP server of Modbus RTU slave. Users can use Modbus TCP/RTU class 0, class 1 and partial class 2 function to communicate with it. In addition, we also provide Utility software for users to configure their device parameters for the GW-7433D. The following figure shows the application architecture for the GW-7433D.

Figure 1.2 System application

1.4 Hardware Features

- 80186, 80MHz CPU, or compatible
- Philip SJA1000 CAN controller with 16M Hz clock
- Phillip 82C250 CAN Transceiver
- 1K VDC isolation on CAN side.
- Support both CAN specification 2.0A and 2.0B.
- Jumper select 120Ω terminator resistor for CAN channel
- 10/100 BASE-T DM9000AE compatible Ethernet Controller
- Support one RS-232 port, one RS-485 port and one CAN port
- Built-in self-tuner ASIC controller on RS-485 port
- 7-segment LED display.
- MI LED , CI LED , ERR LED

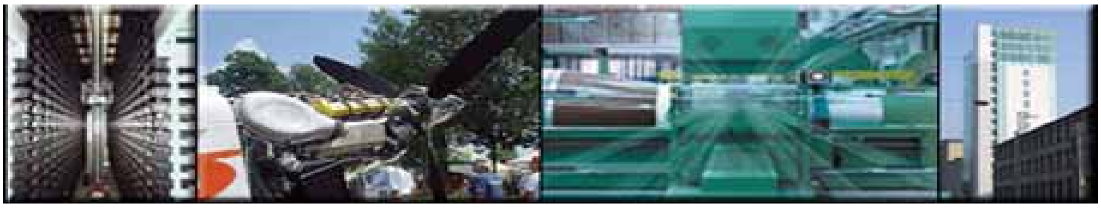
1.5 Firmware Features

- Programmable CANopen Master.
- Programmable CANopen transfer-rate 10K, 20K, 50K,125k, 250k, 500k, 800k,1M.
- Supports maximum CANopen slave devices up to 10
- Predefined Master/Slave Connection Set
- The maximum Fragment number is (Input/Output) up to 64
- Supports RxPDO counts till 50.
- Supports TxPDO counts till 50.
- Supports SDO counts till 15.
- Supports on-line adding device into and resetting device from CANopen network.
- Supports boot-up auto communicate with slave devices.
- Modbus address mapped to CANopen object by using utility tool
- Support Modbus TCP server or Modbus RTU slave functions at the same time
- Allowed multi-Modbus TCP clients access simultaneously

1.6 Specifications

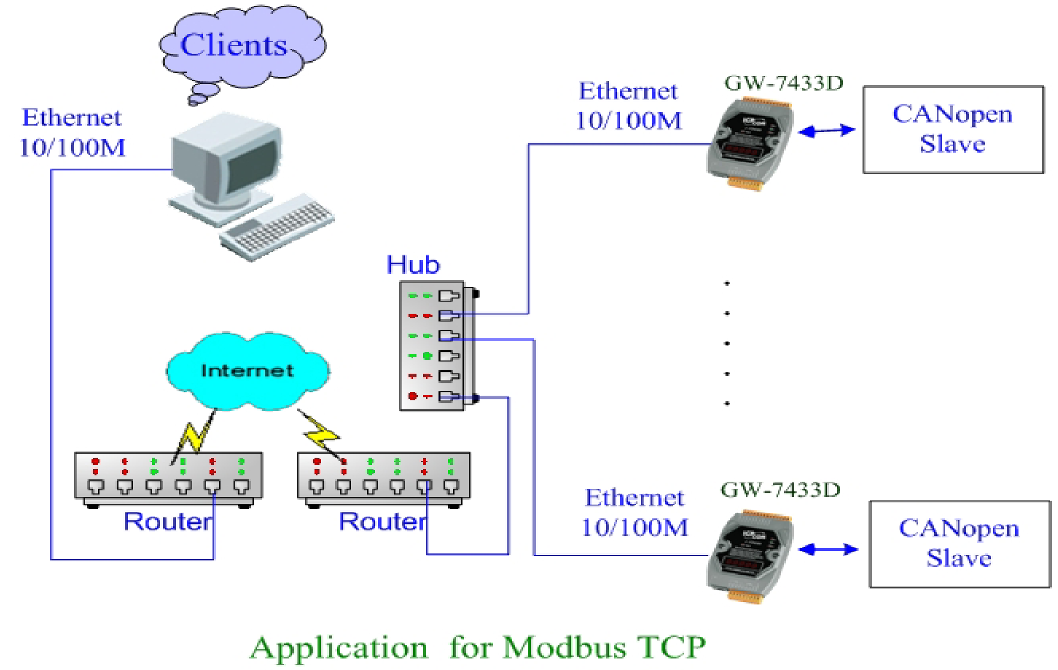
- RS-232 specification (**COM1**):
    - Communication speed: 115200 Max.
    - RS-232: TXD, RXD, RTS, CTS, GND
    - Non-isolation
    - Program download port
    -

- RS-485 specification (**COM2**):
    - Communication speed: 115200 Max.
    - D2+, D2-
    - Non-isolation
    - Self-tuner ASIC inside
- Ethernet specification:
    - 10/100 Base-T
- CAN specification:
    - CAN signal support: CAN_H, CAN_L;
    - CAN bus interface: ISO 11898-2, 5 pin screw terminal connector.
    - Isolation voltage: 1K VDC isolation on the CAN side;
- Power requirement:
    - Unregulated +10V DC ~ +30V DC.
    - Power reverse protection, Over-Voltage brown-out protection;
    - Power consumption: 3W;
- Module specification:
    - Dimensions: 123mm x 64.5mm x 19.6mm;
    - Operating temperature: -25 to 75°C;
    - Storage temperature: -40 to 80°C;
    - Humidity: 5 to 95%, non-condensing;
    - LEDs: Power led, MS Led, NS Led, Err Led, 5-digits 7 segment led displays
- Software Utility tool:
    - Online adding/removing CANopen devices via Ethernet
    - Online monitoring and configuring devices status via Ethernet.
    - Get/Set Modbus TCP/RTU input/output memory address
    - Support CANopen I/O mapping table.
    - Support communication modes setting.
- Application:
    - Factory Automation;
    - Building Automation;
    - Home Automation;
    - Control system;
    - Monitor system;
    - Vehicle Automation;



1.7 Typical Applications

The GW-7433D is designed as a Modbus TCP server or Modbus RTU Slave to CANopen master Gateway. It can access the CANopen slave device by the GW-7433D. A GW-7433D can also be able to link up to 10 CANopen slaves device; you will need to use the GW-7433D Utility to set the GW-7433D IP address and COM port, than Ethernet network hubs or RS-232/RS-485 are already in existence for system network, as figure1.3.



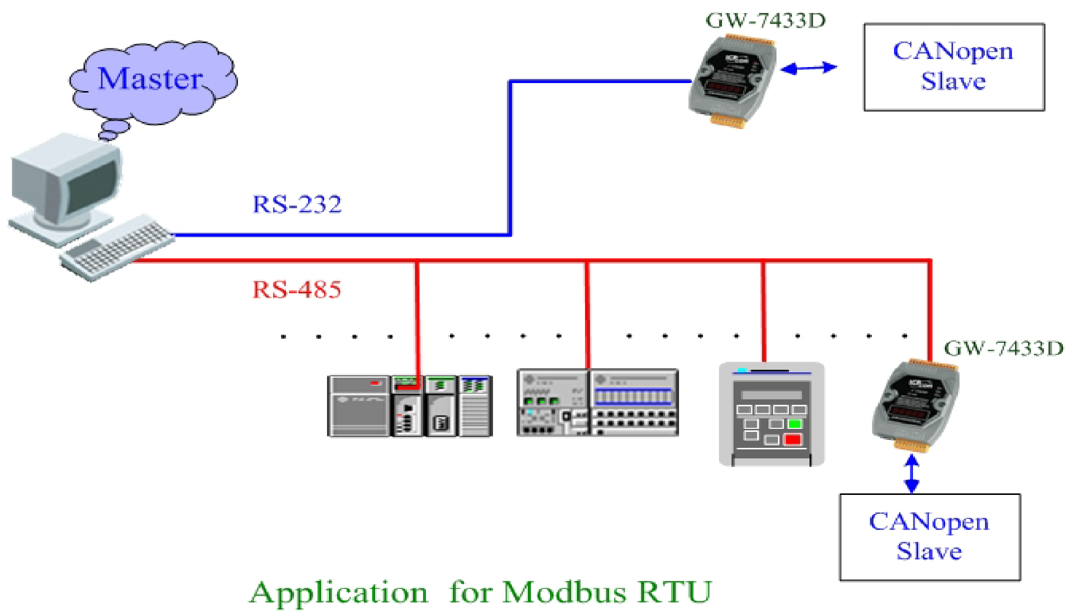


Figure 1.3: Application of GW-7433D

1.8 Modbus TCP/RTU to CANopen gateway

The GW-7433D provides centralized data storage, “Internal Memory”, for data that is shared between the CANopen and Modbus TCP/RTU network. Data is placed into the “Internal Memory” by one network interface, allowing the data to be read/wrote through the other network interface. Follow figure 1.4.

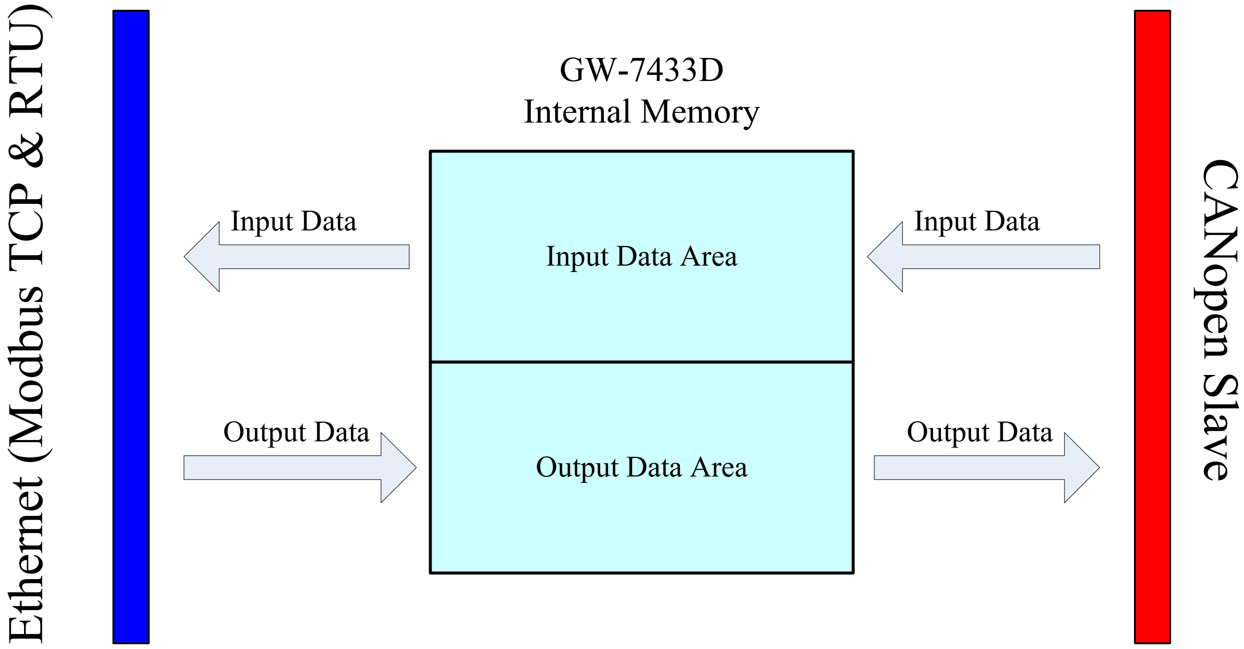


Figure 1.4: Internal Memory Table of GW-7433D

ICP DAS CANopen Master Module (GW-7433D) provides users to establish CANopen network rapidly by Master/Slave connection model. The GW-7433D Module acts as a Modbus TCP server or Modbus RTU slave to CANopen master gateway. Using the module, users don’t need to take care of the detail of the CANopen protocol. The module will implement the CANopen protocol automatically. It can reduce the complexity of user’s CANopen Master Software. The module mainly supports the Predefined Master/Slave Connection Set functions to allow users to merge I/O data into CANopen network by using Modbus TCP/RTU commands. It can help users to establish the connection with CANopen slave devices easily.

The module only provides the CANopen Master mechanism to communicate with slave devices by the Predefined Master/Slave Connection Set, The GW-7433D provide two kinds protocol of firmware, One is the PDO protocol and the others is SDO protocol. Note that before communicating I/O data with CANopen slave devices, the Master device must connect to slave devices by explicit message connection to define the connection object. Follow Figure 1.6.

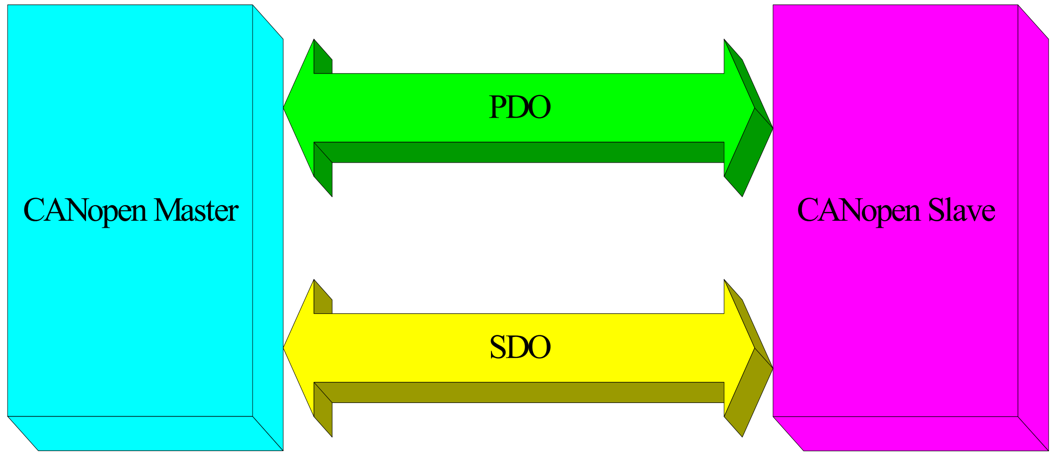


Figure1.6 CANopen Messaging

The CANopen Communication Protocol is based on the concept of connections method. Master should create connections with slave devices based on the command of exchanging information and I/O data. To establish master control mechanism, there are only four main steps to be followed. Figure 1.7 demonstrates the basic process for the CANopen master communication. The every step function is described in below:

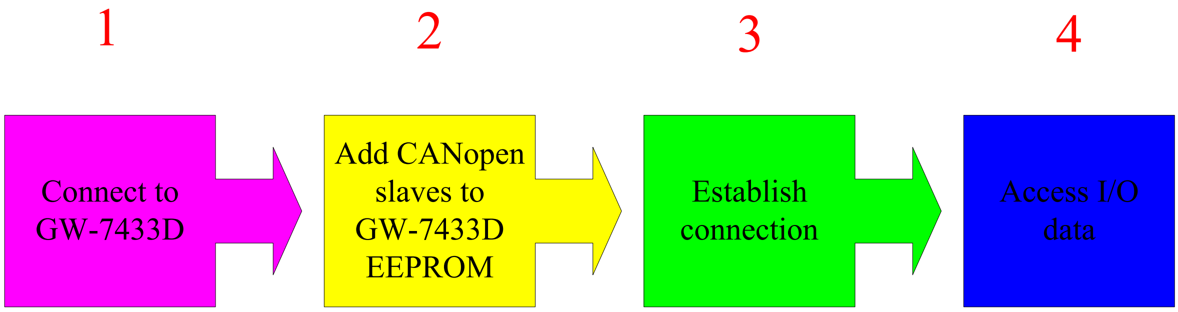


Figure 1.7 Four steps to establish connection

1. Connect to GW-7433D

You have to use the GW-7433D Utility to connect to the GW-7433D first.

