GW-7472

EtherNet/IP to Modbus RTU Gateway

User Manual

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, starting 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 published 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 © 2010 by ICP DAS Co., Ltd. All rights are reserved.

Trademark

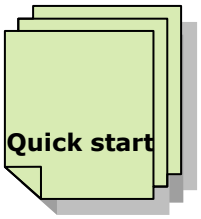
The names used for identification only may be registered trademarks of their respective companies.

Table of Contents

Packing List

The shipping package includes the following items:

- One GW-7472 series hardware module
- One printed Quick Start Guide
- One software utility CD



Note!!

If any of these items is missed or damaged, contact the local distributor for more information. Save the shipping materials and cartons in case you want to ship in the future.

More Information

Documentations

Fieldbus_CD:\EtherNetIP\Gateway\GW-7472\Document

Firmware

Fieldbus_CD:\EtherNetIP\Gateway\GW-7472\Firmware

Utility

Fieldbus_CD:\EtherNetIP\Gateway\GW-7472\Utility

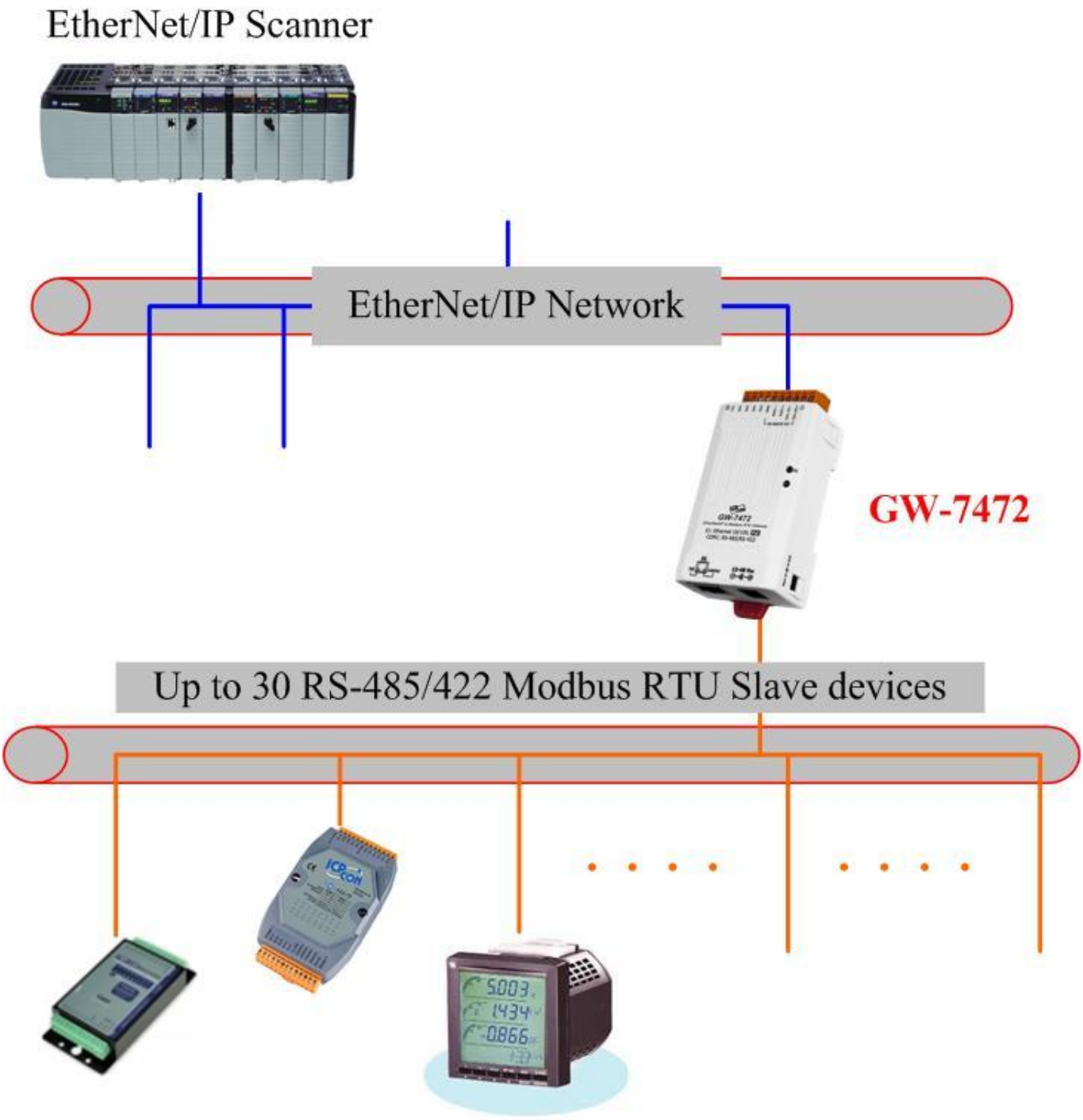
1. Introduction

The GW-7472 (EtherNet/IP adapter to Modbus RTU Master Gateway) is helpful for data-exchanging between the Modbus RTU network and the EtherNet/IP network. It reads the register data from the Modbus RTU slaves and publishes these data to the input register data of the EtherNet/IP scanner. The output data transmitted by the EtherNet/IP scanner are updated to the register data of Modbus RTU slaves via the GW-7472. Through the GW-7472, all of the Modbus slaves can be regard as one EtherNet/IP adapter. The GW-7472 allows maximum 6 connections for the Explicit Messages and 1 connection for the Implicit Messages at the same time. It means that 7 EtherNet/IP scanners can connect to one GW-7472 at the same time.

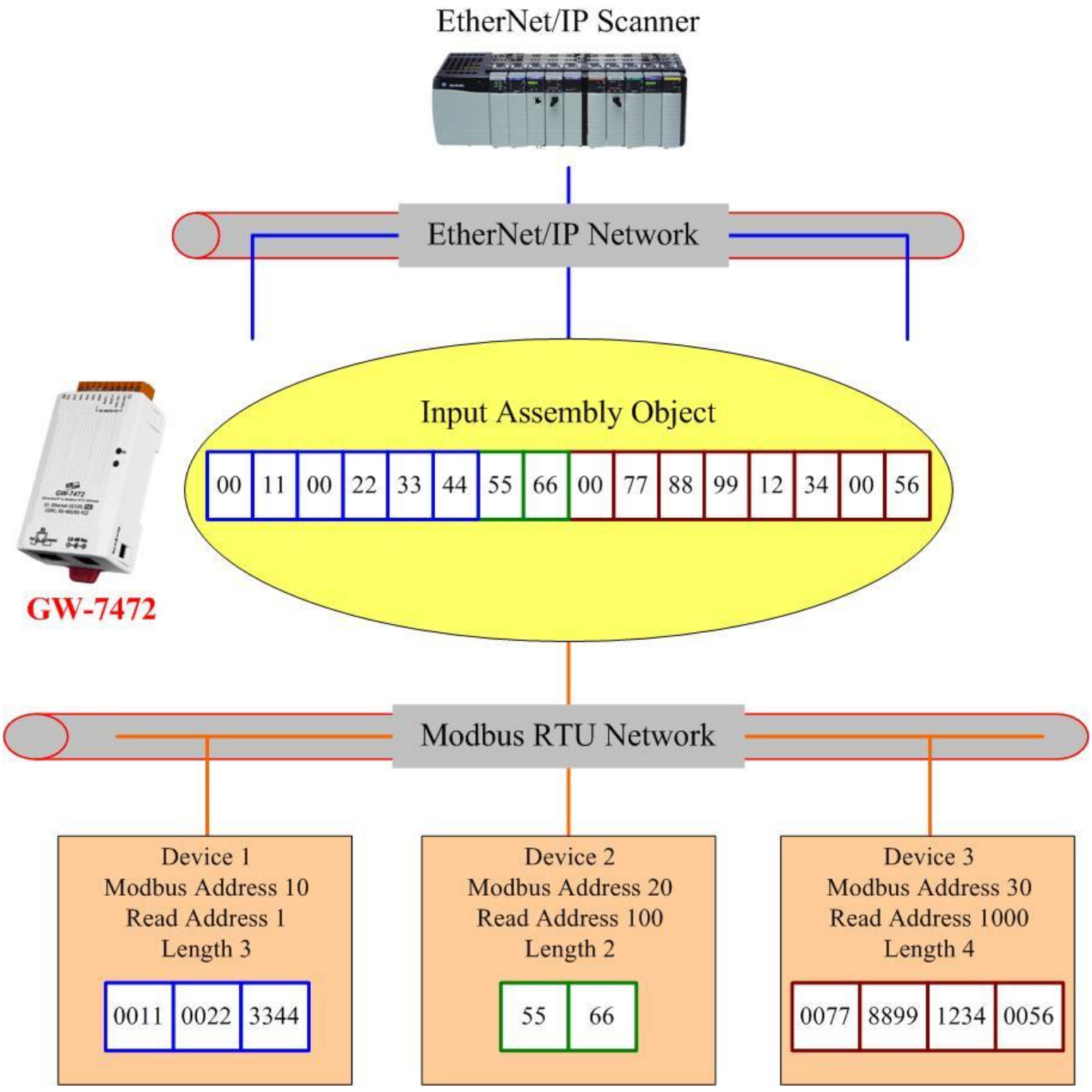
The Modbus RTU master functions of the GW-7472 can scan up to 30 Modbus RTU commands. After configuring the Modbus RTU master behavior of the GW-7472 and the mapping status between the Modbus RTU registers and EtherNet/IP registers by using the GW-7472 Utility tool, the input/output registers of the Modbus RTU slaves are mapping to the output/input registers of the EtherNet/IP adapter. While booting up, the GW-7472 scans the pre-defined register addresses in all of the Modbus RTU slaves according to the sequence defined in the utility tool. The input and output register data of the Modbus devices are updated as soon as the GW-7472 could.

In order to save the installation space, the GW-7472 is offered in an amazing tiny form-factor that makes it easy to install in anywhere, even directly attached to a serial device or embedded into a machine. The GW-7472 features a powerful 32-bit MCU to handle efficient network traffic and it provides the IEEE 802.3af-compliant (classification, Class 1) Power over Ethernet (PoE) with a standard category 5 Ethernet cable. Through the NS-205PSE, Poe switch, the GW-7472 can be powered via the Ethernet cable. When there is no PoE switch on site, the GW-7472 provide another way to be powered from DC adapters. These two power interfaces are redundant. If one fails, another will take it over to supply the proper power to the GW-7472.

The following figure briefs the concept of the data exchange between the EtherNet/IP and the Modbus RTU network. In this system, there are three Modbus RTU slaves connected to the Modbus RTU master provided by the GW-7472. The EtherNet/IP adapter interface of the GW-7472 is connected to an EtherNet/IP scanner through an Ethernet switch. Register data of the three Modbus RTU slaves is presented to the EtherNet/IP scanner as the I/O data.