2. Add CANopen slave devices to GW-7433D EEPROM

You can use the GW-7433D Utility to set the CANopen slave device's I/O configuration and parameters into the GW-7433D 's EEPROM, and you should provide these parameters.

3. Establish connection

After configuring connections, users can start communicating with CANopen slave devices.

4. Access I/O data

After communicating with slave devices, you can access the I/O data with corresponding read/write of Modbus TCP/RTU function.

After establishing the explicit connection, the connection path is for used to exchange the general information from one node to the others. And the users can create the I/O connections in the next step. Once I/O connections have been created, I/O data may be exchanged among devices in the CANopen network according to master device demand. Therefore, the master device can access I/O data of the slave devices by PDO protocol or SDO protocol methods. The module is not only easy to use but also providing a lot of the Modbus TCP/RTU commands to retrieve and deliver the CANopen slave's I/O data.

2. Hardware

2.1 Pin Assignment

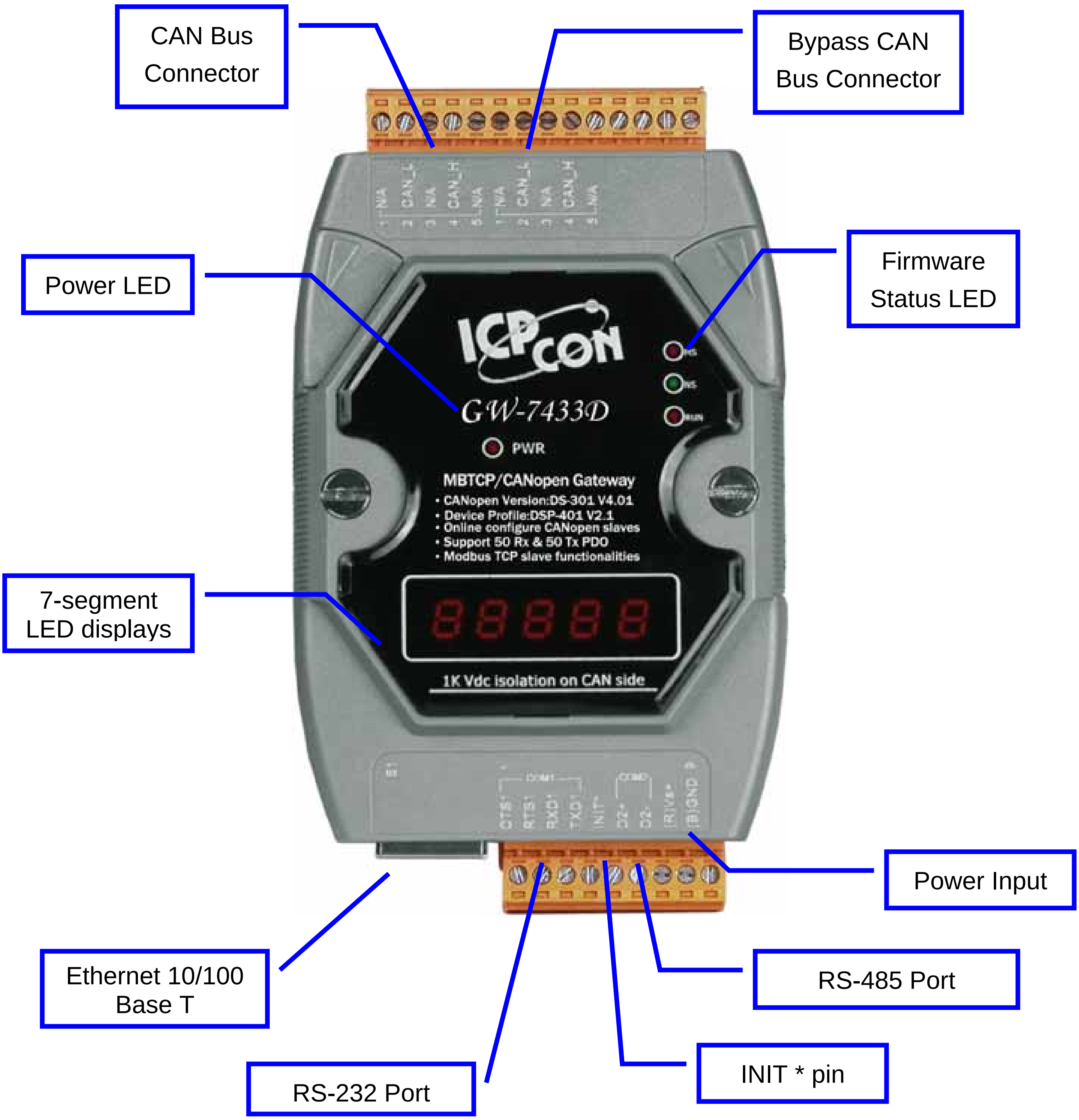


Figure 2.1 Pin assignments on the GW-7433D

2.1.1 RS-232 & RS-485 & Power supply interface

The GW-7433D provides one RS-232 interface and one RS-485 interface with hardware flow control. The GND-signal of COM1 is shared with pin-9, GND. The pin assignment is shown in table 2-1.

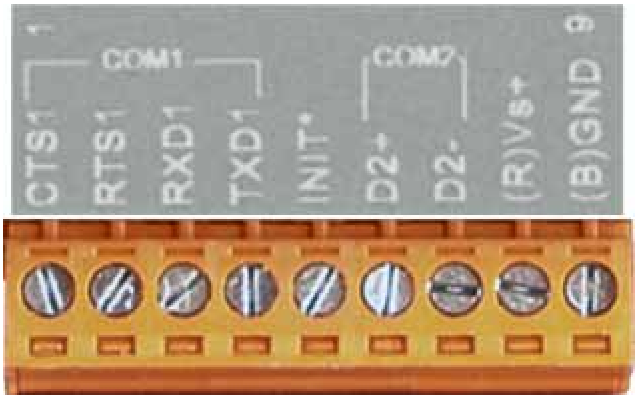


Figure 2.2

Pin	Name	Description
1	CTS1	CTS of COM1 (RS-232)
2	RTS1	RTS of COM1 (RS-232)
3	RXD1	RXD of COM1 (RS-232)
4	TXD1	TXD of COM1 (RS-232)
5	INIT*	Initial pin for enable/disable AUTOEXEC.BAT
6	D2+	Data+ of COM2(RS-485)
7	D2-	Data- of COM2(RS-485)
8	Vs+	V+ of power supply (+10V to +30V DC unregulated)
9	GND	GND of power supply

Table 2-1: COM Connector Pin Assignment

2.1.2 Connect to CANopen devices

In order to provide an easy CAN bus wiring, the GW-7433D supplies one CAN port with two CAN bus connector interfaces. Each connecter built on the GW-7433D looks like as figure 2.3 and table 2-2.

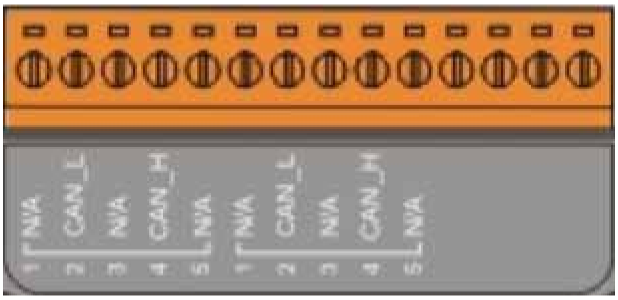


Figure 2.3

Pin No.	Signal	Description
1	N/A	Unavailable
2	CAN_L	CAN_L bus line (dominant low)
3	N/A	Unavailable
4	CAN_H	CAN_H bus line (dominant high)
5	N/A	Unavailable

Table 2-2 CAN bus Connector Pin Assignment

Note that the bypass CAN bus connector is not another CAN channel. It is designed for connecting to another CAN device conveniently. The structure of the inside electronic circuit is displayed as figure 2.4.

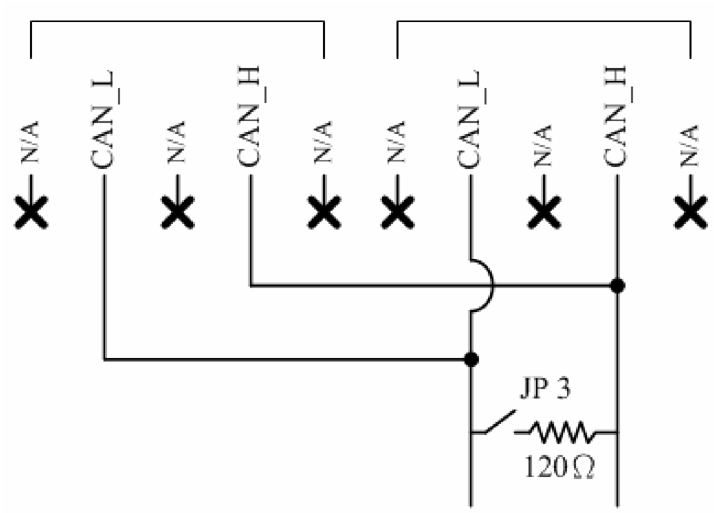
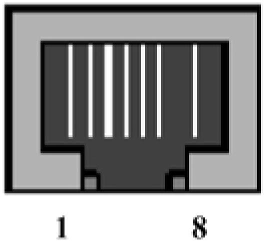


Figure2.4 Electronic circuit of CAN bus connector

2.1.3 Ethernet connection

The Ethernet (10/100 Base-T) signals are routed to an RJ45 socket for easy connection using a standard CAT 3 or CAT 5 network cable. On power on of the GW-7433D, it will auto-negotiate the network speed and connection.



Pin	Name	Description
1	TX+	Transmit Data +
2	TX-	Transmit Data -
3	RX+	Receive Data +
4	N.C.	Not Connected
5	N.C.	Not Connected
6	RX-	Receive Data -
7	N.C.	Not Connected
8	N.C.	Not Connected

Table 2-3: Ethernet Connector Pin Assignment

2.1.4 Terminator resistor settings

In order to minimize reflection effects on the CAN bus line, the CAN bus lines have to be terminated at both ends by two terminal resistances. Based on the ISO 11898-2 spec, each terminal resistance is 120Ω (or between 108Ω~132Ω). The length related resistance should have 70 mΩ/m. Users should check the resistances of their CAN bus, before they install a new CAN network as figure 2.5.

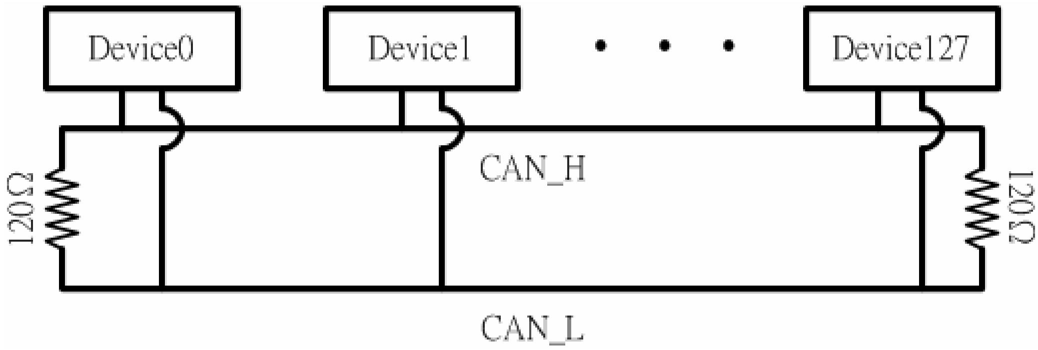


Figure 2.5 Terminator resistor

Moreover, to minimize the voltage drop on long distance, the terminal resistance should be higher than the value defined in the ISO 11898-2. Table 2-4 may be used as a reference.

Bus Length (meter)	**Bus Cable Parameters**		**Terminal Resistance (Ω)**
	Length Related Resistance (mΩ/m)	**Cross Section (Type)**	
0~40	70	0.25(23AWG)~ 0.34mm^2(22AWG)	124 (0.1%)
40~300	< 60	0.34(22AWG)~ 0.6mm^2(20AWG)	127 (0.1%)
300~600	< 40	0.5~0.6mm^2 (20AWG)	150~300
600~1K	< 20	0.75~0.8mm^2 (18AWG)	150~300

Table 2-4: Relation between bus cable and length

Therefore, the GW-7433D module supplies a jumper for users to connect the terminator resistor or not. If users want to use this terminator resistor, please open the GW-7433D cover and use the JP3 jumper to activate the 120Ω terminator resistor built in the system, as in the figure 2.6. Note that the default setting is active. And about the J3 jumper setting, please refer the table 2-5.

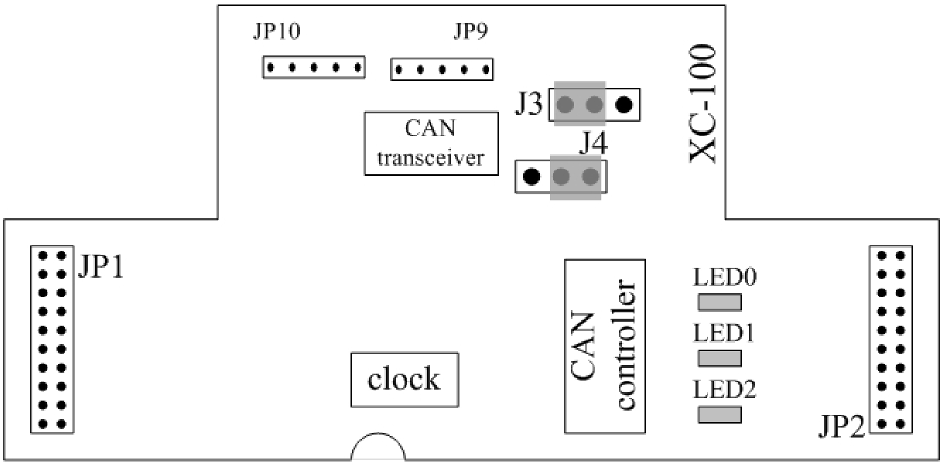


Figure 2.6 XC100 I/O expansion board LAYOUT

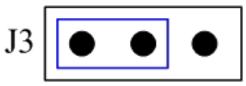
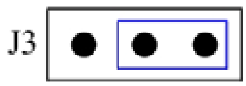
Apply the termination resistor(120Ω)	**Don’t apply the termination resistor**
	

Table 2-5 J3 Jumper Selection

2.2 LED Indication

The GW-7433D acts as a Modbus TCP/RTU to CANopen gateway. It provides some LEDs to indicate what situation is in the GW-7433D.

2.2.1 Power LED

There is a red indicator-LED in the GW-7433D as follow:

- Firmware is running: **flashing red**

The default shipping of GW-7433D will be firmware inside, so the red indicator-LED of GW-7433D will be flashing two times per second periodically.

2.2.2 Module Status indicator LED

The GW-7433D includes three single-color LED displays to indicate the status of module, network and I/O device. They are MS LED (it is red), NS LED (it is green), and RUN LED (it is red). The Indicators assist maintenance personnel in quickly identifying a problem unit. When the GW-7433D events occur, these indicators will be triggered to glitter with different conditions.

- MI LED

The led provides GW-7433D module is act or not, in the table 2-6 that is display the condition of MI status. Therefore, when the GW-7433D's firmware is normal running, the MI led must be turned on.

Condition	**Description**
Off	Module's firmware is not running.
Red	Module's firmware is normal running.

Table 2-6 MI Led conditions

- CI LED

This led indicates the CANopen communication status of the module. Table 2-7 shows the conditions of NS status. When module is online and start to communicate with the devices, it will be flash green. If there are no TxPDO, RxPDO, TxSDO, RxSDO protocol running, then the GW-7433D's NS led will be turn off.

Condition	**Description**
off	Module stops to send or receive PDO and SDO
Flashing green	Module starts to send or receive PDO and SDO

Table 2-7 CI Led conditio

- ERR LED & ERROR Message show in 7-LED

  This led indicates the CANopen status of the GW-7433D. Table 2-8 shows the conditions for RUN status.