The following figure illustrates how to deploy the register data of the three Modbus devices in the previous example to the EtherNet/IP scanner. Assume that there are 3, 2, and 4 input registers data in the Modbus slaves No.1, No.2 and No.3 respectively. The data format of the register in the Modbus slave No.1, and No.3 is WORD, and that in the Modbus slave No.2 is BYTE. All of these input registers are mapping to the corresponding input registers of the EtherNet/IP adapter of the GW-7472 sequentially by using the BYTE format.



The output register data of the Modbus RTU slaves are mapping in exactly the same way. The Modbus slave output registers are mapping as the output registers of the EtherNet/IP adapter of the GW-7472. Users can set the maximum 500 bytes for input data and 500 bytes for output data which are mapping to the EtherNet/IP adapter of the GW-7472. All of these configurations are defined by using the GW-7472 Utility tool. While the GW-7472 gets the EtherNet/IP commands from the EtherNet/IP scanner, it collects the input register data from Modbus RTU slaves and updates the output register data to the Modbus RTU slaves as soon as possible.

2. Hardware Information

2.1 Specifications

Model	GW-7472
System	
CPU	32-bit MCU
Communication Interface	
Ethernet	10/100 Base-TX, 8-pin RJ-45 x 1, (Auto-negotiating, Auto-MDI/MDIX, LED indicator) PoE (IEEE 802.3af, Class 1)
COM1	2-wire RS-485 / 4-wire RS-422
Self-Tuner	Yes, automatic RS-485 direction control
UART	16c550 or compatible
COM Port Format	
Baud Rate	1200 to 115200 bps.
Data Bit	7, 8
Parity	None, Odd, Even
Stop Bit	1, 2
General	
Power Input	PoE: IEEE 802.3af, Class 1 DC jack: +12 ~ 48 V_{DC}
Power Consumption	0.05 A @ 24 VDC
Connector	10-Pin Removable Terminal Block x 1
Mounting	DIN-Rail
Flammability	Fire Retardant Materials (UL94-V0 Level)
Operating Temperature	-25° ~ 75°C
Storage Temperature	-30° ~ 80°C
Humidity	10 ~ 90% RH, non-condensing

2.2 Features

General Features:

- Powerful 32-bit MCU handles efficient network traffic
- 10/100 Base-TX Ethernet, RJ-45 x1
  (Auto-negotiating, auto MDI/MDIX, LED Indicators)
- Redundant power inputs: PoE (IEEE 802.3af, Class 1) and DC jack
- Automatically RS-485 direction control
- Support ARP,TCP, UDP, ICMP, DHCP, BOOTP and TFTP protocols
- Easy firmware update via Ethernet
- Removable terminal block connector
- Tiny form-factor and low power consumption
- RoHS compliant with Halogen-free
- Fire retardant materials (UL94-V0 Level)

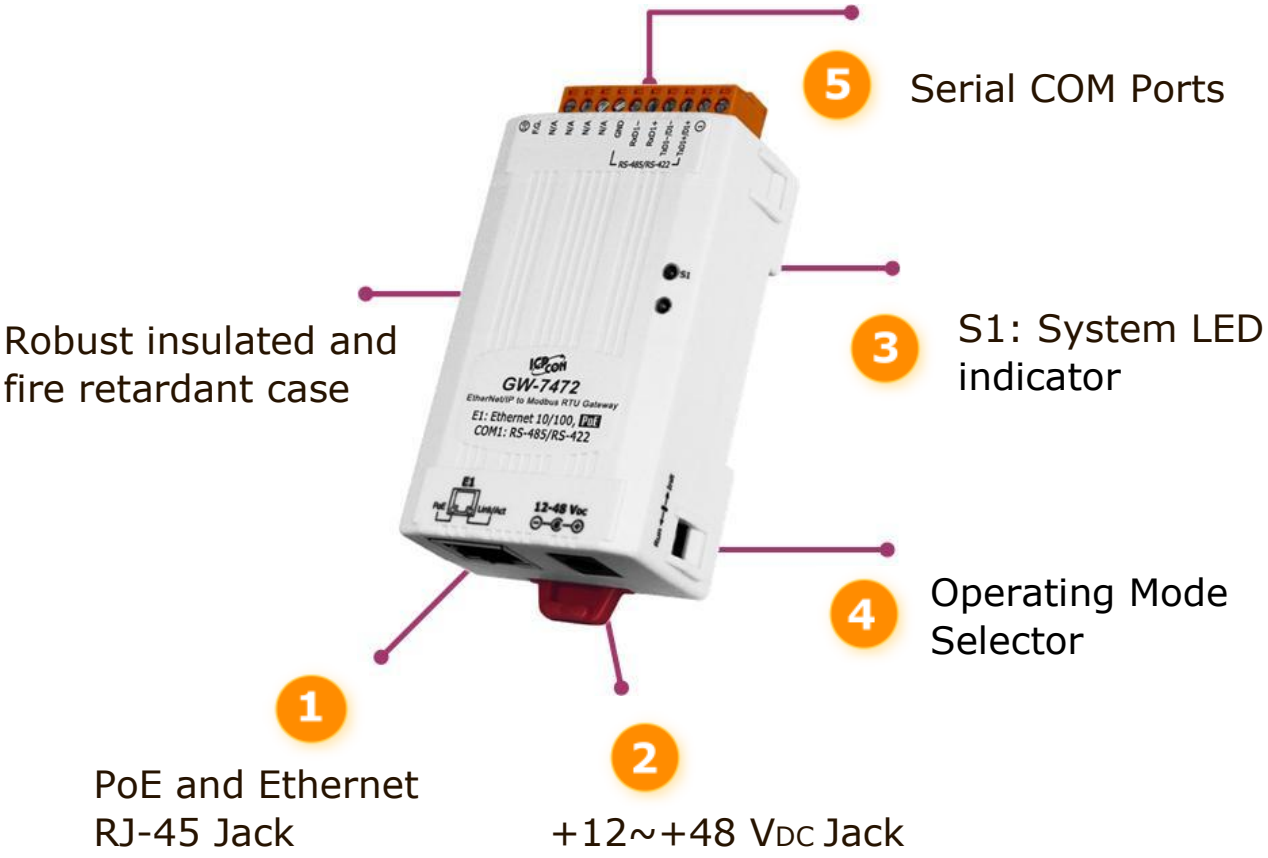
EtherNet/IP Features:

- Ethernet Protocol: EtherNet/IP adapter
- Maximum number of connections for Explicit Messages: 6
- Maximum number of connections for Implicit Messages: 1
- Supported I/O connection methods:
  - Transport and trigger: Exclusive-Owner, Cyclic
  - Original to Target Type: POINT2POINT
  - Target to Original Type: POINT2POINT, MULTICAST
- Device Configuration Option: EDS, Utility tool
- Address Configuration: DHCP, Utility tool
- EtherNet/IP Input/Output command data size: maximum 500 bytes
- The numbers of the Modbus RTU slave input registers mapping to the input registers of the EtherNet/IP adapter of the GW-7472: maximum 500 bytes
- The numbers of the Modbus RTU slave output registers mapping to the output registers of the EtherNet/IP adapter of the GW-7472: maximum 500 bytes

Modbus Features:

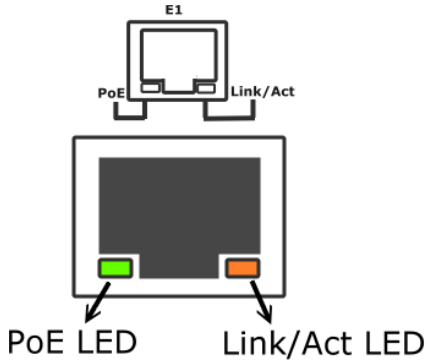
- Modbus Protocol: Modbus RTU Master
- Maximum support 30 Modbus RTU commands
- Supported Modbus RTU Function Codes:
  - 01_{hex}: Read Output Status
  - 02_{hex}: Read Input Status
  - 03_{hex}: Read Multiple Data Registers
  - 04_{hex}: Read Input Registers
  - $0F_{hex}$: Write Multiple Bits
  - 10_{hex}: Write Multiple Data Register
- Maximum data size of one Modbus RTU command: 240 bytes

2.3 GW-7472 Front View



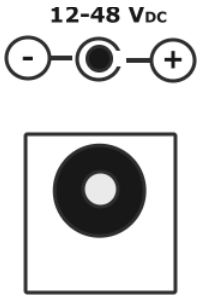
1. PoE and Ethernet RJ-45 Jack:

The GW-7472 is equipped with a RJ-45 jack for the 10/100 Base-TX Ethernet port and features networking capability. When the Ethernet link is detected and Ethernet packet is received, the **Link/Act LED (Orange)** indicator will be turned on. When the power is supplied via PoE (Power-over-Ethernet), the **PoE LED (Green)** indicator will be turned on.



2. +12~+48 VDC Jack:

The GW-7472 is equipped with a +12~+48 VDC jack for the power supply. When there is no PoE switch on site, the GW-7472 accepts the power from the DC adapter. Please refer to the following web site for more details.
http://www.icpdas.com/products/Accessories/power_supply/fra05-s12-su.htm



3. S1: System LED indicator:

After power on the GW-7472, the system LED indicator is as follows:

Function	System LED Action
Running Firmware	Flashing per second
Hardware checking error	Flashing per 0.3 seconds
Hardware error	blank

4. Operating Mode Selector:

Init Mode: Configuration mode
Run Mode: Firmware running mode

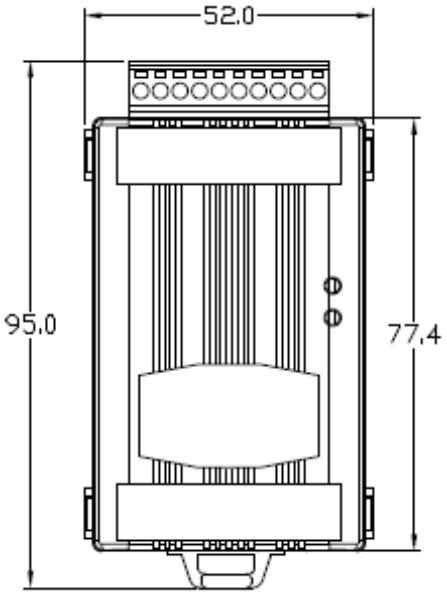
Generally, the switch is always in the Run position while the GW-7472 works. Only when updating or setting the GW-7472, the switch needs to be set to the Init position. Move the switch to the Run position and then re-power on the GW-7472 after the update is completed.

Mode	Firmware Running	Flash Protection	Firmware Update	Configuration
Init	No	No	Yes	Allowed
Run	Yes	Yes	No	Not allowed

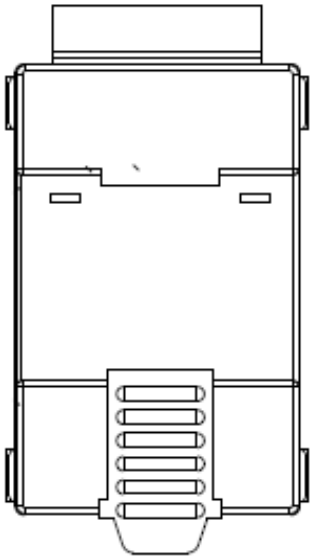
2.4 Dimensions

- GW-7472 dimensions:

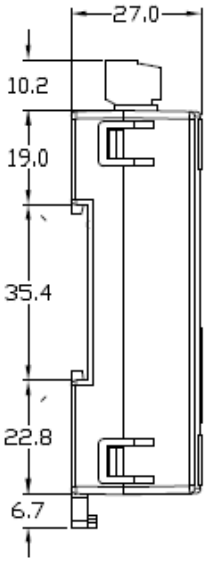
Unit: mm



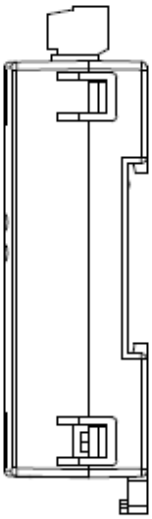
Front View



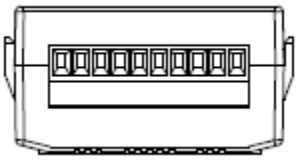
Back View



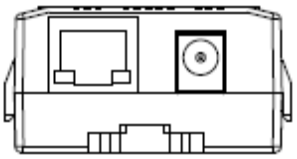
Left Side View



Right Side View



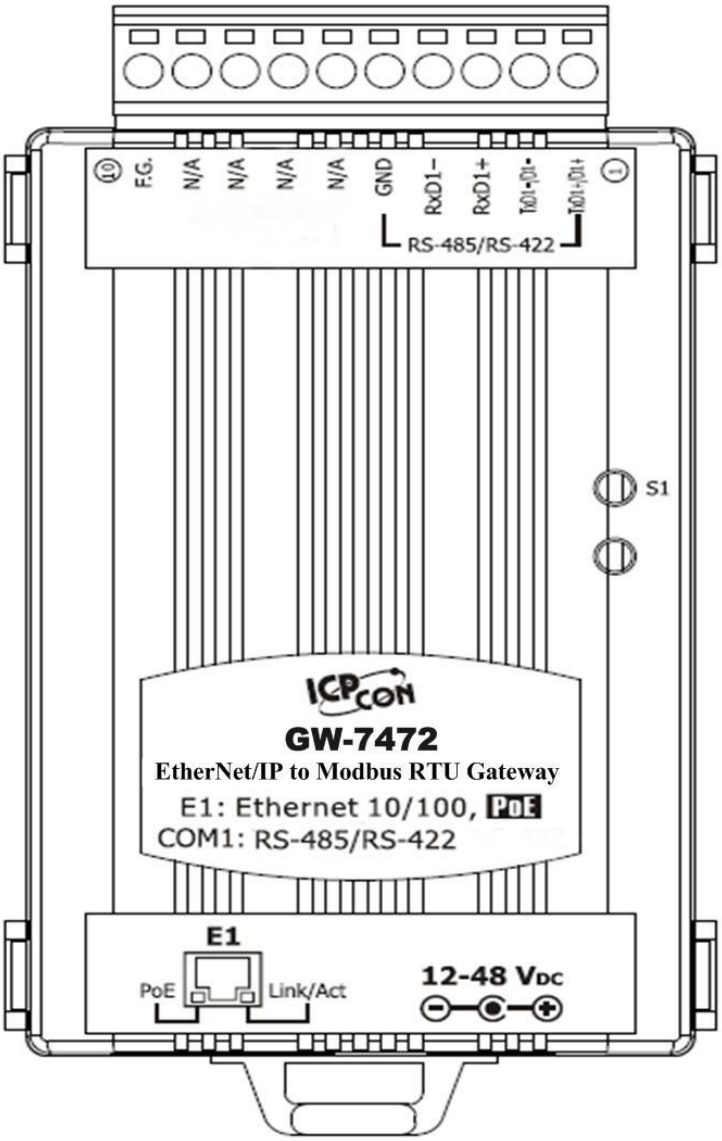
Top View

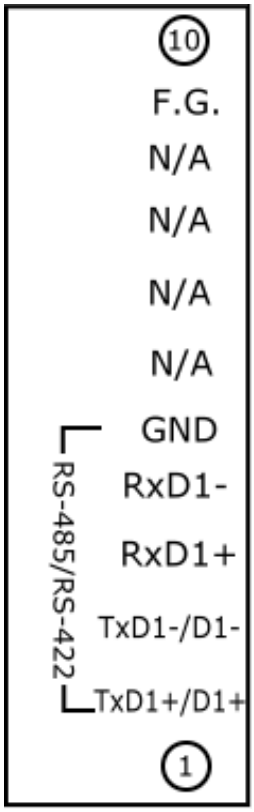


Bottom View

2.5 Pin Assignment

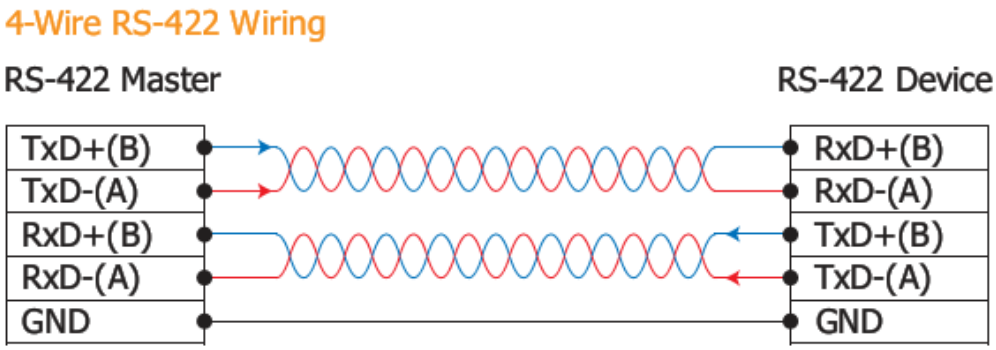
1-Port 2-Wire RS-485/ 4-Wire RS-422 Module



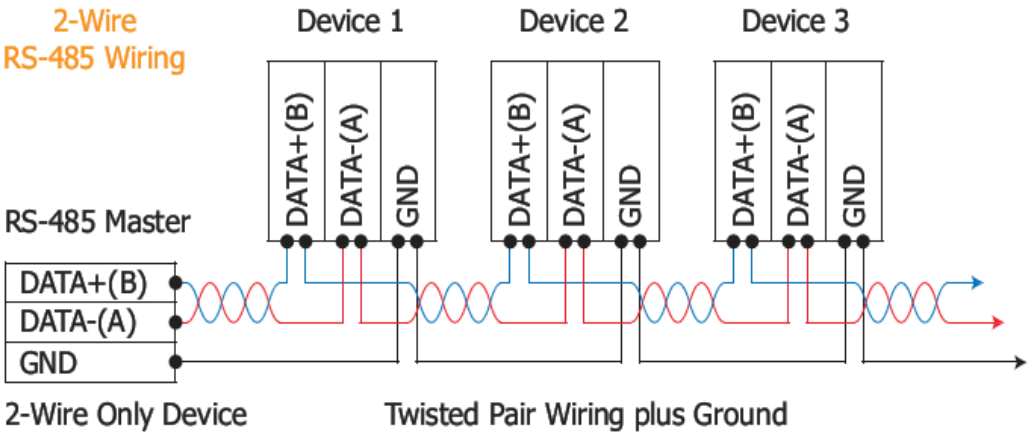


2.6 Wiring Note

2.6.1 RS-422 Wire Connections



2.6.2 RS-485 Wire Connections



Note!!

For non-isolated RS-422/485 ports, you should connect all signal grounds of RS-422/485 devices together. This reduces common-mode voltage between devices.

3. Setup and Test the GW-7472 module

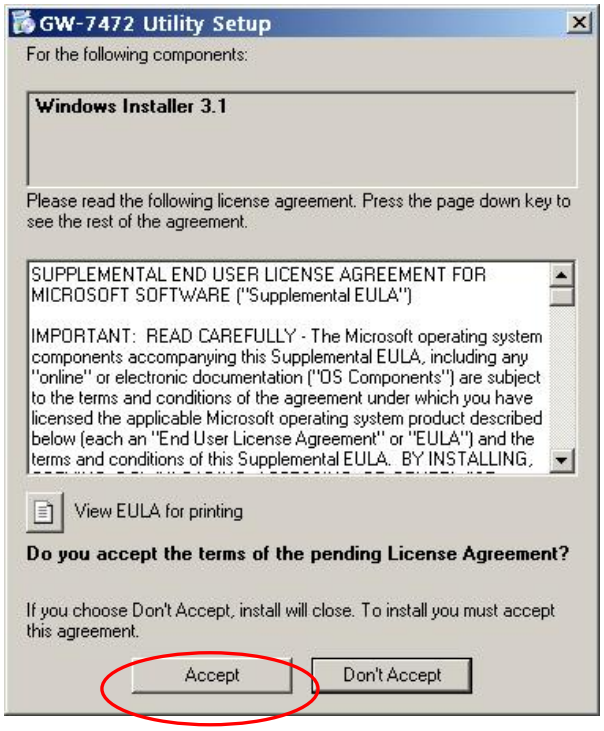
3.1 Install the GW-7472 Utility

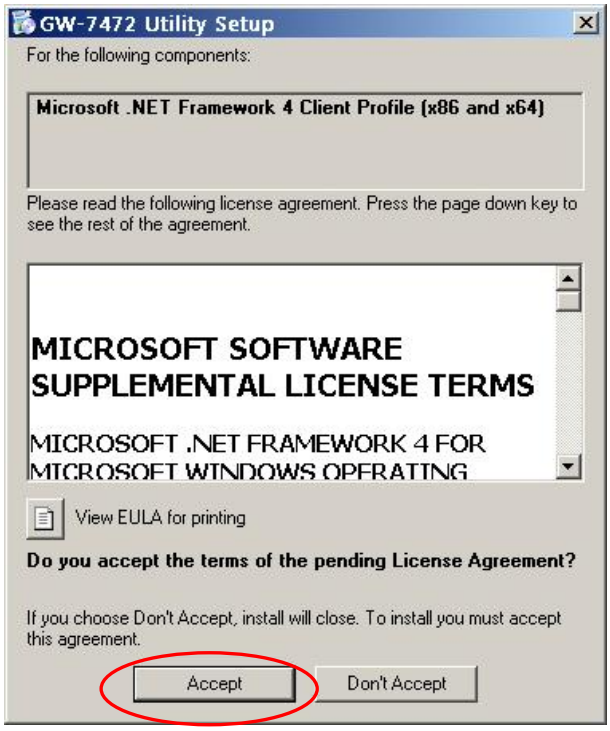
Step 1: Get the GW-7472 Utility

The software is located at:
Fieldbus_CD:\EtherNetIP\Gateway\GW-7472\Utility
http://ftp.icpdas.com/pub/cd/fieldbus_cd/ethernetip/gateway/gw-7472/utility/

Step 2: Install .NET Framework 4 component

The GW-7472 Utility tool requires the Windows Installer 3.1 and the .NET Framework 4 components. These components can be obtained from the web site.