Condition	**Description**
Off	CANopen status is normal.
On	Some errors occur on the module

Table 2-8 ERR Led conditions

2.2.3 5-digits 7-Segment LED Displays

The 5-digits 7-SEG LED will show as figure 2.7.

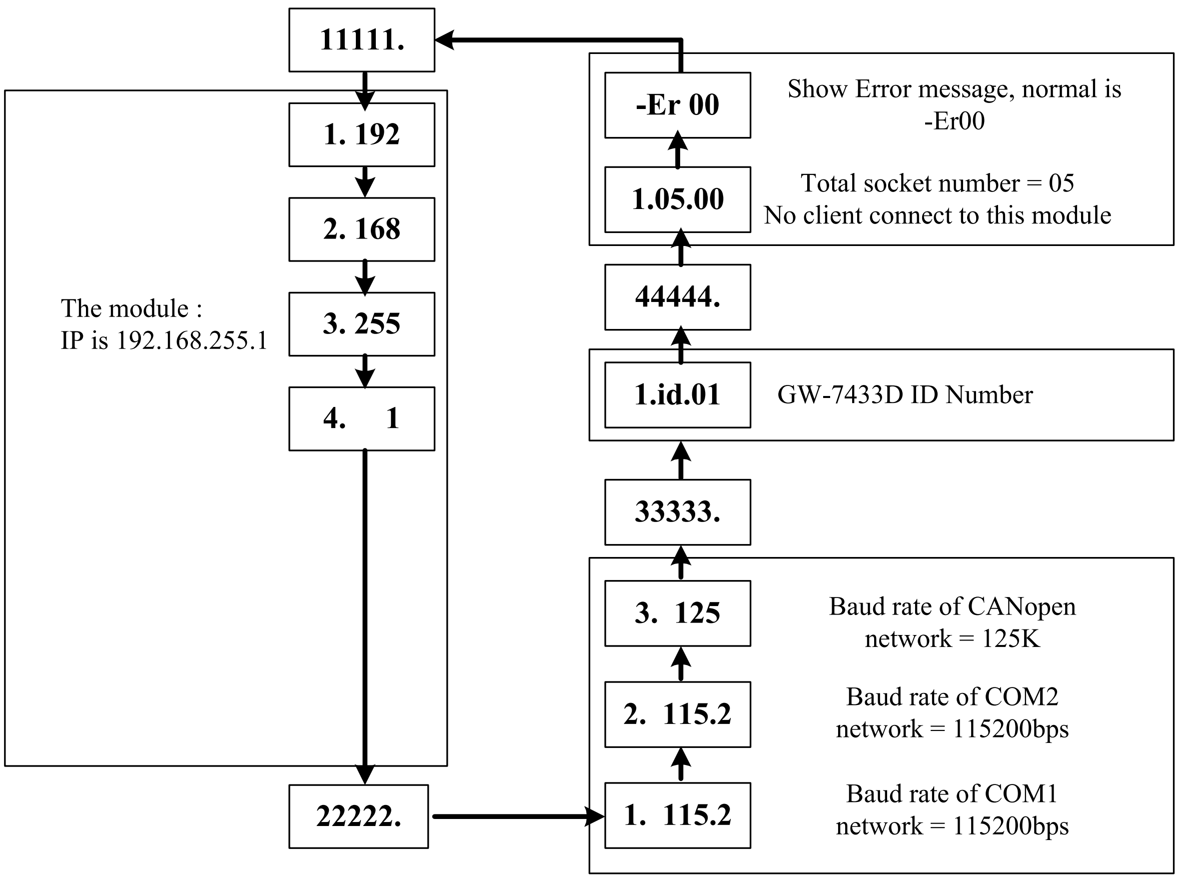


Figure 2.7 7-SEG LED Displays

The important information of GW-7433D can be divided as follows:

- Group-ID 11111: IP information of this GW-7433D
- Group-ID 22222: baud rate of all ports
- Group-ID 33333: configuration of GW-7433D station number.
- Group-ID 44444: client-connected information and system reset state of this GW-7433D

The IP information format of GW-7433D is given as follows:

- Group-ID of 5-digit LED: 11111.
- LED-1: indicator, can be 1 or 2 or 3 or 4 for 4 section of IP address
- LED-2~5: IP address

The LED will show Group-ID first, and then show its IP address as the above diagram indicates. If users change the IP address, the value shown on the LED will be changed immediately. The default IP address is 192.168.255.1.

The baud-rate format of COM1, COM2 and CAN port is given as follows:

- Group-ID of 5-digit LED: 22222.
- LED-1: port number. 1 for COM1, 2 for COM2, 3 for CAN port.
- LED-2~5: baud rate. This value needs to be timed to 1000. For COM1 and COM2, the range is from 1.2~115.2 (means 1.2 kbps ~ 115.2 kbps). For CAN port, the range is from 10~1000 (means 10 kbps ~ 1000 kbps)

The configuration of CAN port is given as follows:

- LED-1: indicators, always 1.
- LED-2/3: fix string, "id.".
- LED-4/5: GW-7433D Module ID number.

The connection-client information is given as follows:

- Group-ID of 5-digit LED: 44444.
- LED-2/3: numbers of free sockets are available, default 5.
- LED-4/5: numbers of sockets are used by clients, default 0.

The Error message of mapping parameters is given as follows:

- LED-1/2/3: fix string, "-Er".
- LED-4/5: error code, normal is 00.
- (Note: if present the Error message, you have to re-configuration GW-7433D parameters using GW-7433D Utility.)

Error code show in 7-LED	Description
-Er00	Normal
-Er51	Device Count>10 || Device Count <= 0
-Er52	127<CANopen Slave ID || CANopen Slave ID <=0
-Er53	COB_ID >0x7FF
-Er55	PDO Count >100
-Er56	SDO Count >15

Table 2-9 Error message table

If any one client connects to this GW-7433D, the free-sockets will be decreased and used-sockets will be increased. If the free-sockets number is reduced to 0, then no extra client can link to this GW-7433D. The default number of free-sockets is 5. Therefore, the GW-7433D allows 5 clients link to it.

3. CANopen Interface

3.1 Network Communication

The GW-7433D, Modbus TCP/RTU to CANopen Gateway, acts as a CANopen master on CANopen network. It can exchange I/O data with up to 10 modules (total 50 TxPDO & 50 RxPDO &15 SDO). Users can use the GW-7433D Utility tool to mapping CANopen slave module's parameters into the GW-7433D via Ethernet, and then the GW-7433D will work.

3.2 Slave Device Communication

After the configuration of CANopen slave devices, please mark the CANopen slave device to pre-operation mode, and using the GW-7433D utility to mapping the CANopen slave parameters, when you finish the GW-7433D utility configuration, the GW-7433D will auto re-boot the firmware then The module will perform all access I/O messages from CANopen slave devices.

3.3 Interaction with Internal Memory

The CANopen interface in the GW-7433D accesses the I/O data areas from slave I/O connections processed by the CANopen master; there is "no synchronization" between the Ethernet and CANopen network interfaces. As shown in the following figure.

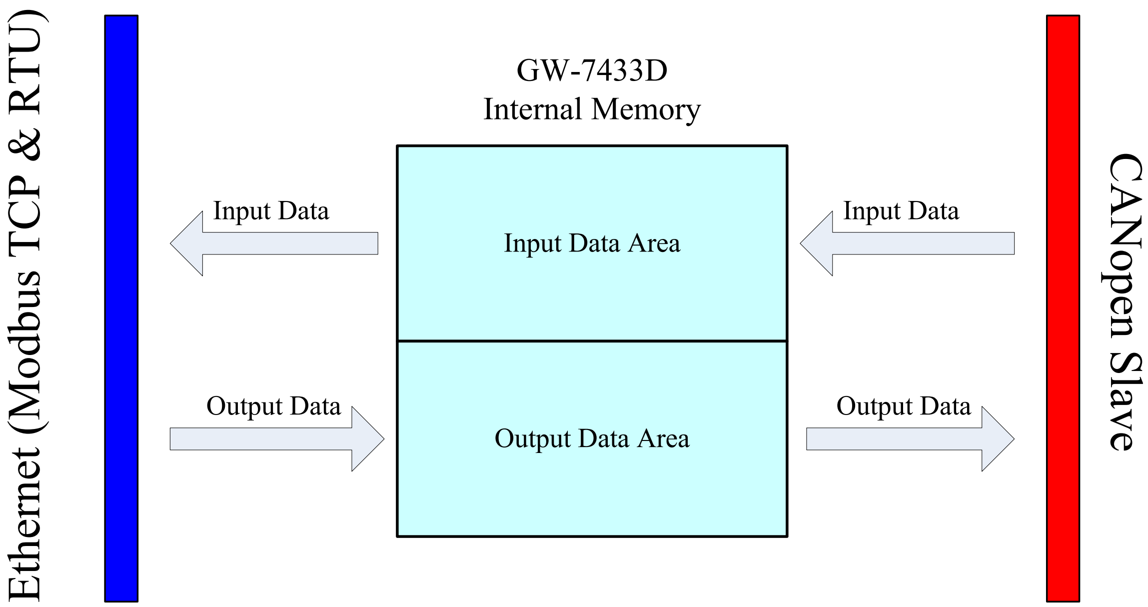


Figure 3.1 Internal Memory Table of GW-7433D

When an I/O connection with a slave requires that output data be sent to the slave, it will be read from the Output Data Area. The data read is what was placed there by the last write to the Output Data Area by using Modbus TCP/RTU commands.

When input data is received from CANopen I/O connection, it is copied to the Input Data Area. This data is available to be read by the Modbus TCP/RTU commands on the next data exchange. And these data are available to be read by the Modbus TCP/RTU commands at any time.

3.4 CAN Status Message

Table 3-1Bit interpretation of the status register (SR) for SJA1000

SR.7	BS	Bus Status; note 1	1	**bus-off**; the SJA1000 is not involved in bus activities
			0	**bus-on**; the SJA1000 is involved in bus activities
SR.6	ES	Error Status; note 2	1	**error**; at least one of the error counters has reached or exceeded the CPU warning limit
			0	**ok**; both error counters are below the warning limit
SR.5	TS	Transmit Status; note 3	1	**transmit**; the SJA1000 is transmitting a message
			0	**idle**; no transmit message is in progress
SR.4	RS	Receive Status; note 3	1	**receive**; the SJA1000 is receiving a message
			0	**idle**; no receive message is in progress
SR.3	TCS	Transmission Complete Status; note 4	1	**complete**; the last requested transmission has been successfully completed
			0	**incomplete**; the previously requested transmission is not yet completed
SR.2	TBS	Transmit Buffer Status; note 5	1	**released**; the CPU may write a message into the transmit buffer
			0	**locked**; the CPU cannot access the transmit **buffer**; a message is waiting for transmission or is already in process
SR.1	DOS	Data Overrun Status; note 6	1	**overrun**; a message was lost because there was not enough space for that message in the RXFIFO
			0	**absent**; no data overrun has occurred since the last clear data overrun command was given
SR.0	RBS	Receive Buffer Status; note 7	1	**full**; one or more messages are available in the RXFIFO
			0	**empty**; no message is available

In above, we provide a table of CAN chip status message. User can read the message by special command, as follow.

The structure of the query and response over Modbus TCP will use the command as follows:

Query (fixed)➔ 1 2 0 0 0 6 1 10 0 0 0 2 4 23 70 44 00

Response➔ 1 2 0 0 0 6 19 75 04 01 0C 00

Analysis Response :

Byte 0	0x 1	Fixed
Byte 1	0x 2	Fixed
Byte 2	0x 0	Fixed
Byte 3	0x 0	Fixed
Byte 4	0x 0	Fixed
Byte 5	0x 6	Fixed
Byte 6	0x 19	Fixed
Byte 7	0x 75	Fixed
Byte 8	0x 04	Fixed
Byte 9	0x01	Fixed
Byte 10	0x0C	CAN status Message.
Byte 11	0x00	Error code

Table 3-2 CAN chip status message

4. Modbus TCP/RTU Interface

GW-7433D supports Modbus TCP/RTU commands. The implementation of the Modbus TCP client or Modbus RTU master is done according to the Modbus TCP/RTU specification 1.0. All commands according to class 0, class 1 and partially class 2 slave functionality that are implemented.
The module can handle maximum 5 simultaneous Modbus TCP connections, and one RS-232(COM1) or RS-485(COM2) connection.

4.1 Commands

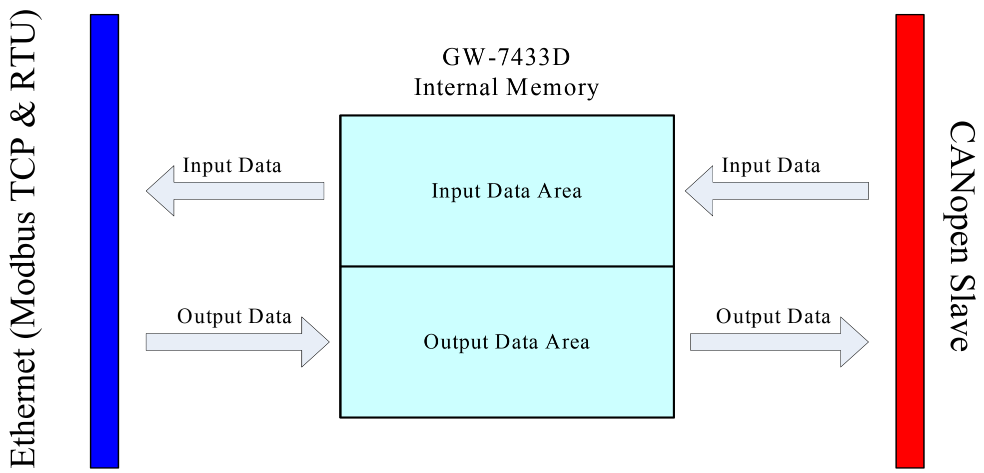
The following Modbus TCP/RTU commands are supported by the GW-7433D.

Function Code	Function Name	Class	Affects	Address Method
1	Read Coils	1	IN/OUT	Bit
2	Read Input Discrete	1	IN/OUT	Bit
3	Read Multiple Registers	0	IN/OUT	Word
4	Read Input Registers	1	IN/OUT	Word
5	Write Coil	1	OUT	Bit
6	Write Single Register	1	OUT	Word
15	Force Multiple Coils	2	OUT	Bit
16	Force Multiple Registers	0	OUT	Word

Table 4-1: Modbus Commands

4.2 Modbus TCP/RTU Addressing

The GW-7433D supports an “Internal Memory” for saving I/O data. The input and output data areas are set to a maximum size of 256 words each. The command area is 255 words. The input status area is 2500 words and the output status area is 2500 words. When accessing these areas, by Modbus TCP/RTU commands, the address is according to the following table4-2 and table4-3.



4.2.1 Input/Output Data Areas

The I/O Data Area are used for users to access devices I/O data directly. After using Utility tool to set the memory mapping of the I/O devices, users can get/set the I/O data according to the mapping address of each I/O data.

Users can use Modbus TCP/RTU function 4 command, read input registers, to get the input data values from GW-7433D's input data area.

Word Address (3xxxx)
0000h
0001h
0002h
:
00FFh

Table 4-2: Input Addressing

Users can use Modbus TCP/RTU function 16 command, force multiple registers, to set the output data values into GW-7433D's output data area.

Word Address (4xxxx)
0000h
0001h
0002h
:
00FFh

Table 4-3: Output Addressing

I/O Data Format

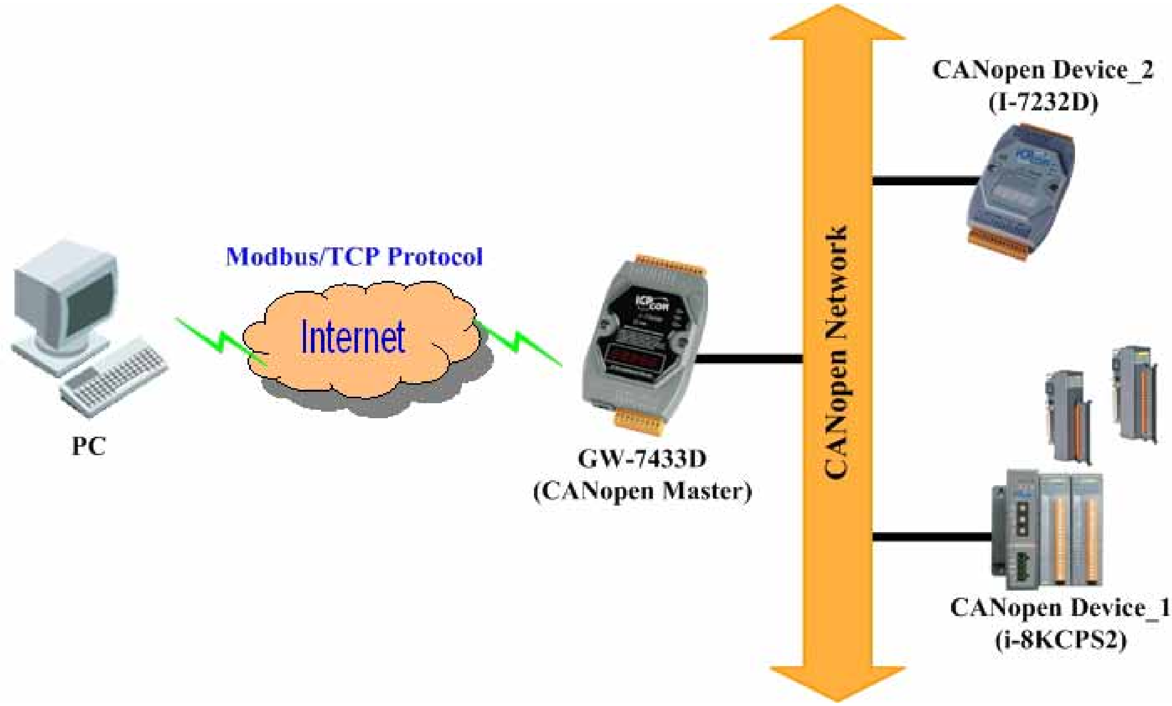
GW-7433D transfers I/O data between Modbus TCP/RTU and CANopen without regard to data content or format. Due to this, the user is responsible for making sure that the devices on either network understand the format of the data.

CANopen is a little-endian protocol; values ate transmitted least significant byte first. Therefore, all data in the I/O Data Areas is assumed, by CANopen nodes, to be stored as little-endian.

The user must know the mapping of I/O Data Areas between the Modbus TCP/RTU and CANopen.

5. Software

5.1 GW-7433D Configuration Tool (GW-7433D Utility).



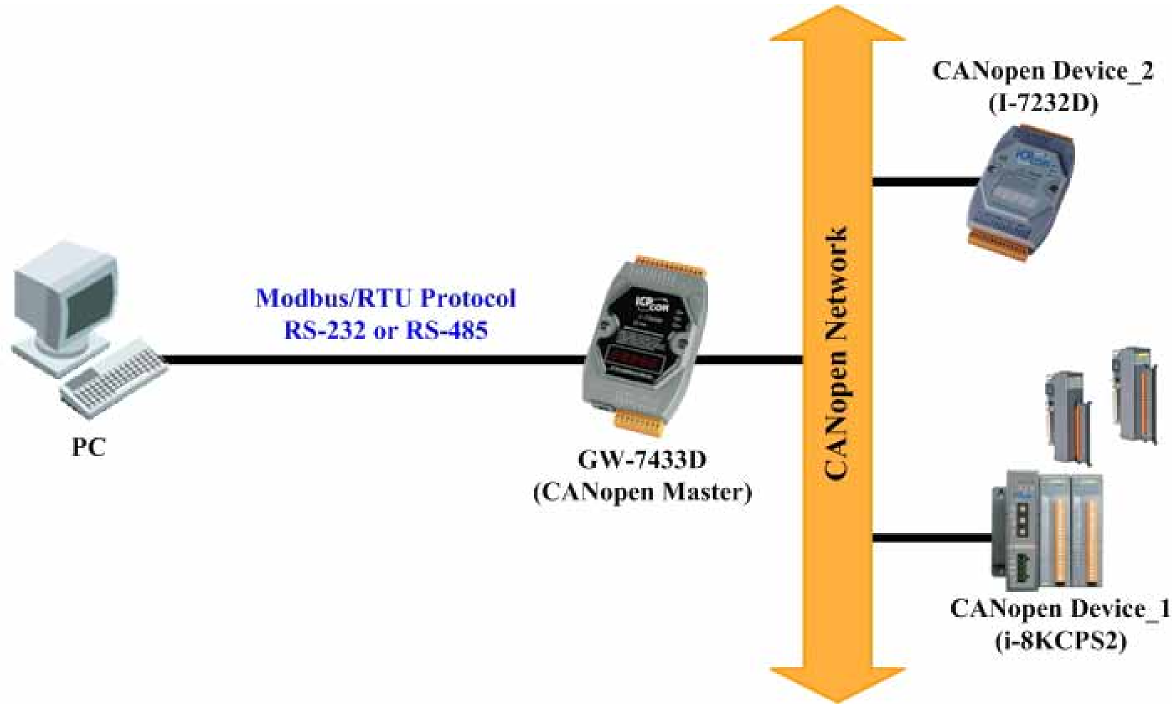


Figure 5-1: GW-7433D Application Construction

The GW-7433D Utility tool can be used to configure the parameters of each CANopen slave device hanged on the CAN port of GW-7433D via Ethernet. Before you use utility tool, connect the GW-7433D's Ethernet port confirm your PC can connect with the GW-7433D via Ethernet. Users may use "ping" command to check if the Ethernet connection work or not. The default IP address of GW-7433D is 192.168.255.1. If users are first time to use the GW-7433D, the IP address of GW-7433D may be needed to modify. The following steps show the installation procedure and configuration of IP address. They are helpful for users to establish the connection with the GW-7433D.

5.2 Install & uninstall the GW-7433D Utility

Install GW-7433D Utility

Step1: Download the GW-7433D Utility setup file from the web site http://www.icpdas.com/products/Remote_IO/can_bus/gw-7433d.htm or the CD-ROM disk following the path of "CAN-CD:\\CANopen\Gateway \GW-7433D\Utility

Step 2: Execute the setup.exe file to installGW-7433D Utility.

Step 3: A "Welcome" window pops up to prompt user to begin the installation. See figure 5-2.

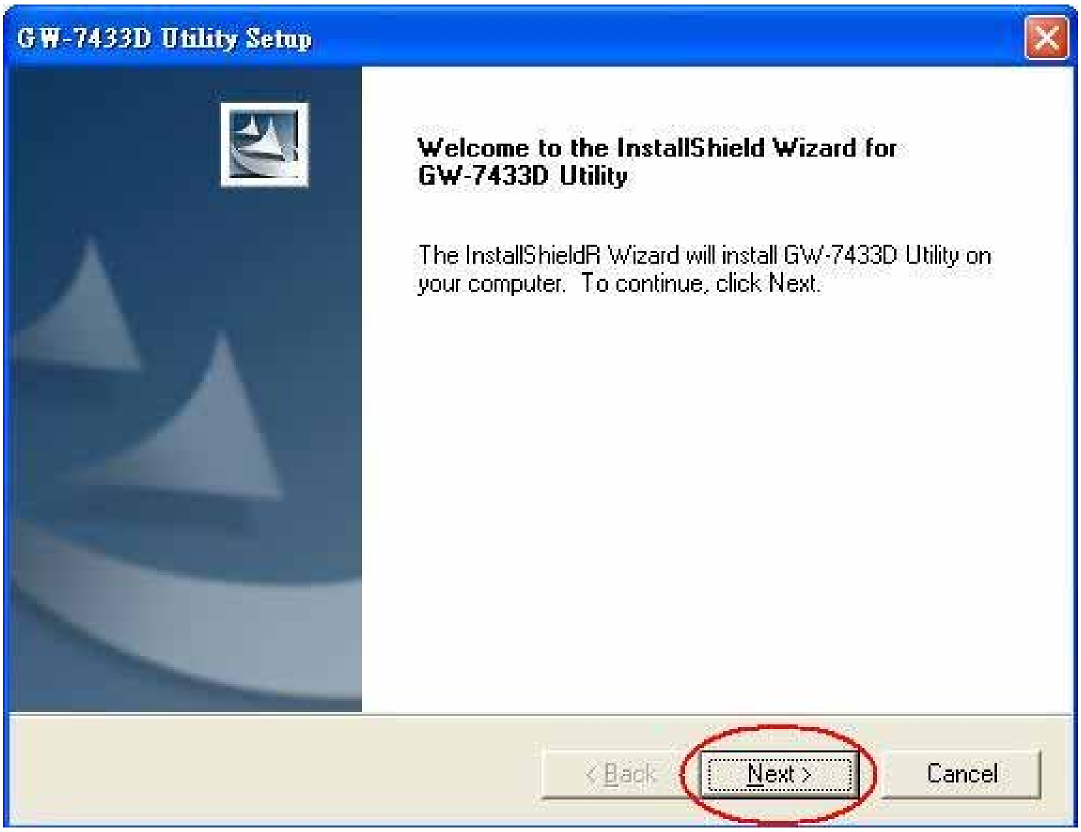


Figure 5-2: Welcome dialog

Step 4: Click the “Next” button and A “Choose Destination Location” window will pop up for deciding the installation path.

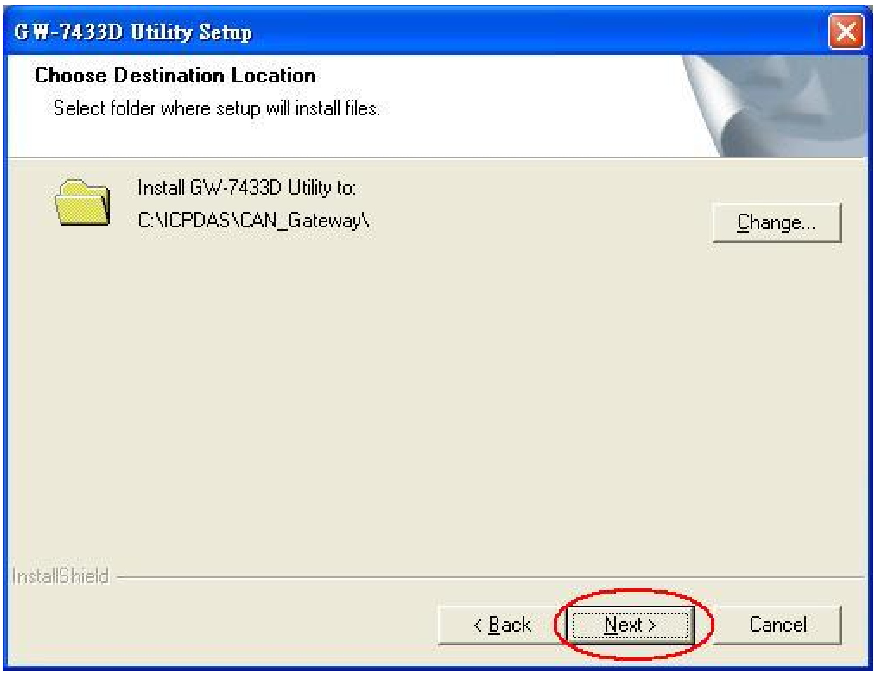


Figure 5-3: “Choose Destination Location” dialog

Step 5: Click “Next” button and a “Ready to Install the Program” window will pop up to prompt user that the wizard is ready to begin the installation See figure 5-4.

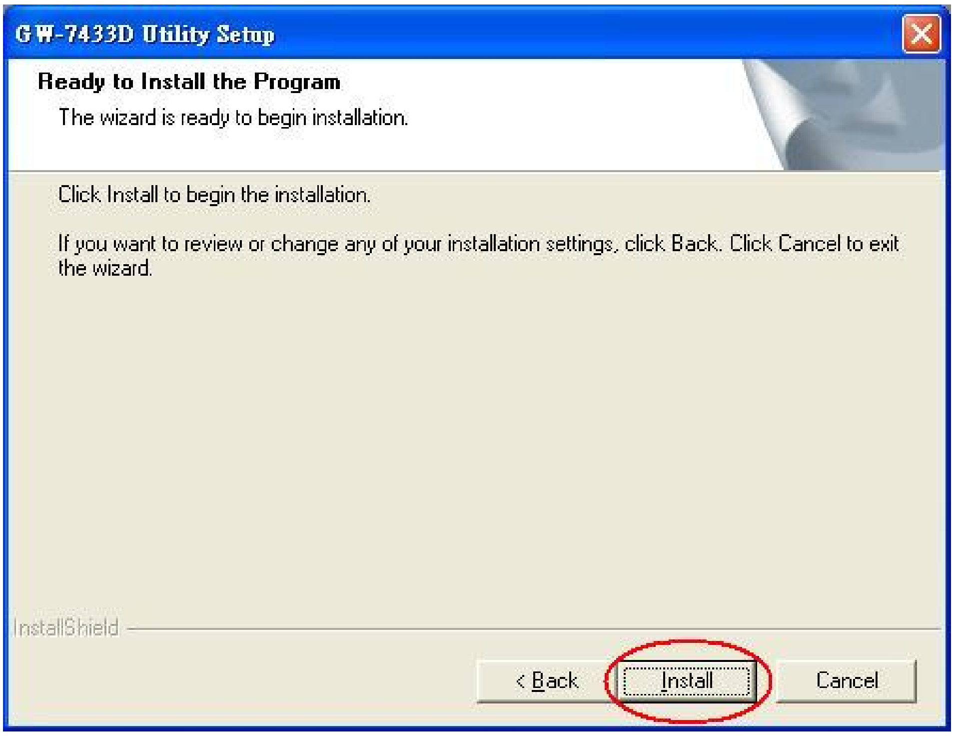


Figure 5-4: “Ready to Install the Program” dialog

Step 6: Click "Install" button and start to install the GW-7433D Utility to the system. After finishing the process, a "Complete" window will pop up to prompt users that the successful completion of the installation. And click "Finish" button to exit. See figure 5-5.

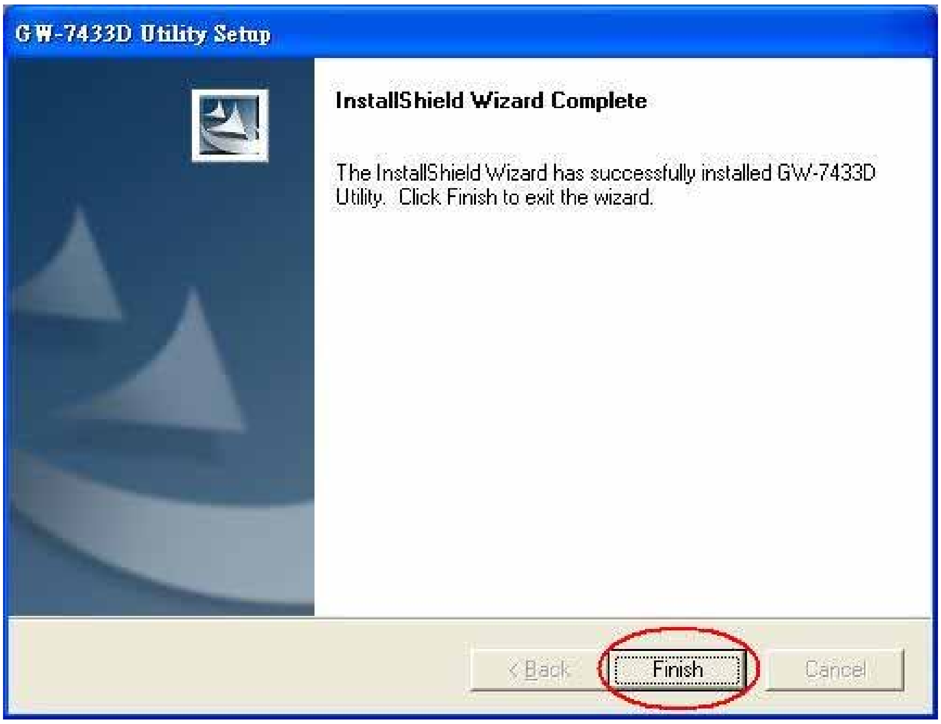


Figure 5-5: "Successful Completion of the Installation" dialog

Step 7: After finishing the installation of the I-7243D Utility, users can find it as shown in figure 5-6.