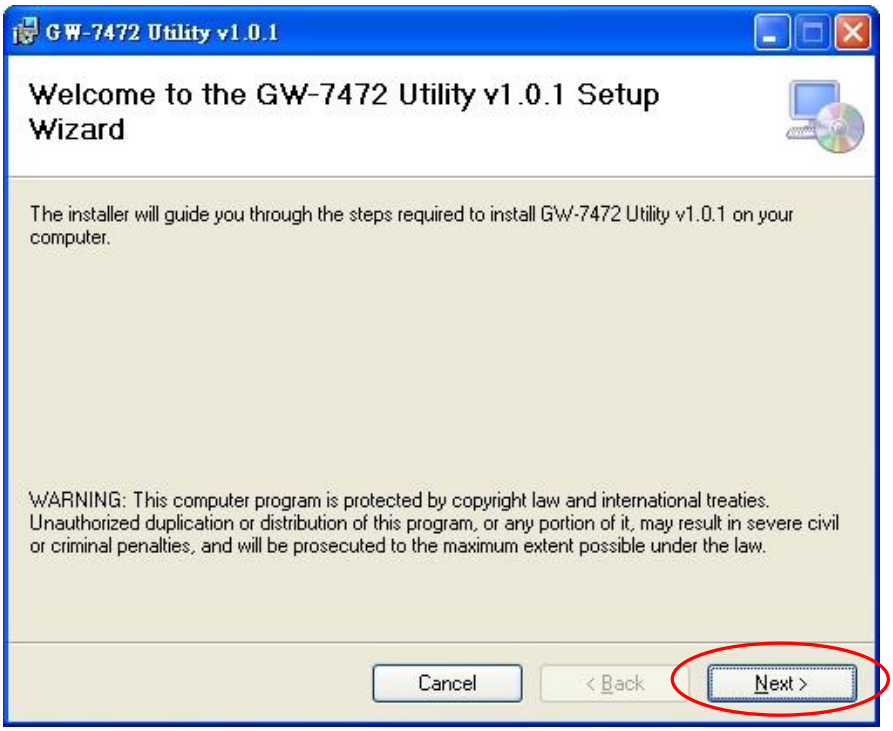




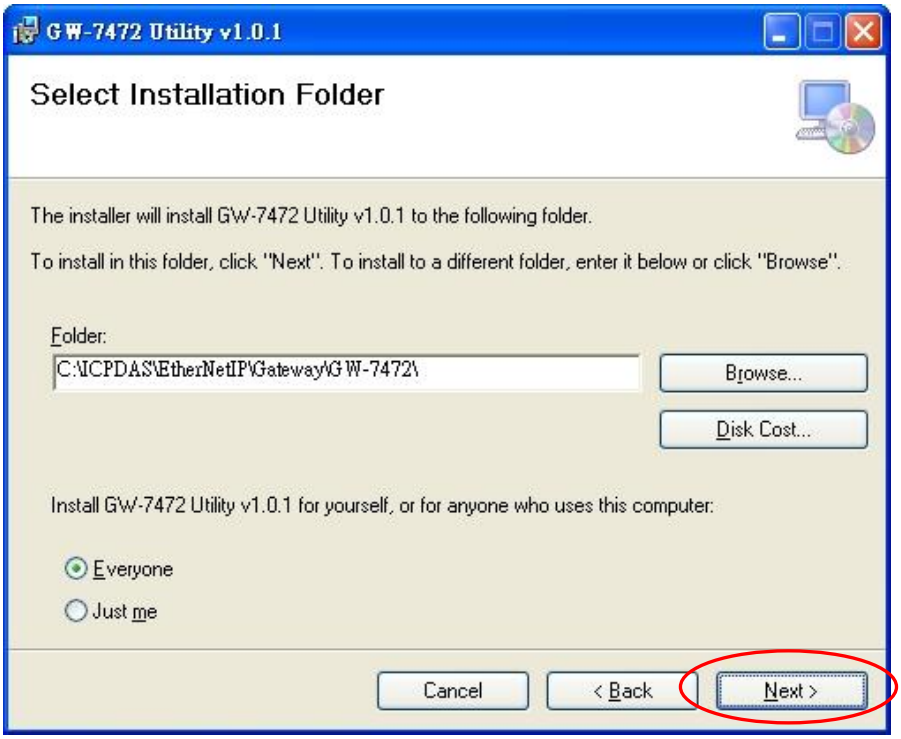
Step 3: Install Utility tool

After installing the .Net Framework components, please run the GW-7472 Utility setup file.

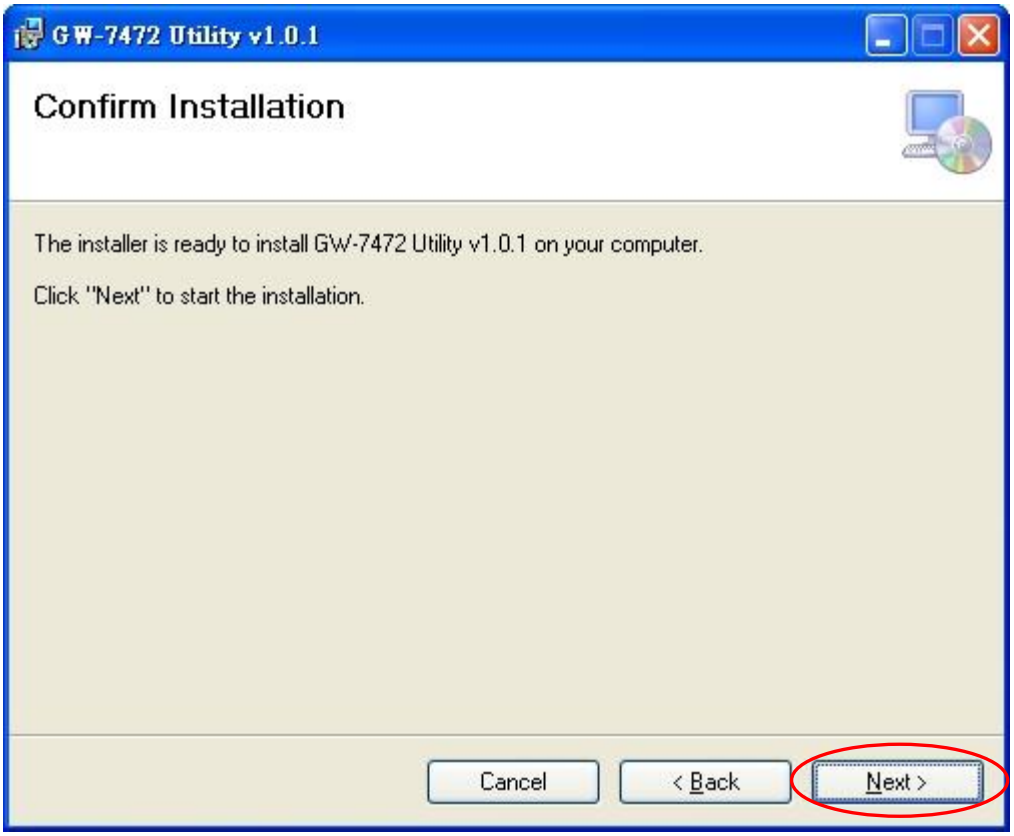
1. Click the “Next” button to continue.



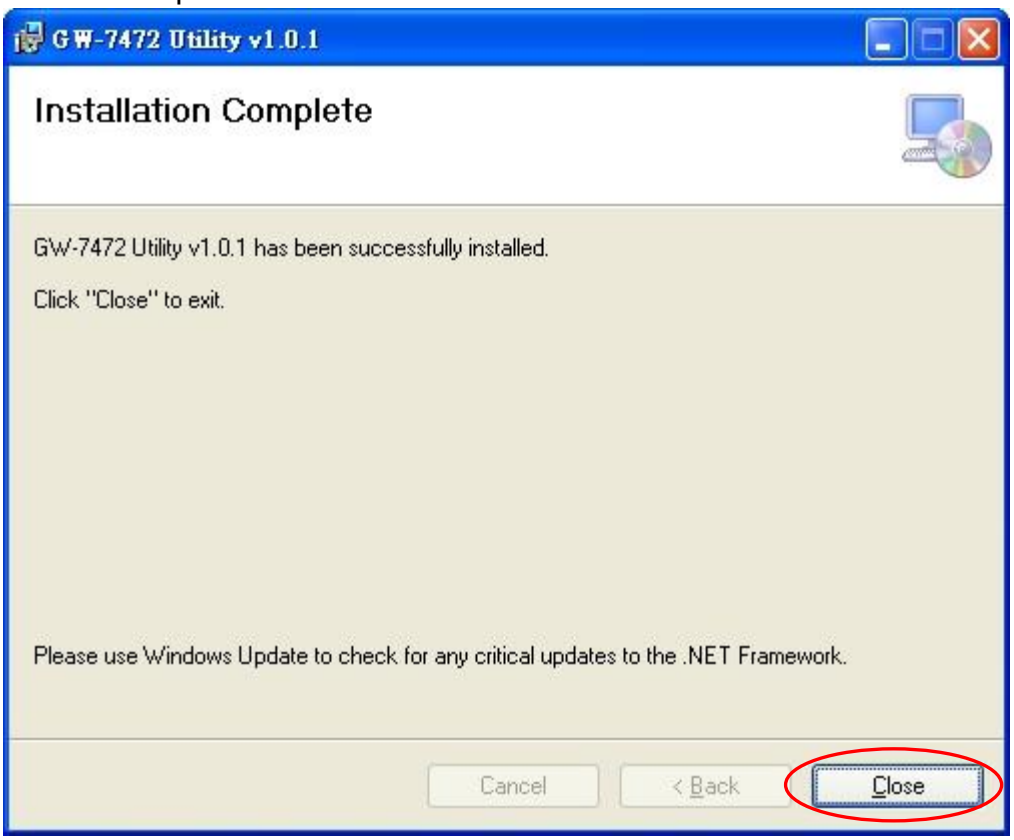
2. Select the installation path of the GW-7472 Utility and click the “Next” button.