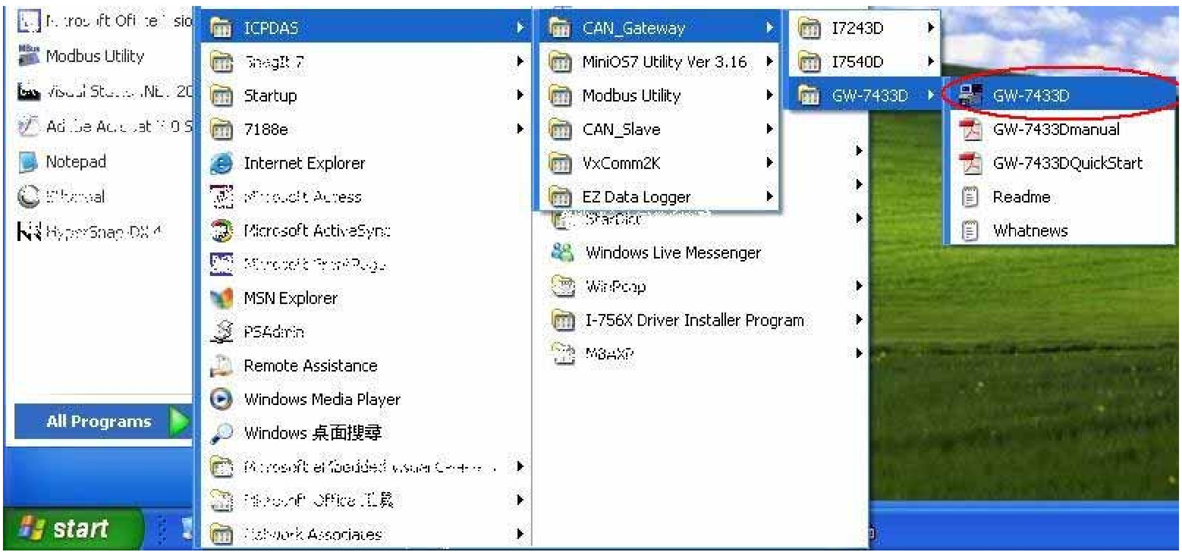


Figure 5-6: The "GW-7433D Utility" path

Uninstall GW-7433D Utility

You can uninstall I-7243D Utility software by the following means described below:

Step 1: Click "Start" in the task bar, then clicks the Control Panel as shown in figure 5-7.

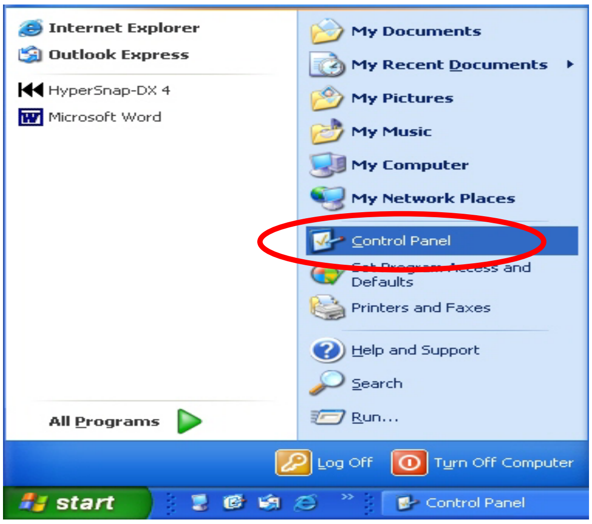


Figure 5-7: Select settings

Step 2: Double click the "Add or Remove Programs" button icon to open the dialog. See figure 5-8.

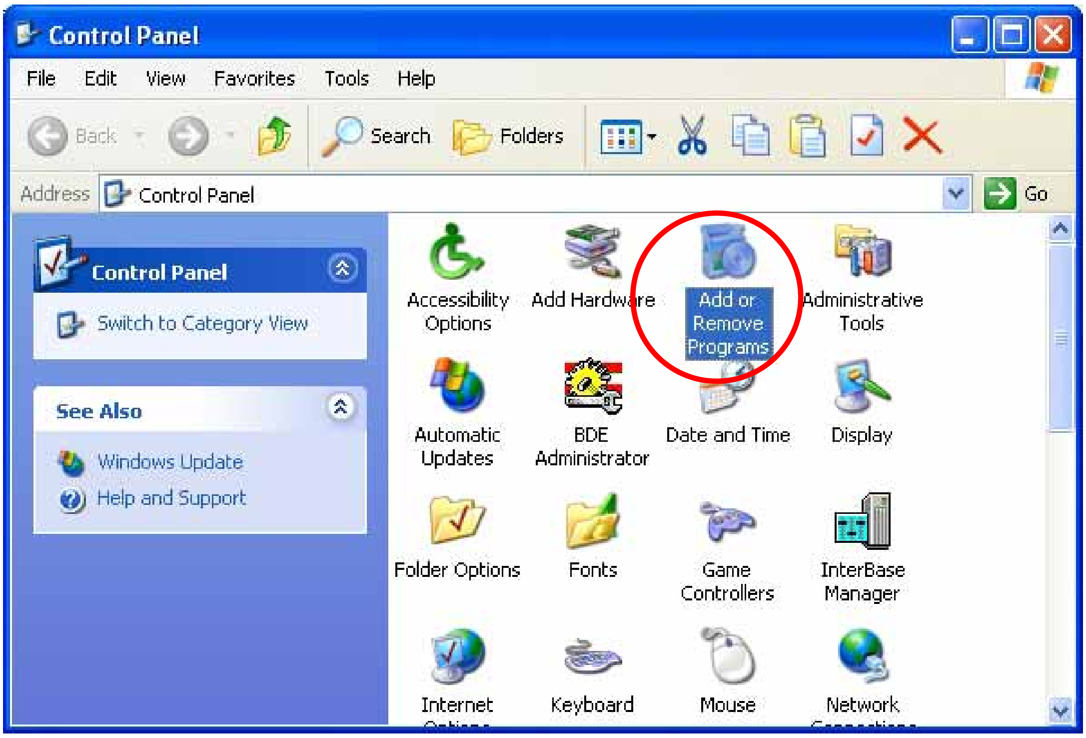


Figure 5-8: "Add or Remove Programs"

Step 3: Find out the GW-7433D Utility, and click the Change/Remove button. See figure 5-9.

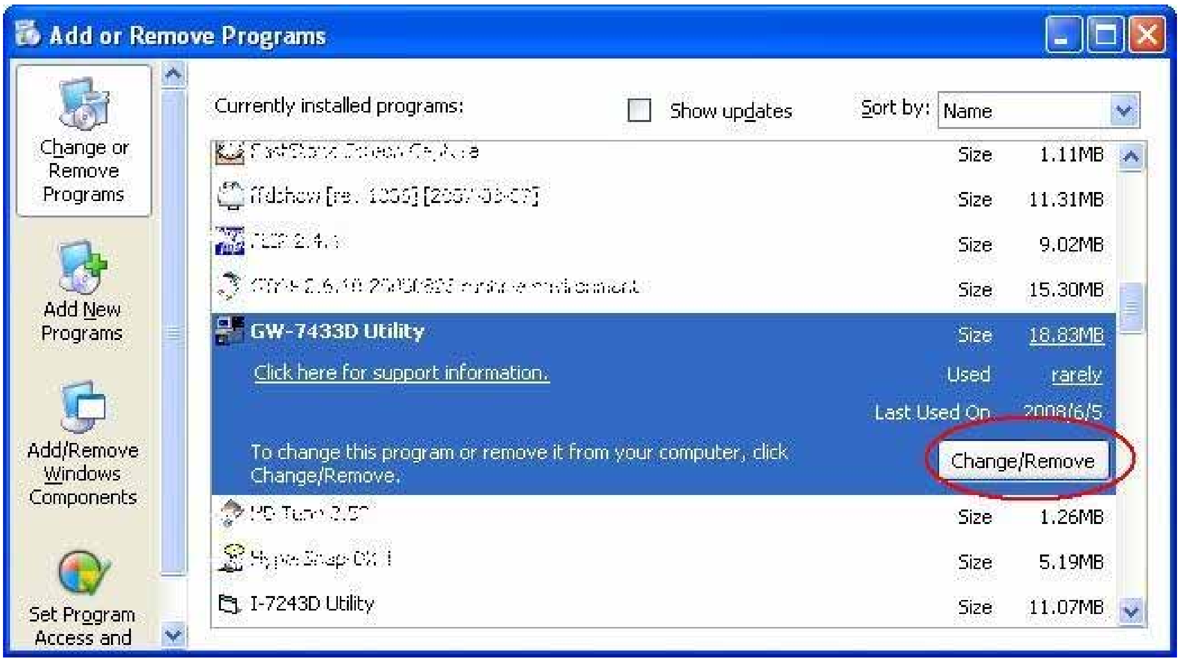


Figure 5-9: Click "Change/Remove" button

Step 4: Select the "Remove" option button, and press the "Next" button to remove GW-7433D Utility. See figure 5-10.

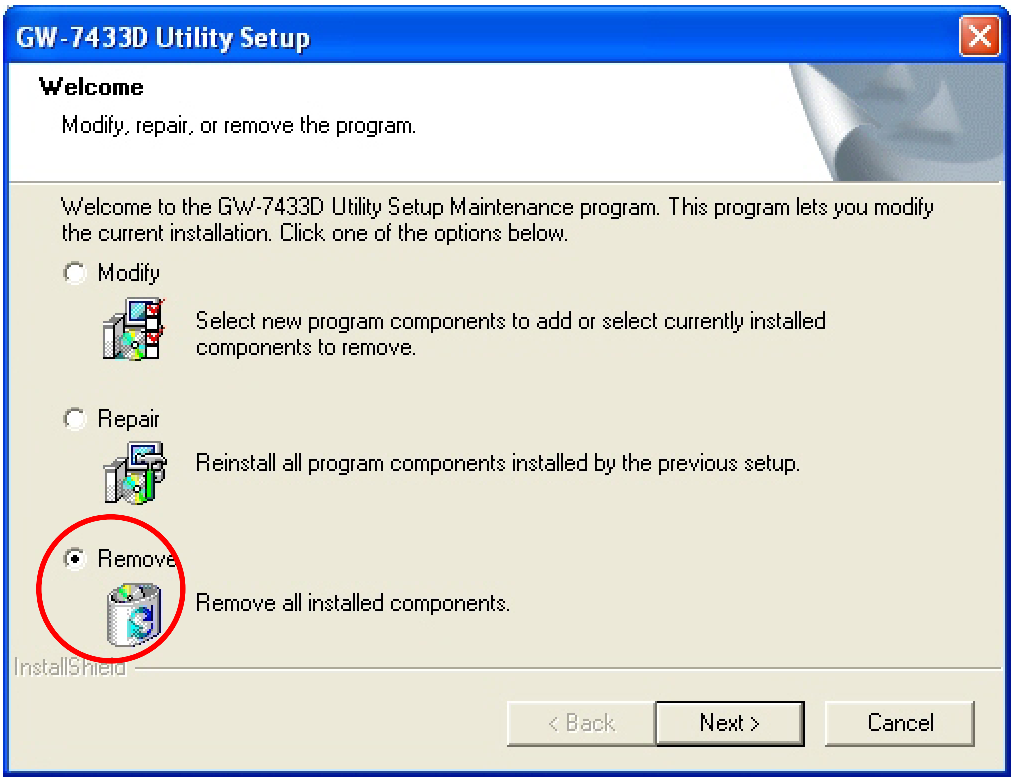


Figure 5-10: "Modify, repair, or remove the program" dialog

Step 5: Click the button "Yes" to remove the software as shown in figure 5-11.

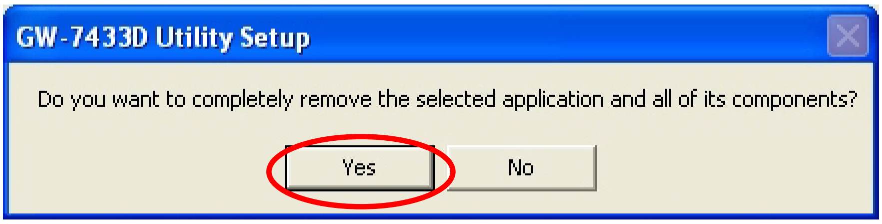


Figure 5-11: Click the button "Yes" to remove the software

Step 6: Finally, click the "Finish" button to finish the uninstall process.

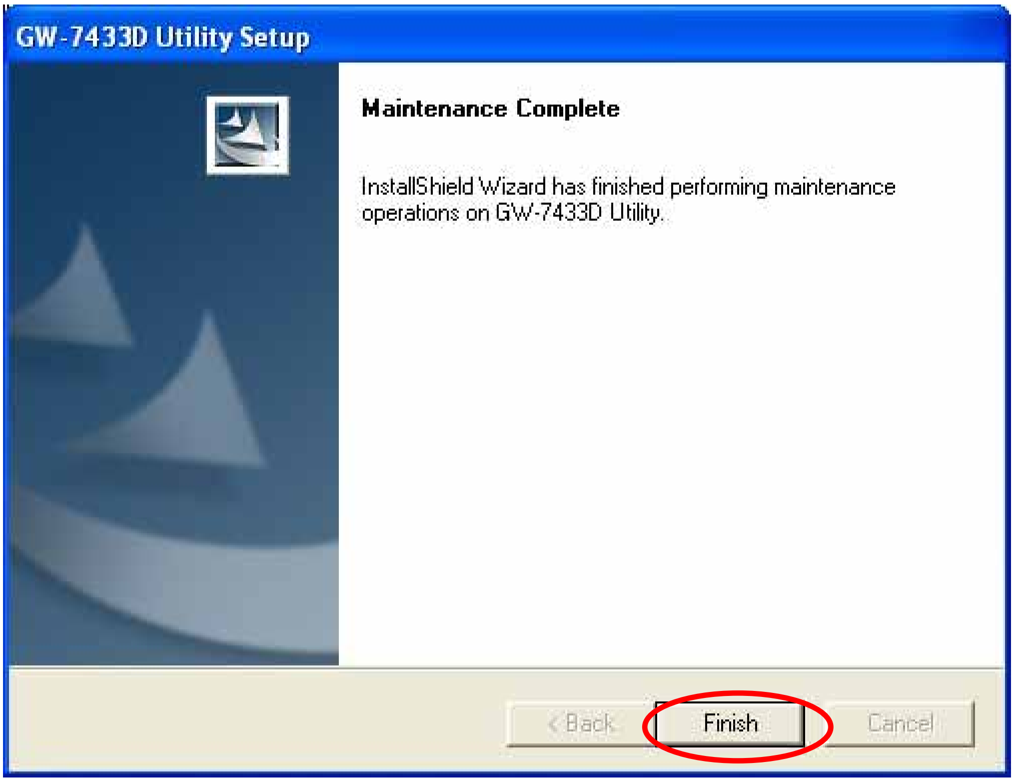
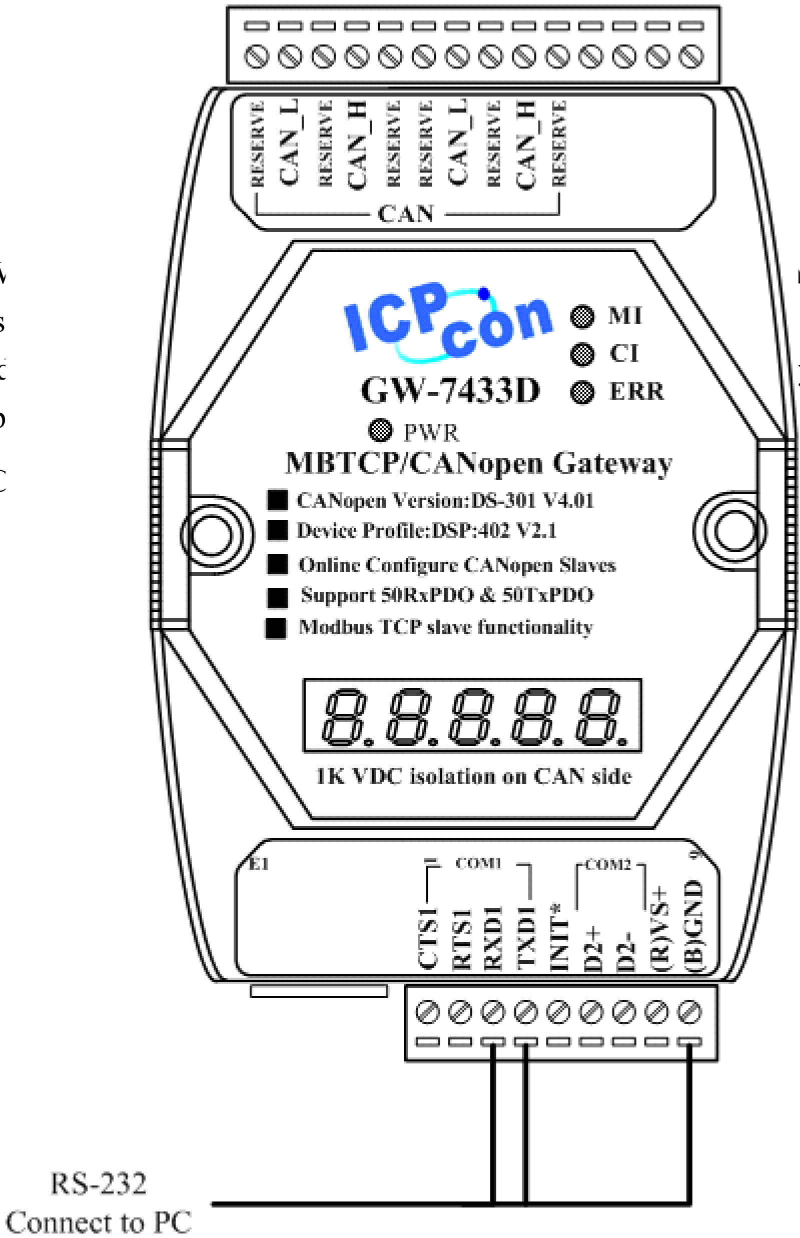


Figure 5-12: "Maintenance Complete" dialog.

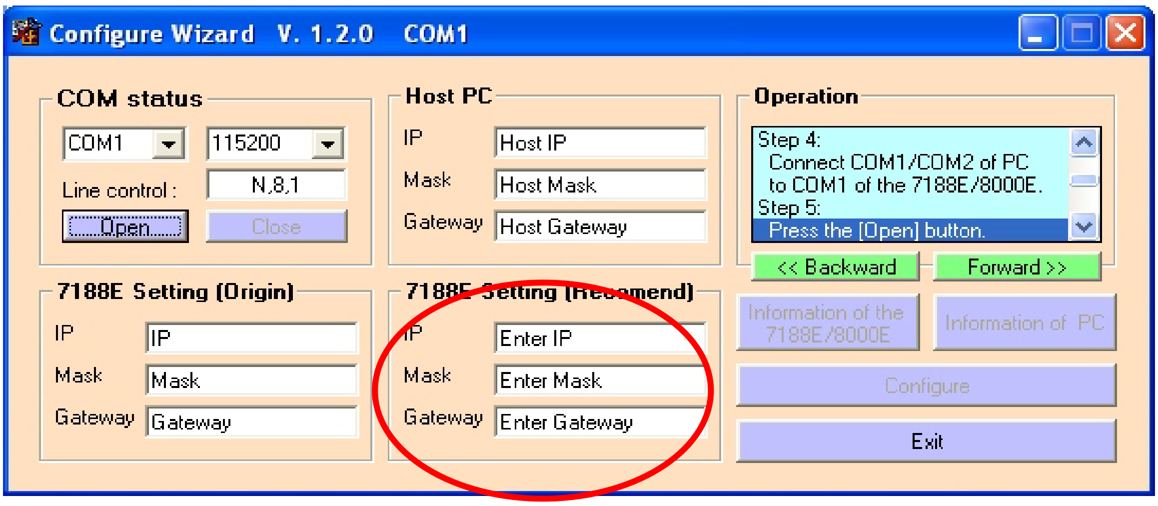
5.3

W ... ieed to adjust each setting
to s ... following procedure will
guic ... y tools. The configuration
step

1. C



2. Provide the proper power (the 10~30 DC volts) to the GW-7433D module.
3. Configure the network settings (IP, Mask, and Gateway) for the Configure Wizard Utility.



To Use the Configuration Wizard, you must install PCDiag. It will be located in the following path: 8000CD:\Napdos\7188e\TCP\PCDiag\Setup\Setup.exe

4. After using configure wizard to set the network parameters of GW-7433D, please power-off and disconnect the RS-232 cable then connect Ethernet cable to GW-7433D and power-on the power source of it again.
5. Then the GW-7433D module's Power LED will flash approximately once per 0.5 second. And the 5-digits 7-segment LED will scroll to display some messages, please refer to section 2.2.3 to check what the status it is.
6. The user must run the GW-7433D's Utility software after they have made a wire connection between the PC and the GW-7433D via the network cable
7. Then click the "Connect to GW-7433D" button. The "Connect to GW-7433D setting" window will be pop-up. Key-in the IP of the GW-7433D and press the "Start Connect" button in order to connect with it. As shown in the following figure.

(Note: if user selects the "To initial modules" item, the GW-7433D will not to read the parameters from the GW-7433D EEPROM. When user first to use the GW-7433D or present Error Code please selects the item.)

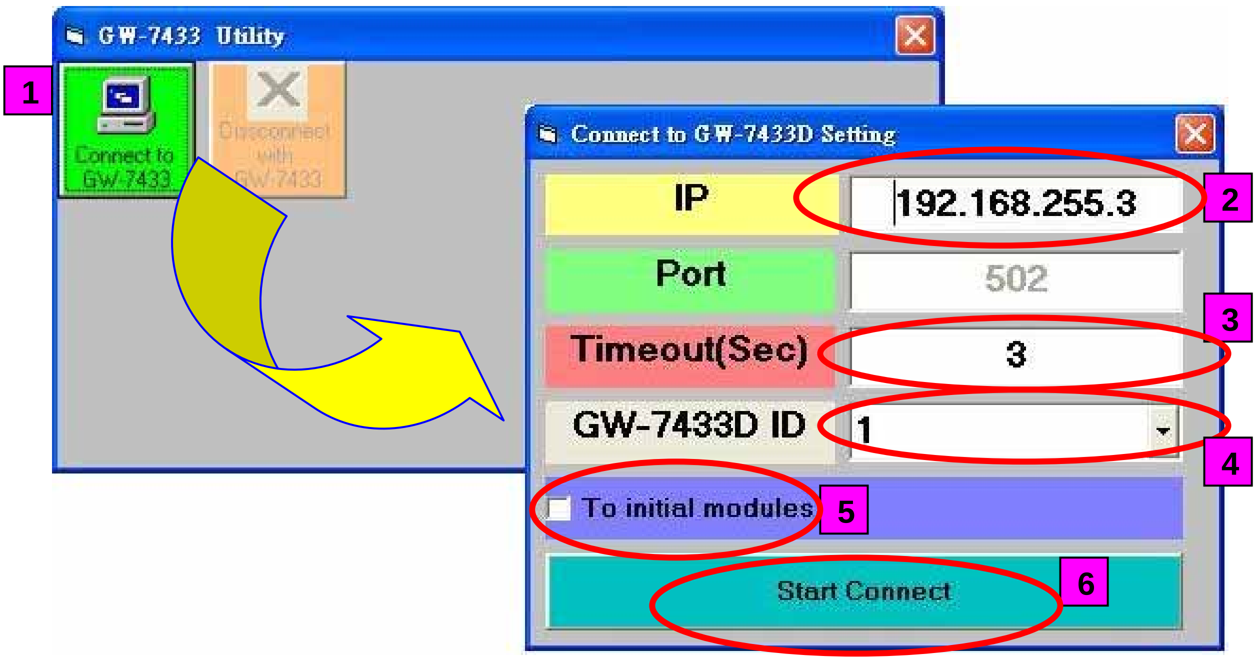


Figure 5-13: Connection setting of GW-7433D

8. If the GW-7433D is online and work normally, the Utility tool will display the "MappingModules" windows. Shown in the following figure.

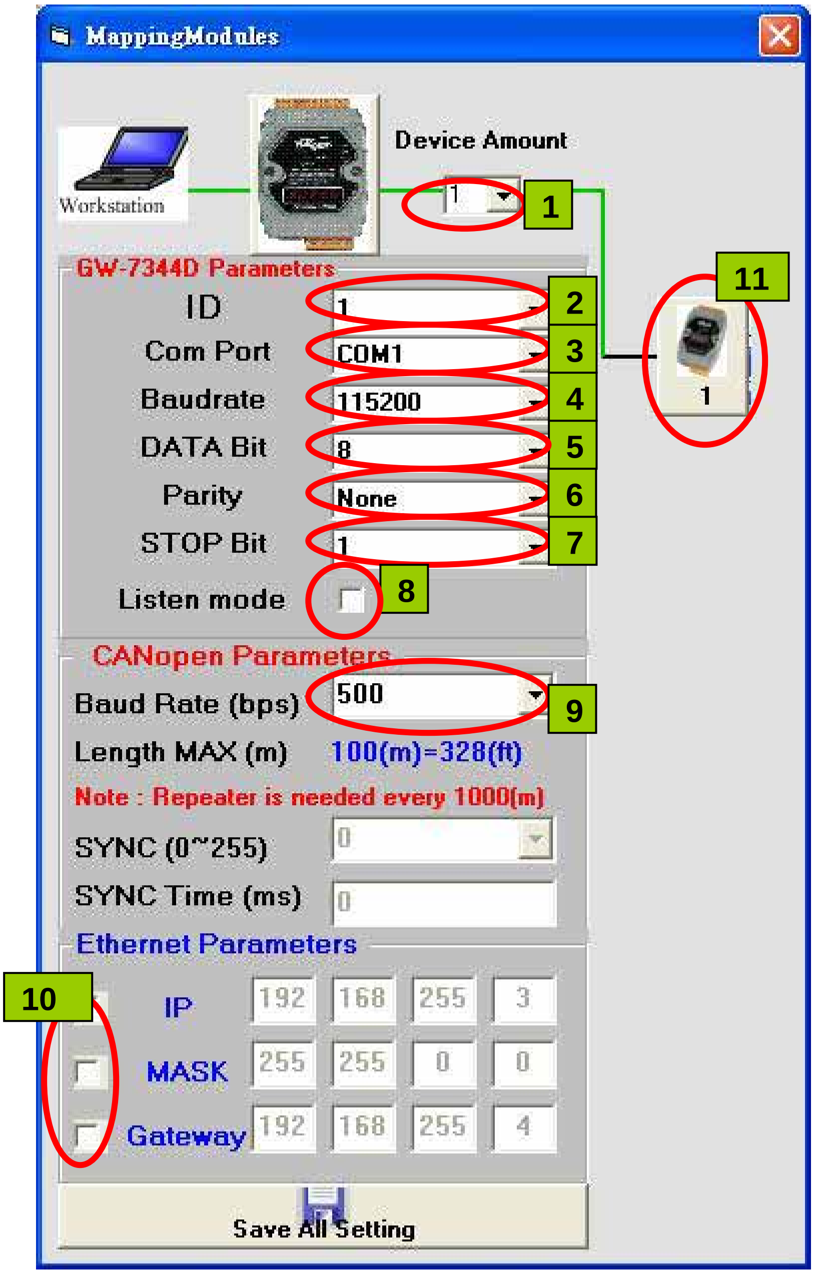


Figure 5-14: Connect to the configuration mode of the GW-7433D

1 → Select the device amount of CANopen slaves. Maximum amount of CANopen slaves = 10.

2 → Set the GW-7433D station number of Modbus RTU slave.

3 → Select the COM port number of GW-7433D to be the Modbus RTU COM port

4 → Select the baud rate of the Modbus RTU COM Port.

5 → Set the data bit of the Modbus RTU COM port.

6 → Set the parity of the Modbus RTU COM port.

7 → Set the stop bit of the Modbus RTU COM port.

8 → Select if the GW-7433D is in listen mode or normal mode.
(Please refer to the section 5.4 to know how to configure the utility in listen mode.).

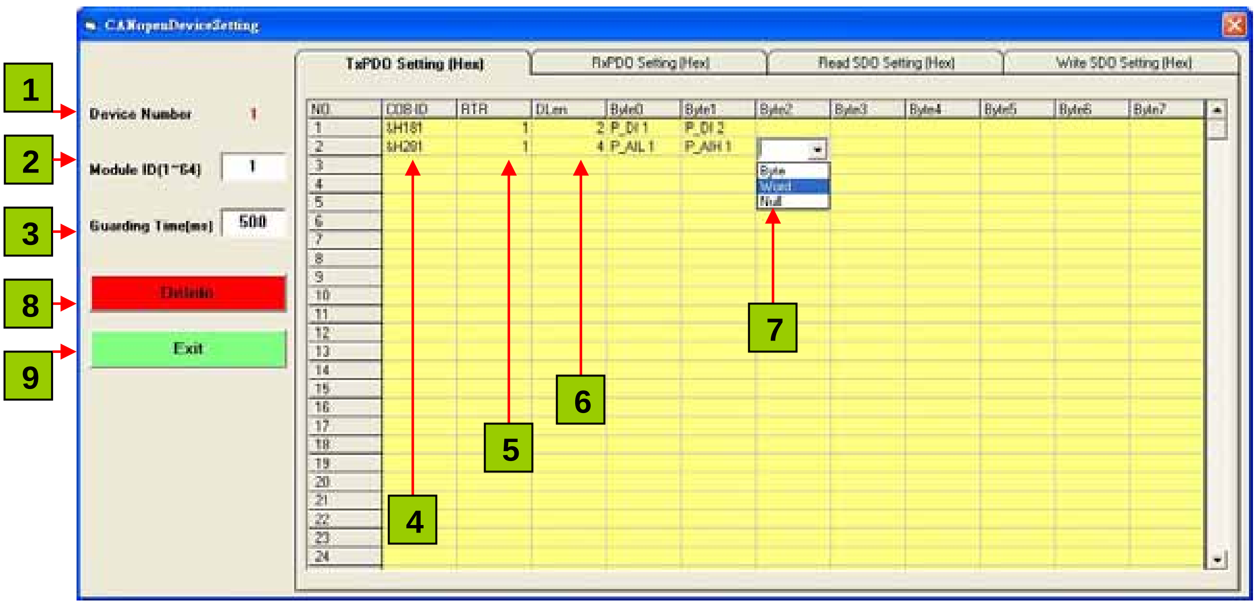
9 → Set the baud rate of CAN bus.

10 → Change the GW-7433D IP, MASK or Gateway.

11 → When user finish step 1 to step 10, please click "the button", the "CANopenDeviceSetting" window will be pop-up then user can set the CANopen slave device parameters into the GW-7433D.

9. Setting the RxPDO, TxPDO, RxSDO, and TxSDO of "CANopenDeviceSetting" window. As follow figures.

(1). TxPDO Setting.