3. Confirm the installation. Click the “Next” button to start the installation



4. Installation complete. Click the “Close” button to exit



3.2 Setting up the GW-7472 module

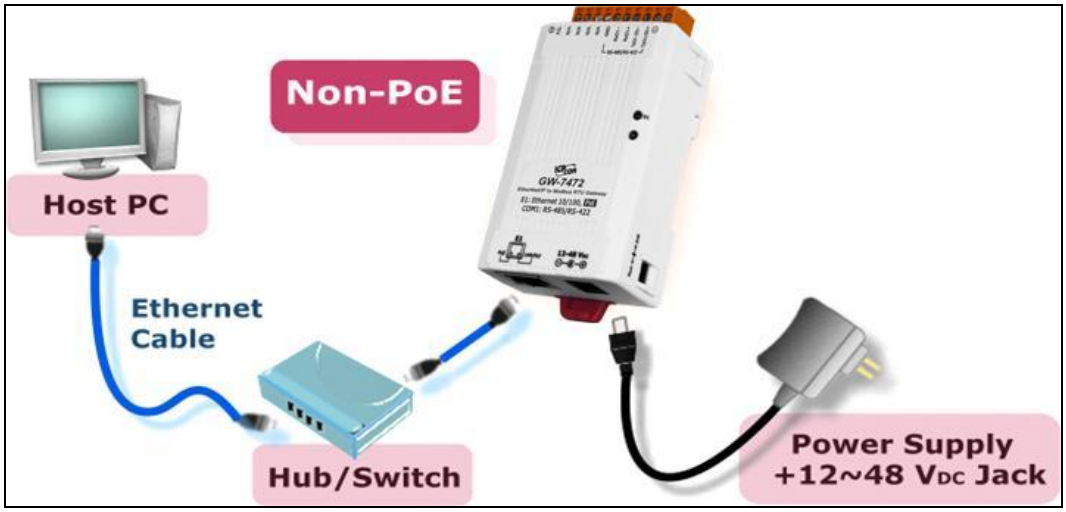
Step 1: Connect the power and host PC

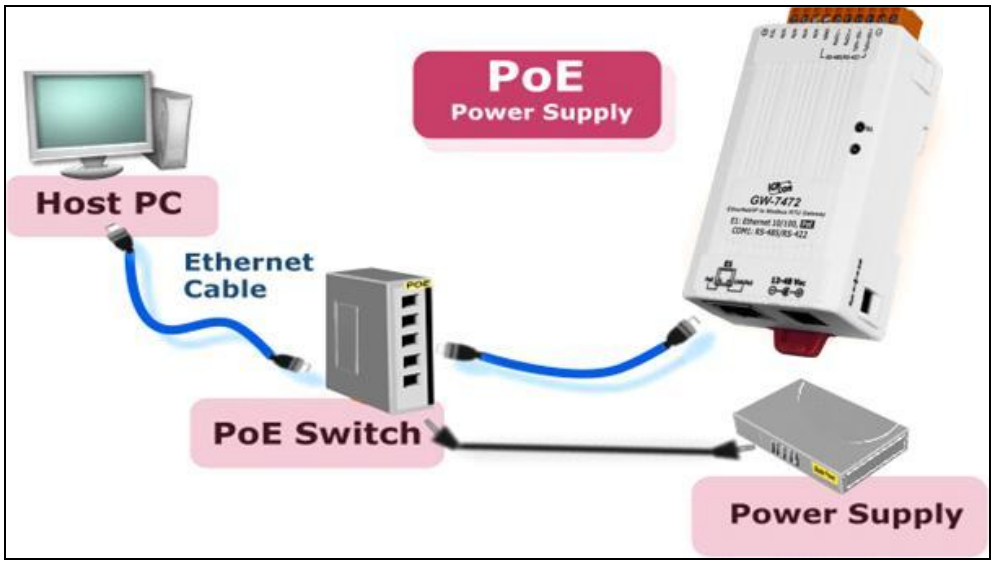
1. Make sure your PC is under the workable network configuration and environment.
2. First, disable or correctly configure the firewall of the Windows system and any anti-virus software. Or, the "**Configure**" function of the GW-7472 Utility may not work. (Contact your system administrator for more details about how to do this.)
3. Check Init/Run switch is on **Init** position.

4. In Init mode, the GW-7472 is forced to the network configuration as following table. Connect the GW-7472 with your computer at the same sub network or by using the same Ethernet switch. Then power the GW-7472 on. Afterwards, you can use the command "ping 192.168.255.1" in the Command Prompt window to test if the connection between the GW-7472 and your computer is OK.

Item	Settings (Init Mode)
IP	192.168.255.1
Gateway	192.168.0.1
Mask	255.255.0.0

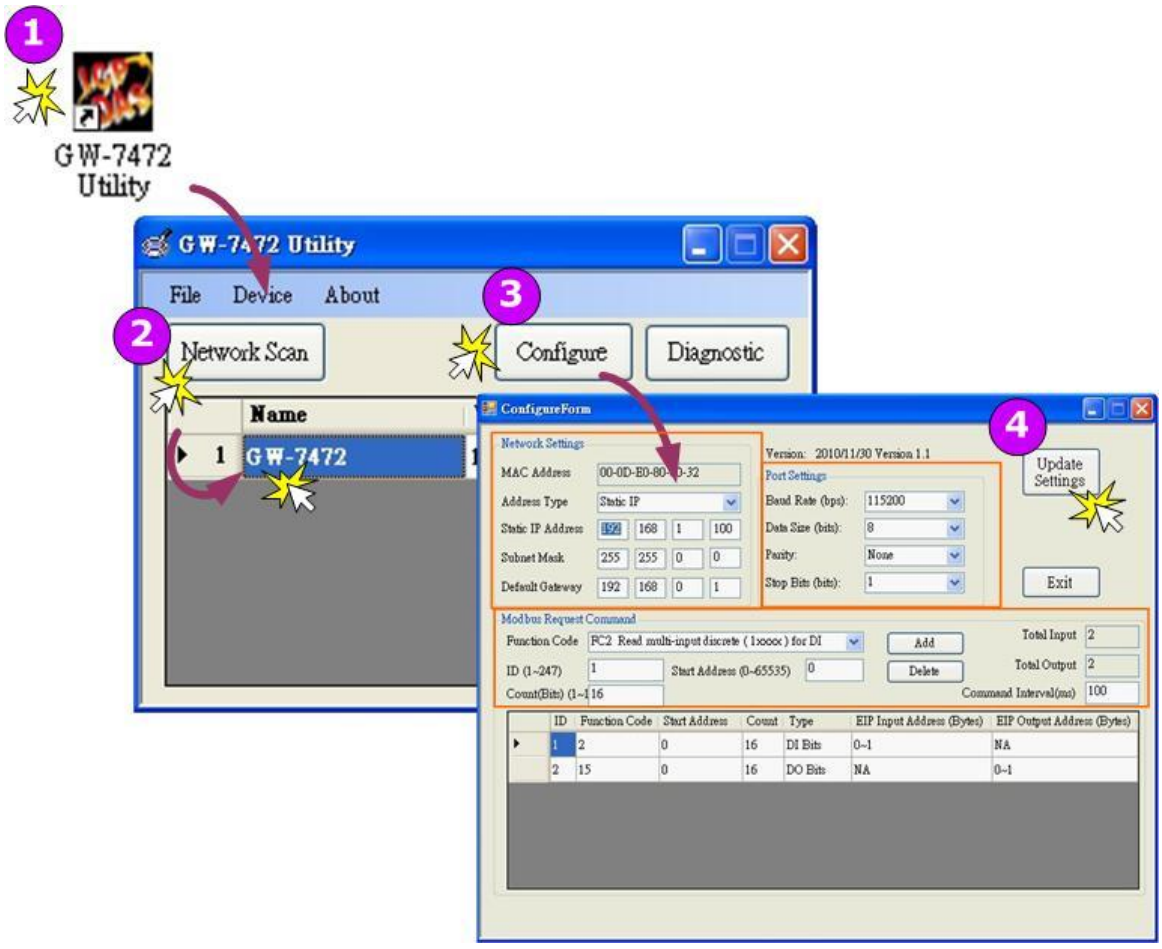
5. Make sure the System LED indicator is flashing.





Step 2: Search and configure the GW-7472

1. Double click the GW-7472 Utility shortcut on the desktop.
2. Click the "**Network Scan**" button to search your GW-7472.
3. Select the item of the GW-7472 and click the "**Configure**" button to open the configuration dialog.
4. After setting all the parameter of the GW-7472, click the "**Update Settings**" and "**Exit**" button to save and finish the configuration.



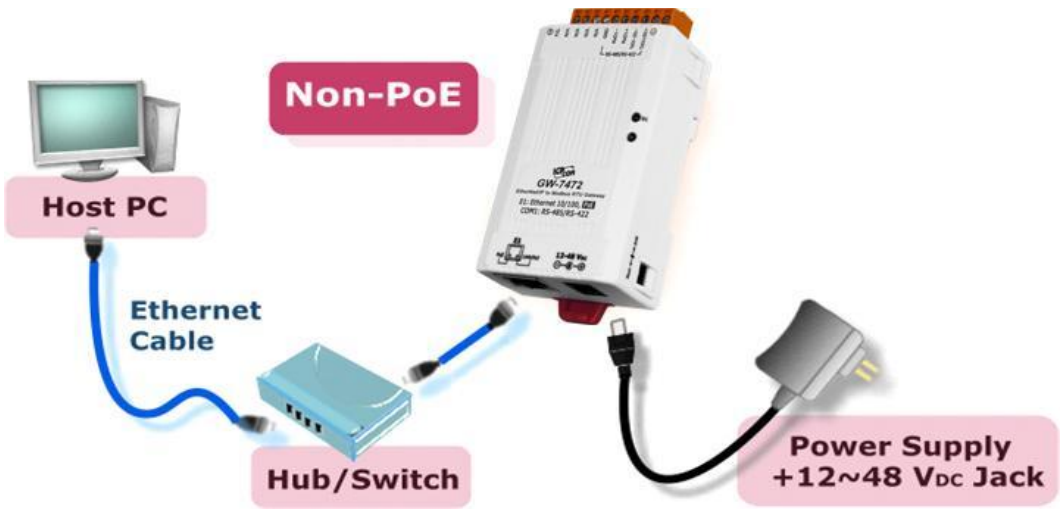
Please refer to the section "4.2 Module Configuration" for details

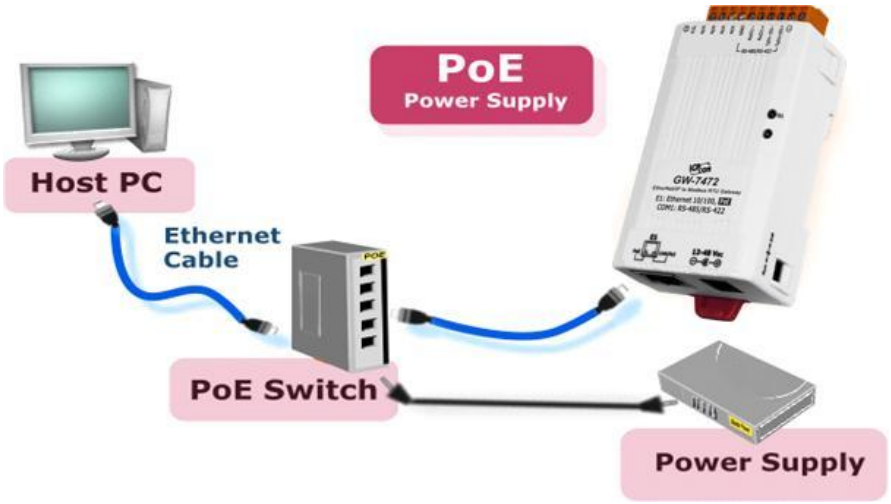
3.3 Testing the GW-7472 module

Step 1: Connect the power and host PC

1. Make sure your PC is under the workable network configuration and environment.
2. First, disable or correctly configure the firewall of the Windows system and any anti-virus software. Or, the "**Diagnostic**" function of the GW-7472 Utility may not work. (Contact your system administrator for more details about how to do this.)
3. Check Init/Run switch is on **Run** position.

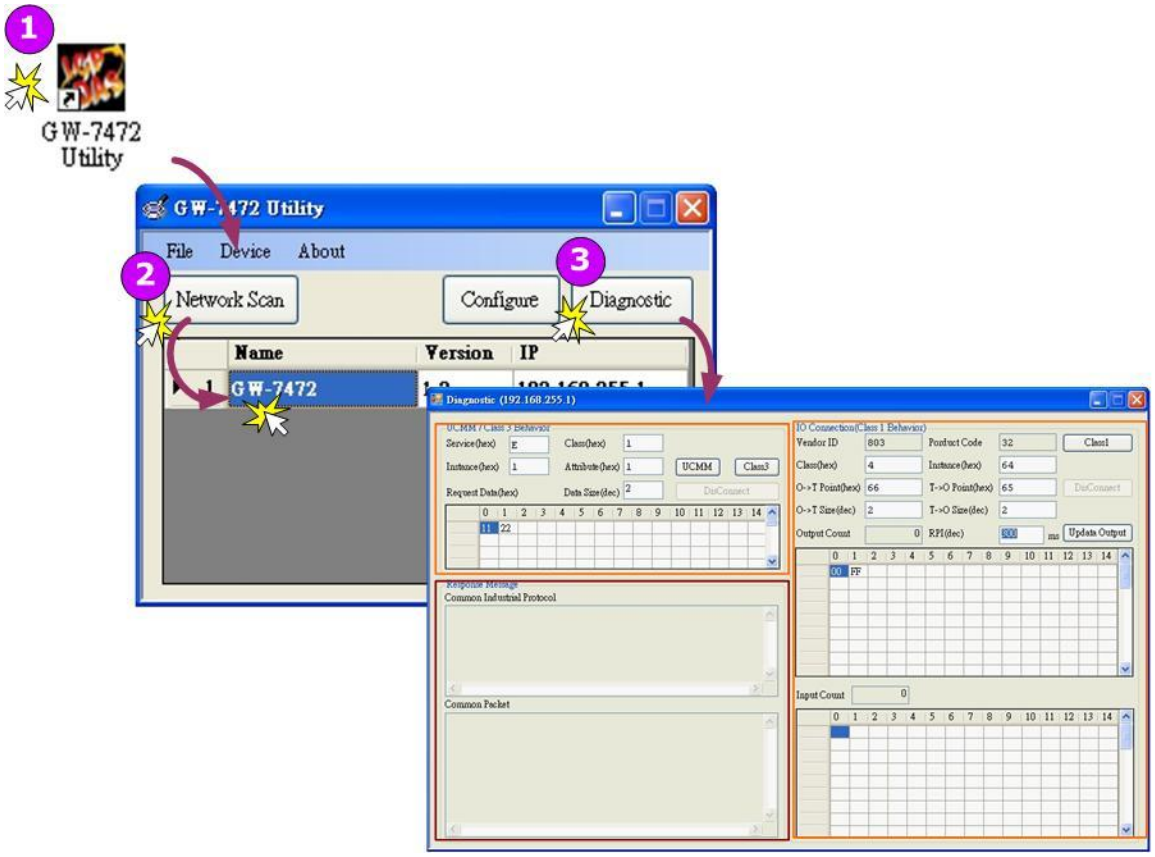
4. Connect the GW-7472 with your computer at the same sub network or by using the same Ethernet switch. Then power the GW-7472 on. Afterwards, you can use the command "ping" in the Command Prompt window to test if the connection between the GW-7472 and your computer is OK.
5. Make sure the System LED indicator is flashing.





Step 2: Search and test the GW-7472

1. Double click the GW-7472 Utility shortcut on the desktop.
2. Click the "**Network Scan**" button to search your GW-7472.
3. Select the item of the GW-7472 and click the "**Diagnostic**" button to open the diagnostic dialog.

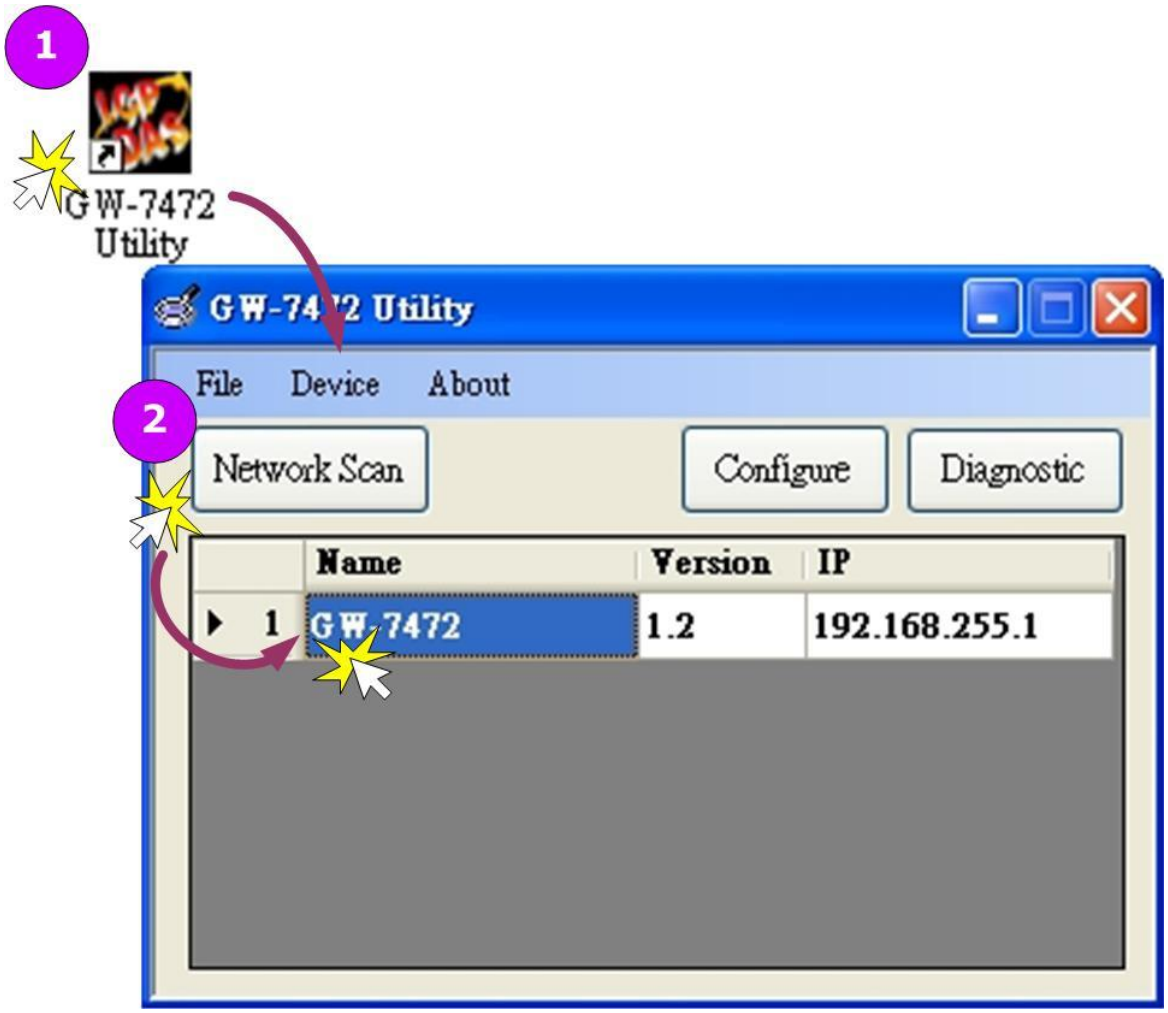


Please refer to the section "4.3 Module Diagnostic" for details

4. GW-7472 Utility Functionalities

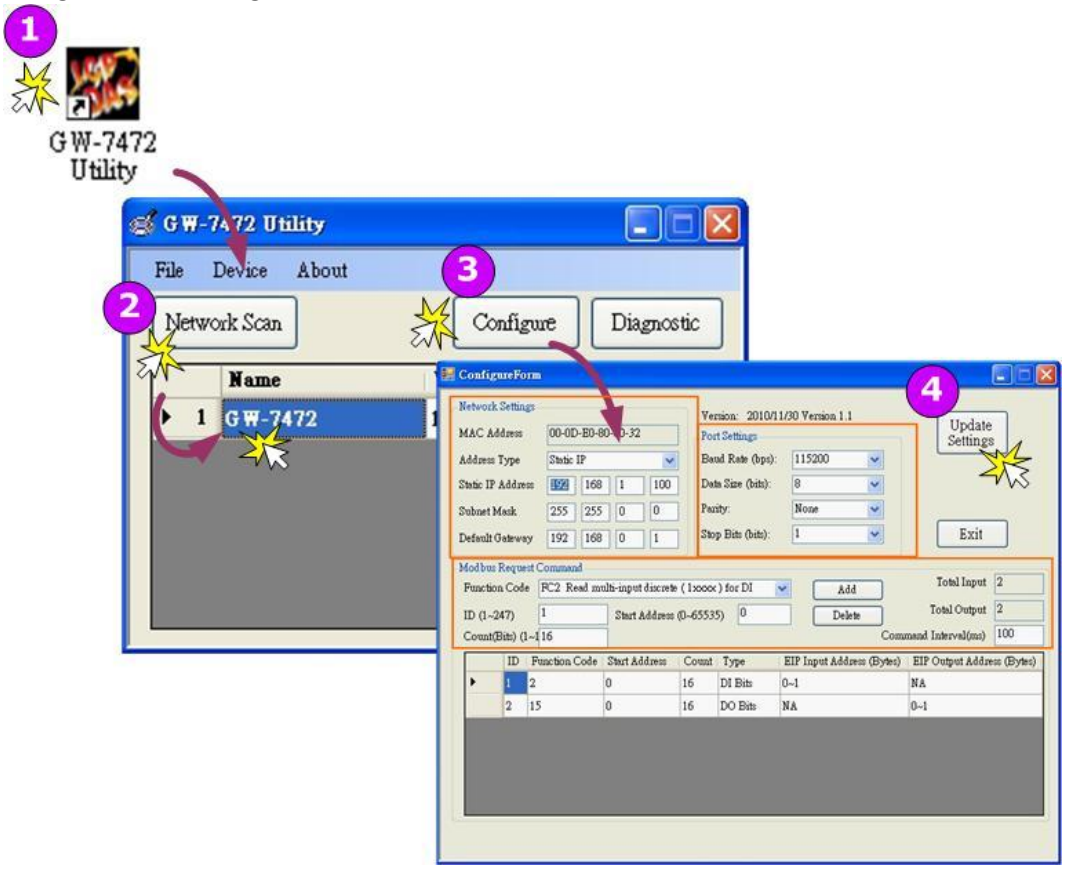
4.1 Network Scan

1. Double click the GW-7472 Utility shortcut on the desktop.
2. Click the "**Network Scan**" button to search your GW-7472. Afterwards, you can see all of the GW-7472 on the same network of your PC.



4.2 Module Configuration

1. Double click the GW-7472 Utility shortcut on the desktop.
2. Click the "**Network Scan**" button to search your GW-7472.
3. Select the item of the GW-7472 and click the "**Configure**" button to open the Configuration dialog.