1 → If the Device Amount is 5 then the Device Number is 1 to 5 in five "CANopenDeviceSetting" windows.

2 → Setting the CANopen slave device's station's ID.

3 → Setting the CANopen slave device's Guarding.

4 → Key in the COB-ID(hex) of the CANopen slave device's mapping parameters.

5 → Select the RTR, in the TxPDO is selection "1", and RxPDO is selection "NO".

6 → Select the length of bytes of data's. user have to select the correct DLen, otherwise the GW-7433D firmware will present some error code, then user have to reset the parameters again.

7 → Select the byte is DI or AI. If user select DI, the byte will present "P_DI1" or "P_DI2"…."P_DIx"….etc.

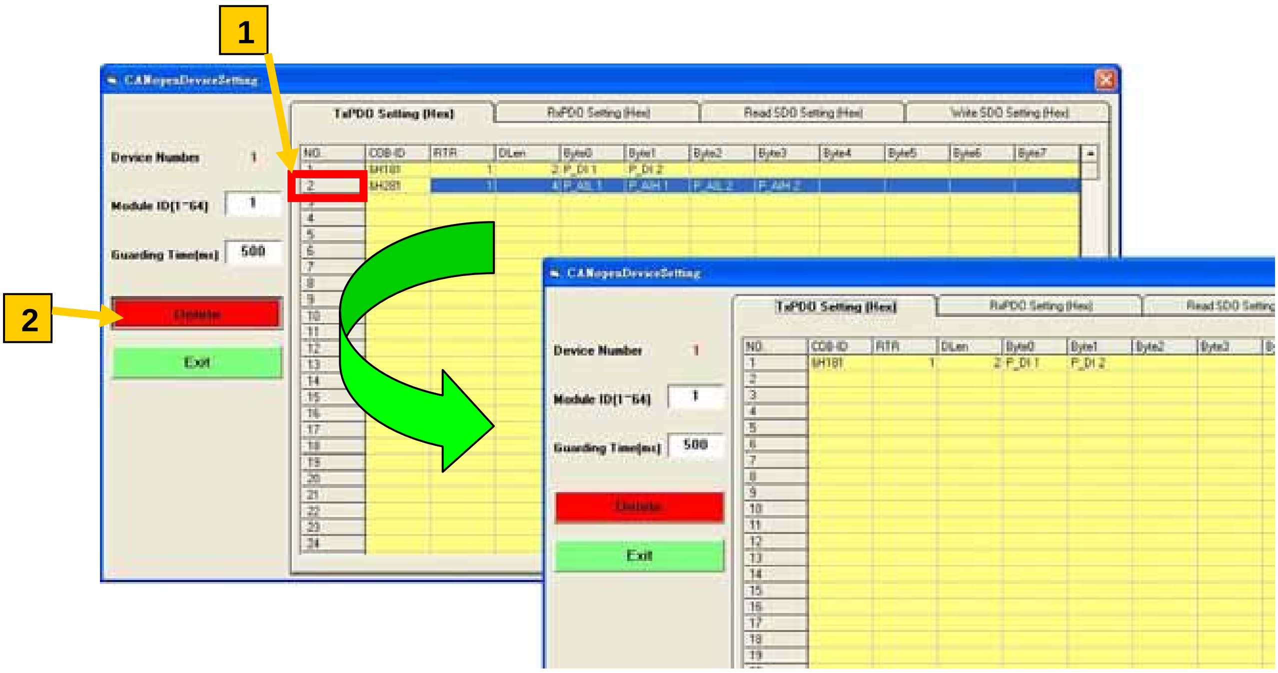
(Note: DI use one byte, AI use two bytes)

P → PDO.

DI →DI.

x → DI counts.

8 → Delete button. As follow figures.



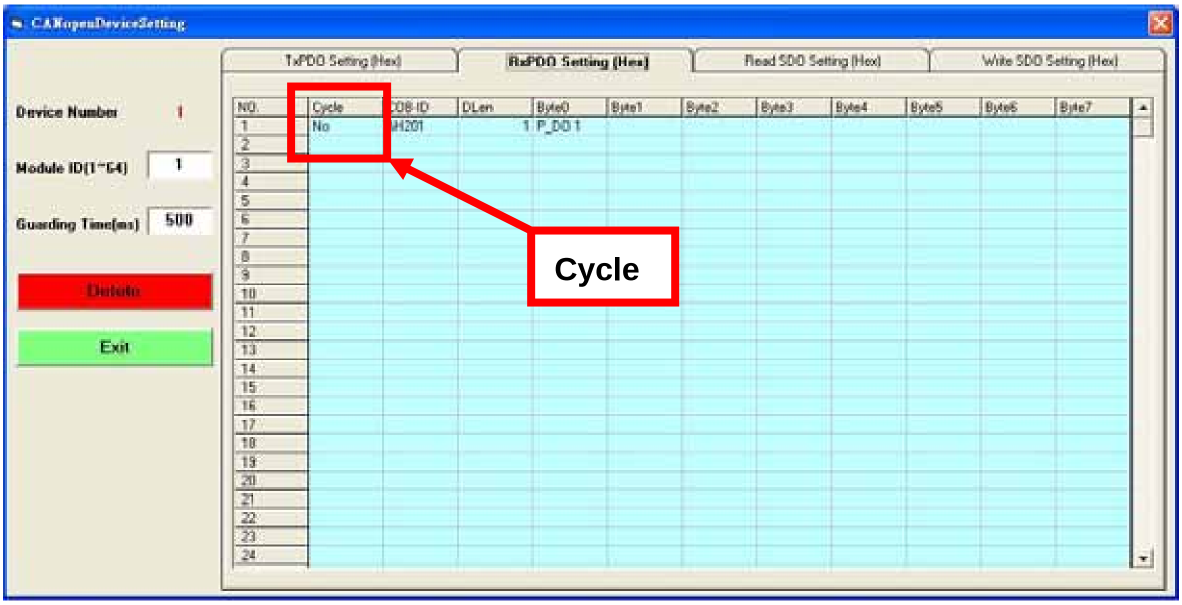
1 → Please move the mouse to "NO" column, and select user want to delete row, then click it.

2 → When user finishes the 1 tep, then click the "Delete" button.

(2). RxPDO Setting

User can set the RxPDO parameters by TxPDO's way, as follow figures.

(note: the Cycle column select always "NO").



(3). TxSDO Setting.

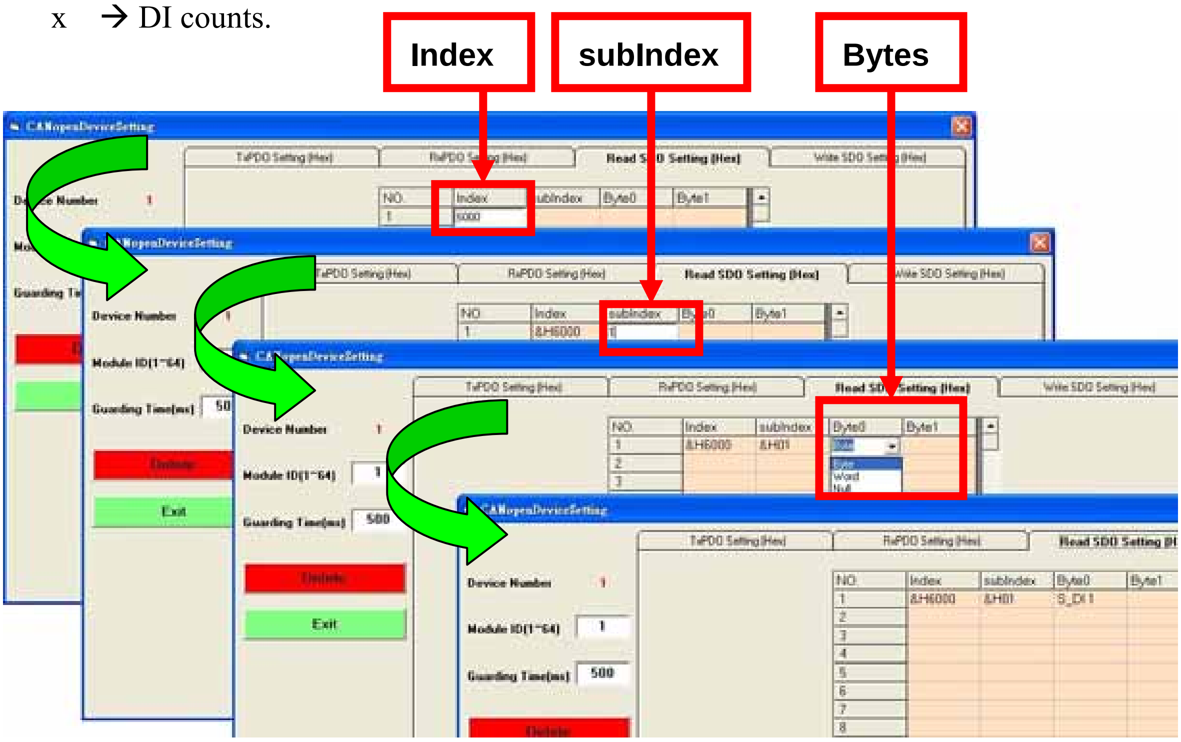
Key in the index (hex) and subindex (hex) of the CANopen slave device's mapping parameters, and select the byte value is DI or AI value. Then it will present "S_DI1","S_DI2"….."S_DIx"…etc.

(Note: DI use one byte, AI use two bytes)

S → SDO.

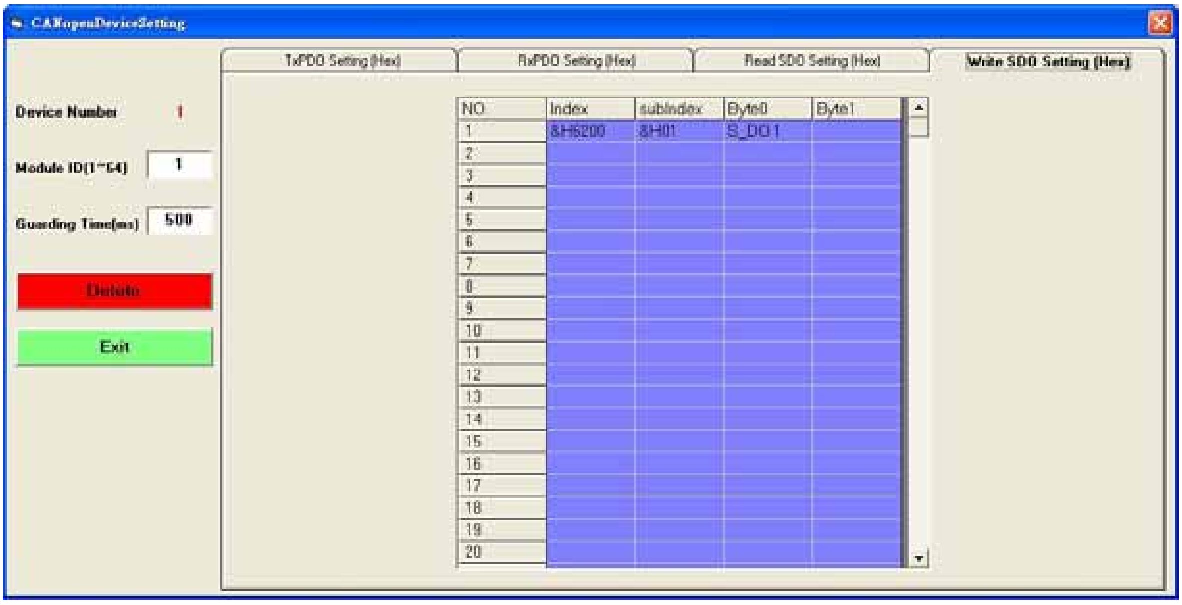
DI → DI.

x → DI counts.



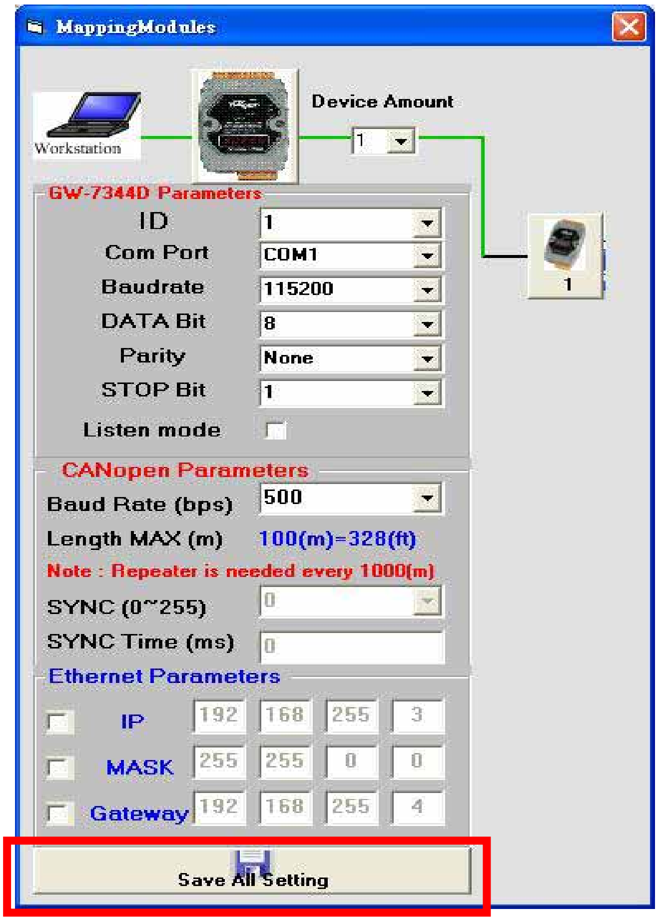
(4). RxSDO Setting

User can set the RxSDO parameters by TxSDO's way, as follow figures.



9 → Exit the "CANopenDeviceSetting" window.

10. When exit the "CANopenDeviceSetting" window, the "MappingModules" window Will pop-up again, as follow figure.



Then user can click the “Save All Setting” button to write the parameters into the GW-7433D’s EEPROM, and the GW-7433D will auto-run the firmware to communication with CANopen slave devices.

After finish the step of above, the “ShowMapping” window will be display. As follow figure.

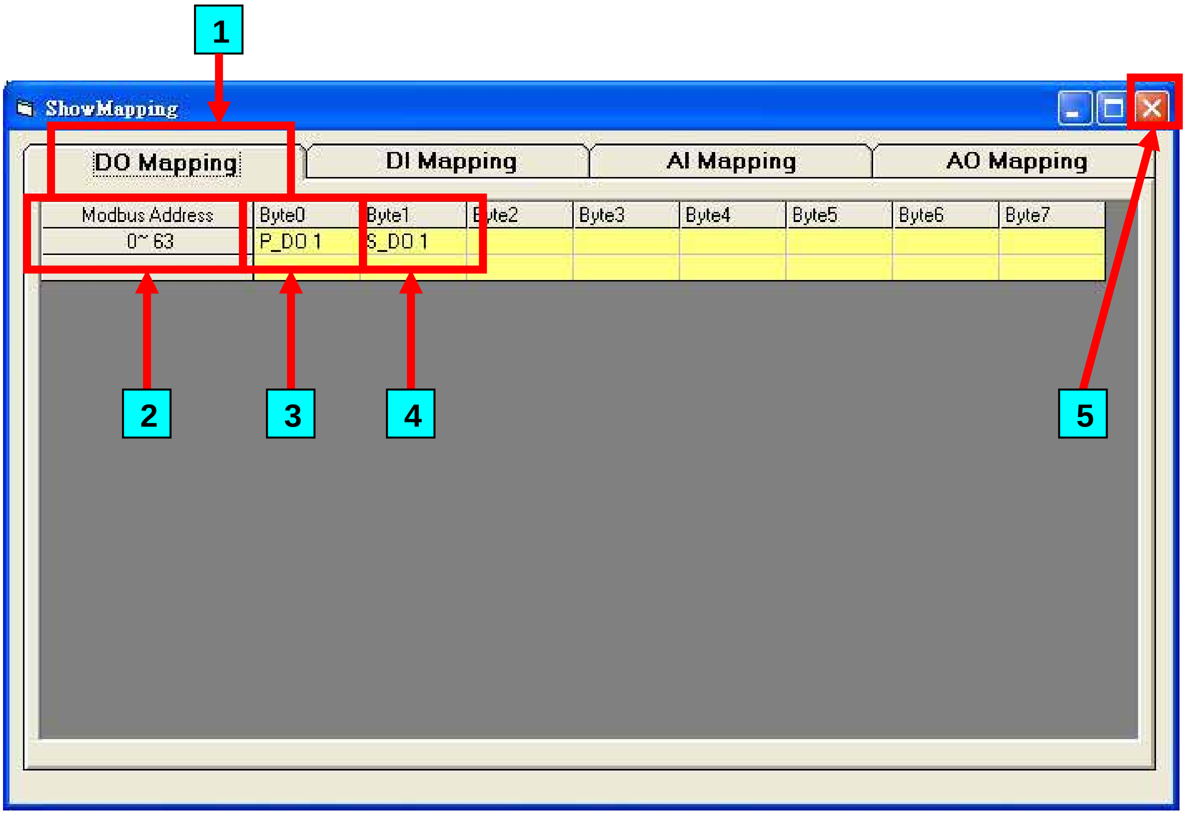


Figure 5-15: CANopen slave device mapping to the Modbus TCP/RTU table

1 → The DO register mapping table of CANopen slave device to the Modbus TCP/RTU.

2 → The column is Modbus TCP/RTU DO address, the row range is 0 ~63.

3 → The row is CANopen slave device’s data byte address mapping to Modbus TCP/RTU address.

4 → The same as above.

5 → Close the “ShowMapping” window.

The “DI Mapping,” ”AI Mapping” and “AO Mapping” of mapping type the same as the “DO Mapping” of mapping type.

In these tab, the PDO and SDO arrayal of type is form PDO bytes to SDO bytes. When user over view the “ShowMapping” window, Plase close the “ShowMapping” window, then the GW-7433D Utility will display the “MappingModules” window, then close the “MappingModules” window too, and the GW-7433D Utility will display the “GW-7433D Utility ver(1.0.0)” window and click the “Disconnect with GW-7433D” button and close the window to finish the GW-7433D Utility setting. As follow figure.

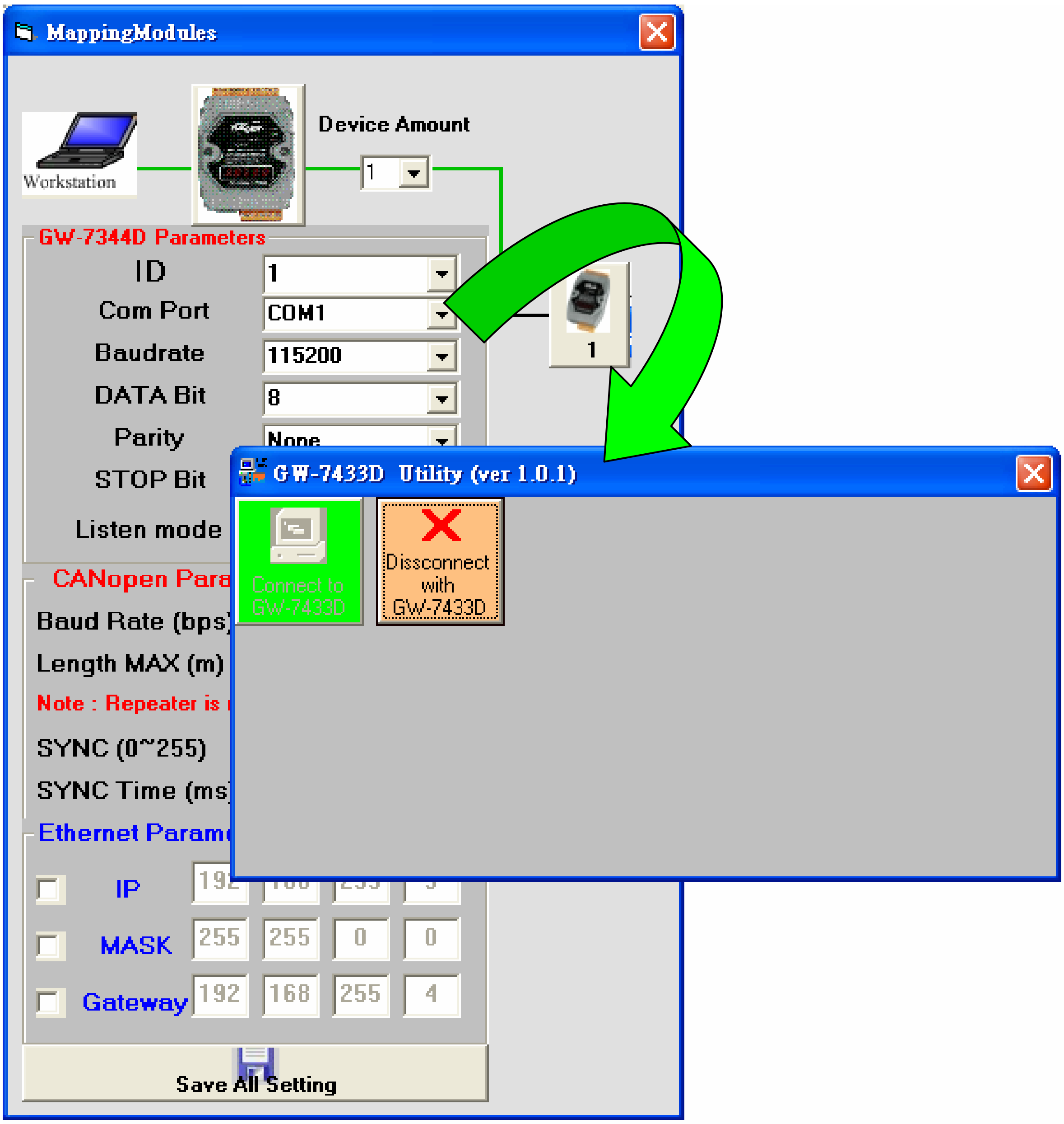
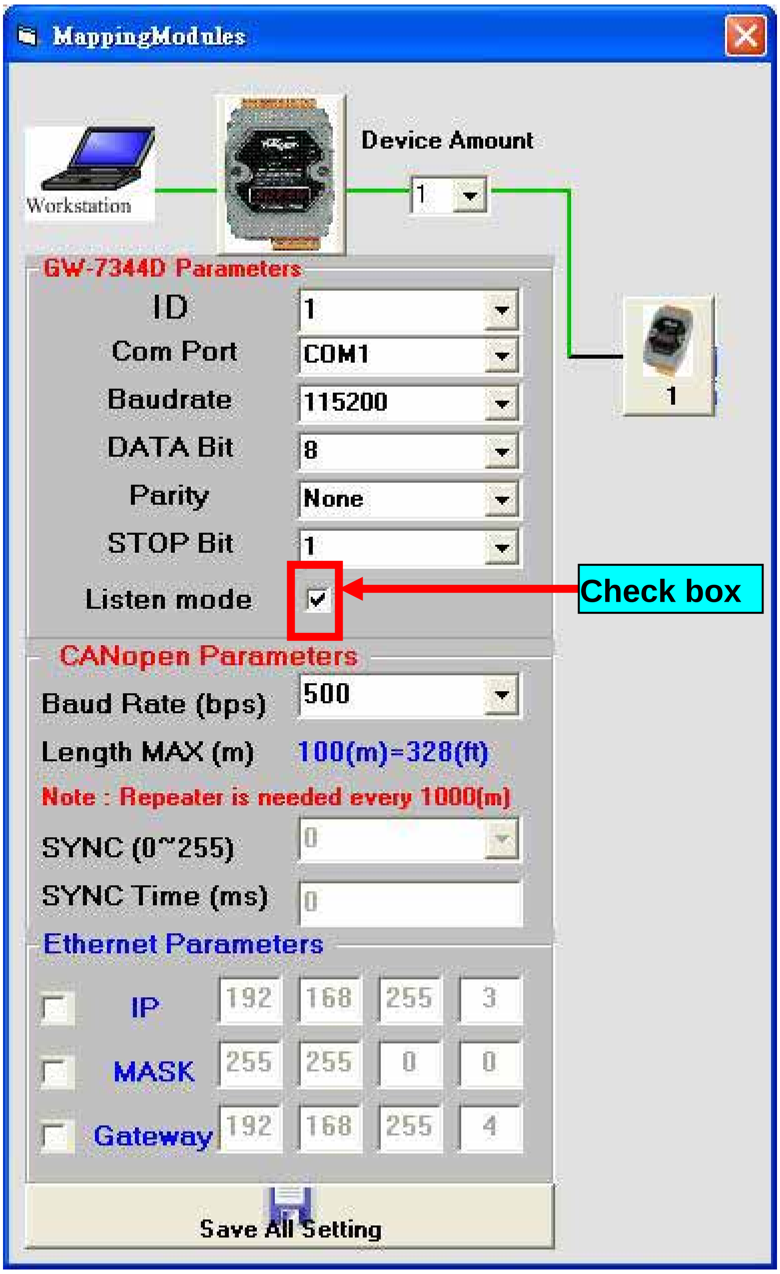


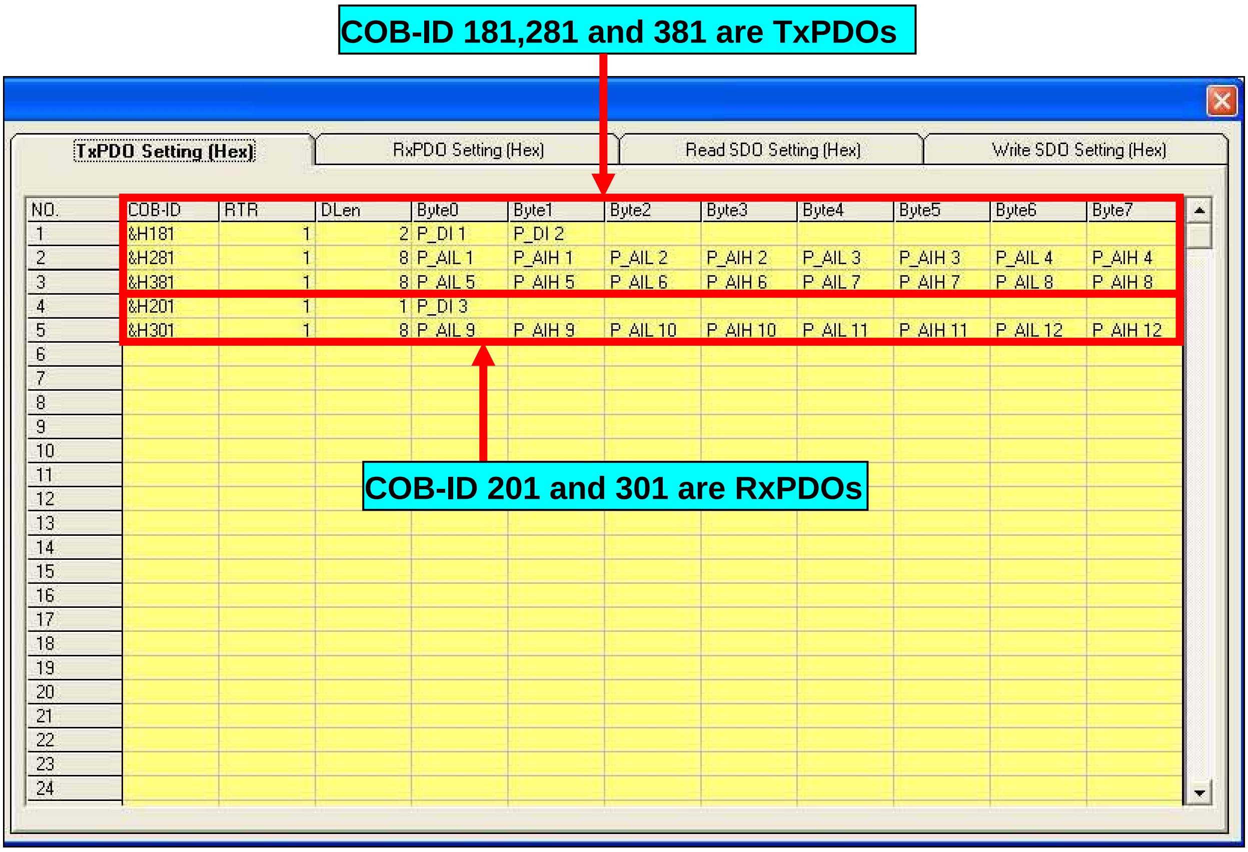
Figure 5-16: Close the GW-7433D Utility setting

5.4 How to configure the utility in listen mode

In some applications, there is already one CANopen master in the CANopen network. If users add one GW-7433D in this kind of CANopen network, the PDO commands from these two CANopen master may conflict. In this case, users can set the GW-7433D into listen mode. Therefore, GW-7433D will only listen to the PDO commands from another CANopen master, and transfer the information into Modbus IO information. If users want to use listen mode of GW-7433D, the first step is to check the check box as follows.



In listen mode, only TxPDO and RxPSO are supported. After selecting the listen mode, users have to input RxPDO and TxPDO configurations into the "CANopenDevicesSetting" window. For TxPDO, users just put the TxPDO configurations in the TxPDO setting window. Specially, if users want to read the data of RxPDOs which are sent from another CANopen master, the configurations of RxPDOs must be added into the TxPDO setting window as follows.



NO.	COB-ID	RTR	DLen	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
1	&H181	1	2	P_DI 1	P_DI 2						
2	&H281	1	8	P_AIL 1	P_AIH 1	P_AIL 2	P_AIH 2	P_AIL 3	P_AIH 3	P_AIL 4	P_AIH 4
3	&H381	1	8	P_AIL 5	P_AIH 5	P_AIL 6	P_AIH 6	P_AIL 7	P_AIH 7	P_AIL 8	P_AIH 8
4	&H201	1	1	P_DI 3							
5	&H301	1	8	P_AIL 9	P_AIH 9	P_AIL 10	P_AIH 10	P_AIL 11	P_AIH 11	P_AIL 12	P_AIH 12
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											

When the configuration of the "CANopenDrivcesSetting" window is finished and the parameters are saved into GW-7433D module by GW-7433D Utility, users can read the TxPDO and RxPDO information by Modbus TCP/RTU command.

5.5 How to get the I/O data of each CANopen slave device

The GW-7433D supports an “Internal Memory” for saving I/O data. After using Utility tool to set the memory mapping of the I/O devices, users can get/set the I/O data according to the mapping address of each I/O data. When accessing these areas, with Modbus commands, the addressing is according to the following tables.

Users can use Modubs Function code 4, read input registers, to get the input data values from GW-7433D’s input data area.

Word Address (3xxxx)	**Byte Address**	
0000h	0000h	0001h
0001h	0002h	0003h
0002h	0004h	0005h
: :	: :	: :
00FFh	01FEh	01FFh

Table 5-1: Input Addressing

Users can use Modbus Function code 16, force multiple registers, to set the output data values into GW-7433D’s output data area.

Word Address (4xxxx)	**Byte Address**	
0000h	0000h	0001h
0001h	0002h	0003h
0002h	0004h	0005h
: :	: :	: :
00FFh	01FEh	01FFh

Table 5-2: Output Addressing

5.5 MBTCP & MBRTU Tools

The MBTCP is a Modbus TCP client tool with source code in VB6. The MBRTU is a Modbus RTU master tool with source code in VB6. These two tools are made by ICP DAS and they are helpful when users want to diagnostic the Modbus TCP server devices or Modbus RTU slaves. Users can free download from the following web site:

http://www.icpdas.com/products/PAC/i-8000/modbus_web_download.htm