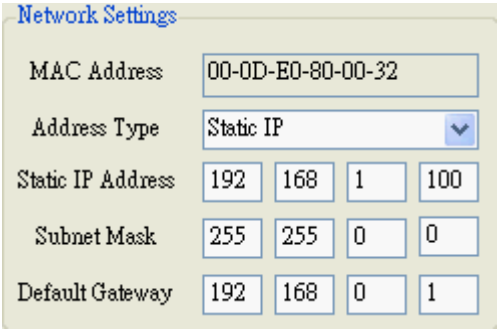
- Item Descriptions:

Item	Description
Network Settings	For configuration of the **Address Type**, **Static IP Address**, **Subnet Mask** and **Default Gateway** of the GW-7472 Please refer to section "**4.2.1 Network Settings**"
Port Settings	For configuration of the **Baud Rate**, **Data Sizes**, **Parity**, **Stop Bits**, of the RS-485/RS-422 port of the GW-7472 Please refer to section "**4.2.2 Serial Port Settings**"
Modbus Request Command	Modbus commands to communicate with the Modbus slaves Please refer to section "**4.2.3 Modbus Request Settings**"

Note!!

All settings will take effected after rebooting the system of the GW-7472 module

4.2.1 Network Settings



The **Address Type**, **Static IP Address**, **Subnet Mask** and **Default Gateway** items are the most important network configuration and should always match the LAN definition of your PC. Or, the connection between the GW-7472 and your PC may have problem. Contact your network administrator to obtain a proper network configuration for the GW-7472.

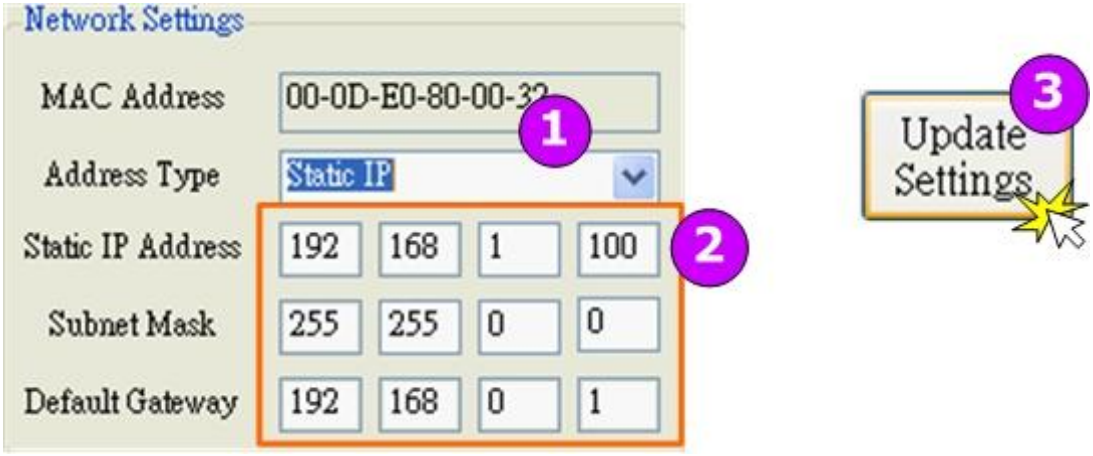
- Item Descriptions:

Item	Description
Address Type	**Static IP:** If you don't have a DHCP server in your network, configure the network settings manually. Please refer to the section "**4.2.1.1 Manually Configuration**"
	DHCP: Dynamic Host Configuration Protocol (DHCP) is a network application protocol that automatically assigns IP address to devices by the DHCP server. If there is no DHCP server in the network, the static IP must be used. Please refer to the section "**4.2.1.2 Dynamic Configuration**"
Static IP Address	Each GW-7472 on the network must have a unique IP address. This field is used to assign an IP address for the GW-7472.
Subnet Mask	The subnet mask defines which IP addresses of the network device are in the same sub-network.
Default Gateway	A gateway (or router) is a device that is used to build a connection between two sub-networks.
MAC Address	The MAC address of the GW-7472.
Update Settings	Click this button to save the new settings to the GW-7472.

4.2.1.1 Manually Configuration

In manually configuration, you have to assign all the network settings by yourself. The steps are shown below:

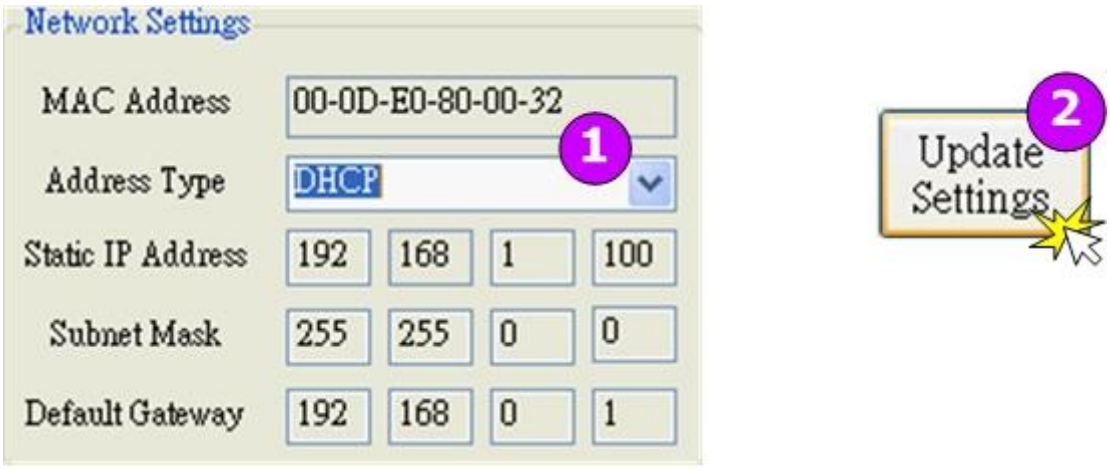
Step1: Select the "**Static IP**".
Step2: Enter the **network settings**.
Step3: Click the "**Update Settings**" button to finish the configuration.



4.2.1.2 Dynamic Configuration

The procedure of the dynamic configuration is very easy. If you have a DHCP server, network address can be configured dynamically by the following steps:

Step1: Select the "**DHCP**".
Step2: Click the "**Update Settings**" button to finish the configuration.



4.2.2 Serial Port Settings

There four parameters in the serial port configuration dialog.

- Item Descriptions:

Item	Description	Default
Baud Rate (bps)	Set bard rate of the RS-485/422 ports.	115200
Data Size (bits)	Set data size of the RS-485/422 ports.	8
Parity	Set parity of the RS-485/422 ports.	None
Stop Bits (bits)	Set stop bits of the RS-485/422 ports.	1

Step1: Enter the **port settings**.

Step2: Click the "**Update Settings**" button to finish the configuration

4.2.3 Modbus Request Settings

The settings for the Modbus commands are provided as the following list:

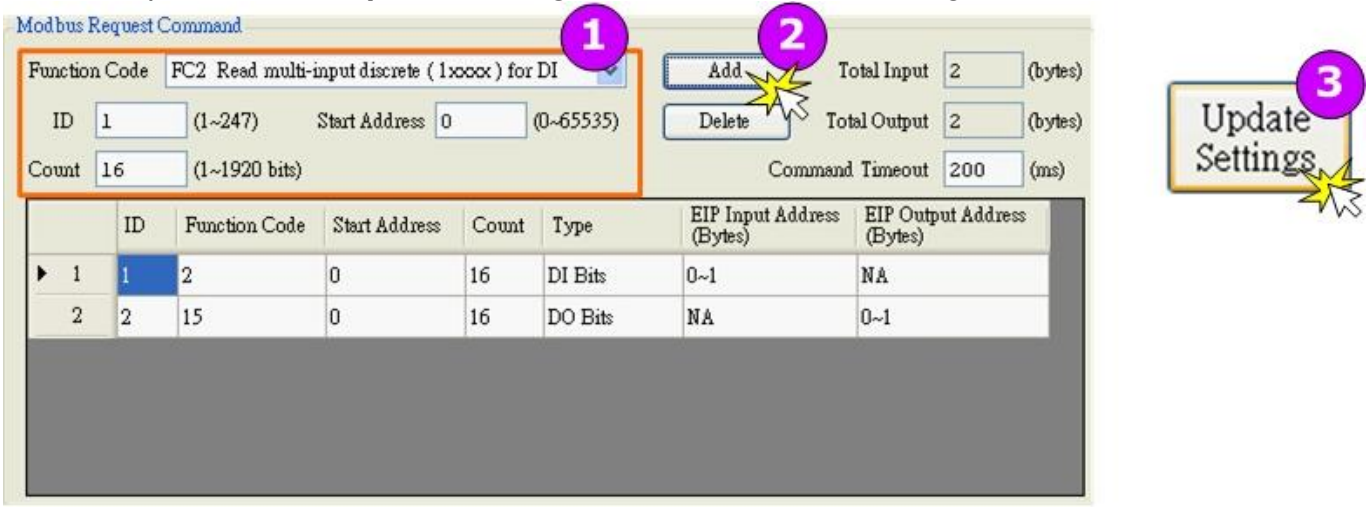
- Item Descriptions:

Item	Description
Function Code	Supported Modbus Function codes are 01_{hex}, 02_{hex}, 03_{hex}, 04_{hex}, $0F_{hex}$ and 10_{hex}
ID	The Modbus slave device ID specifies the address of the device on the RS-485/422 network. This ID can be 1 ~ 247.
Start Address	The start address of the input/output registers stored in the Modbus slaves. This address can be 0 ~ 65535.
Count Bits/Words	Number of register data to be accessed from the Modbus slave
Total Input	Show how many bytes have been mapped in of the EtherNet/IP input registers
Total Output	Show how many bytes have been mapped in of the EtherNet/IP output registers
EIP Input Address (Bytes)	The mapping address in the EtherNet/IP input register.
EIP Output Address (Bytes)	The mapping address in the EtherNet/IP output register.
Command Interval (milliseconds)	Interval value of the Modbus RTU commands. If the command is replied by the Modbus slave immediately, the GW-7472 still waits until the time interval passes. Set range value: 10 ~ 30000 (milliseconds); Default: 200 ms

Step1: Enter the **Modbus Request commands**.

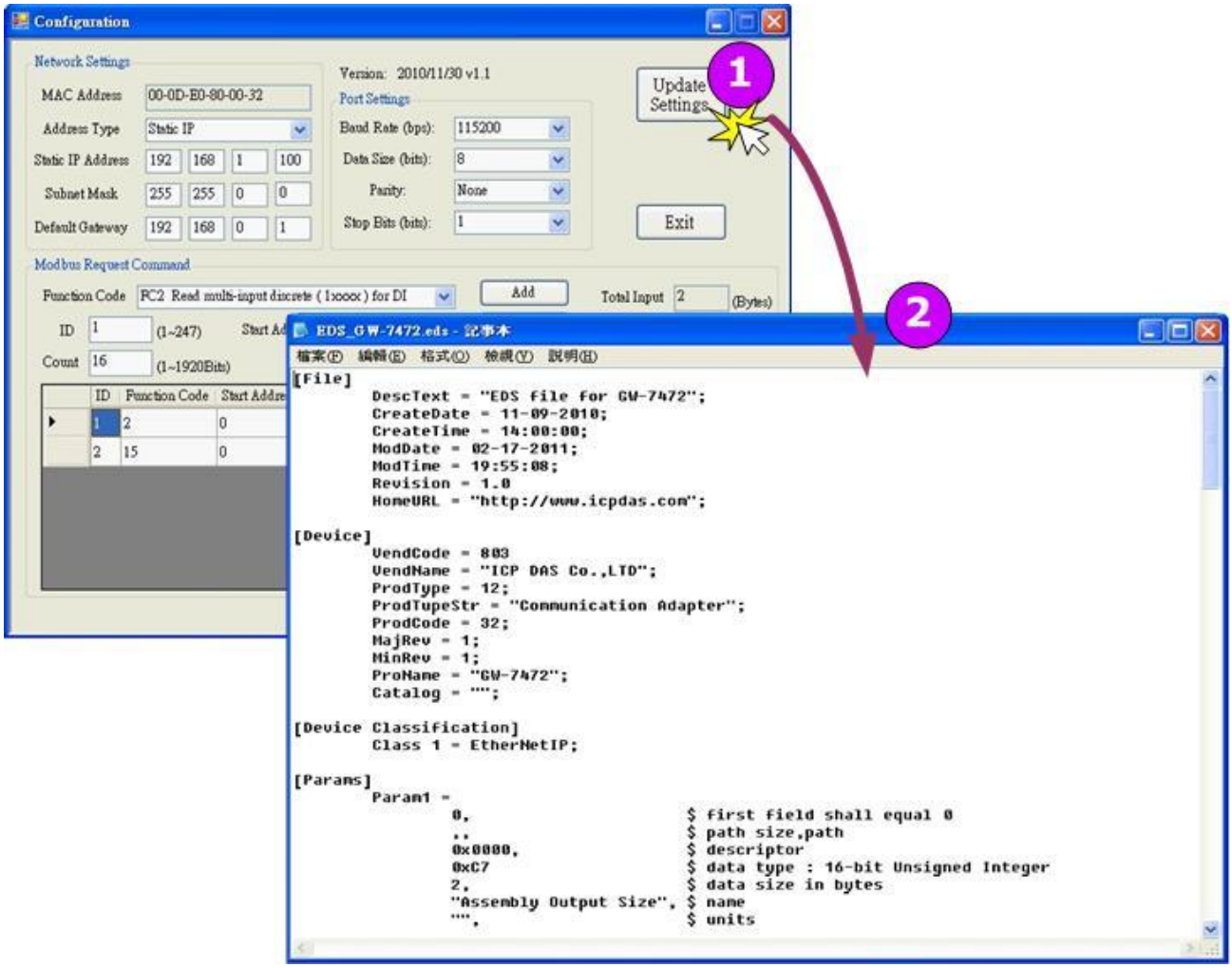
Step2: Click the "**Add**", "**Delete**" buttons to add and remove the Modbus commands.

Step2: Click the "**Update Settings**" button to finish the configuration

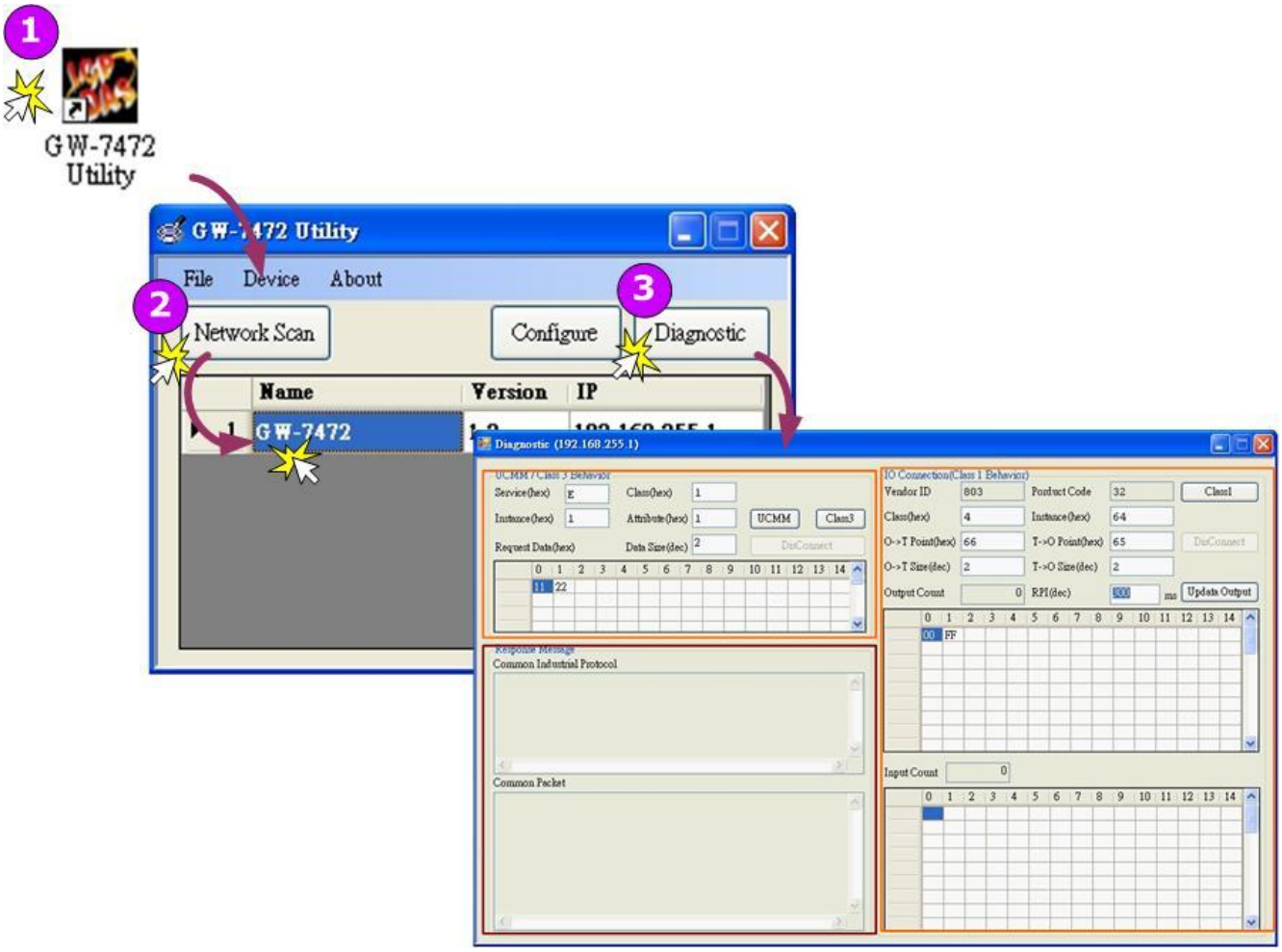


4.2.4 Electric Data Sheet

The Electric Data Sheet (EDS) is a kind of file recorded all of the necessary information which is useful while an EtherNet/IP scanner want to access an EtherNet/IP adapter. It is an important bridge between the variety EtherNet/IP adapters and the configuration tool of the EtherNet/IP scanner. Through the EDS file, the configuration tool from 3rd parity is able to easily know that which parameters can be accessed or altered. After setting the parameters of the GW-7472, an EDS file ("EDS_GW-7472.eds") will be created in the same folder of the Utility tool.



4.3 Module Diagnostic



- Item Descriptions:

Item	Description
UCMM/Forward Open Class 3 Behavior	Send UCMM packets or use the Forward_Open service to build the CIP class 3 connection to communicate with the GW-7472. Please refer to section "**4.3.1 UCMM/Forward Open Class 3 Behavior**"
Forward Open Class1 Behavior	Use the Forward_Open service to build the CIP class 1 connection to communicate with the GW-7472. Please refer to section "**4.3.2 Forward Open Class 1 Behavior**"
Response Message	EtherNet/IP packets responded from the GW-7472.

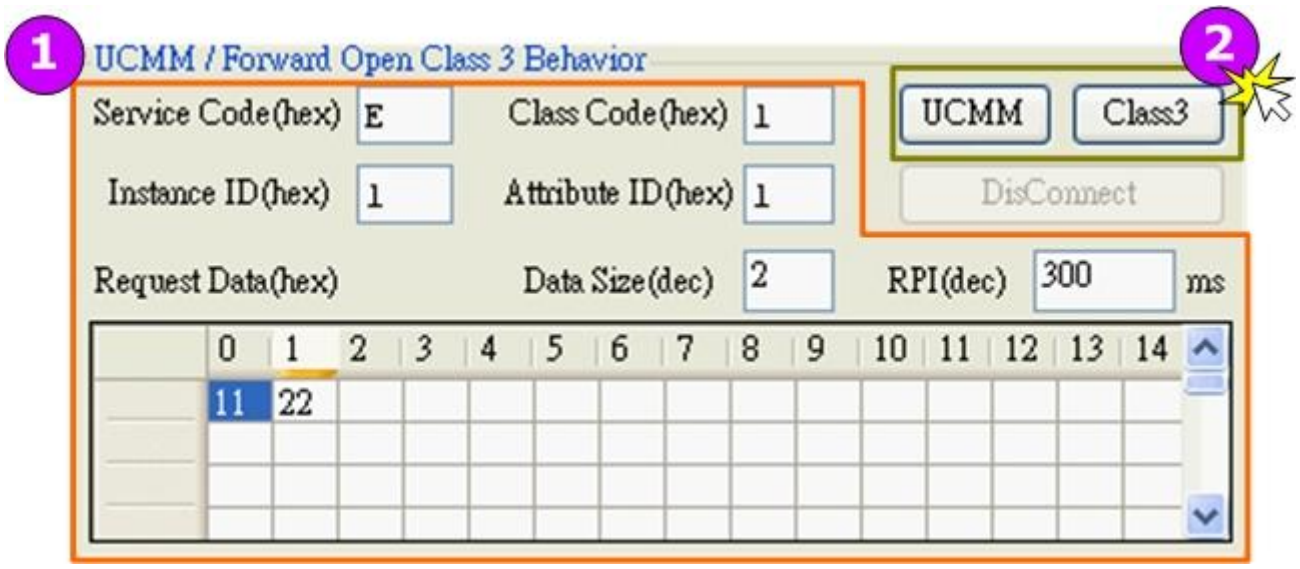
4.3.1 UCMM/Forward Open Class 3 Behavior

This field is applied to send UCMM (Unconnected Message Manager) packages or the Forward Open service to build the CIP class 3 connection. Both of these two methods can be used to communicate with the GW-7472.

Step1: Enter the **Service Code**, **Class Code**, **Instance ID**, **Attribute ID**, **Requested Data size**, **Request Data**, and **Request packet interval** parameters.

Step2: Click the "**UCMM**" or "**Class3**" buttons to communicate with the GW-7472.

Step3: Click the "**DisConnect**" button to stop to communicate with the GW-7472.



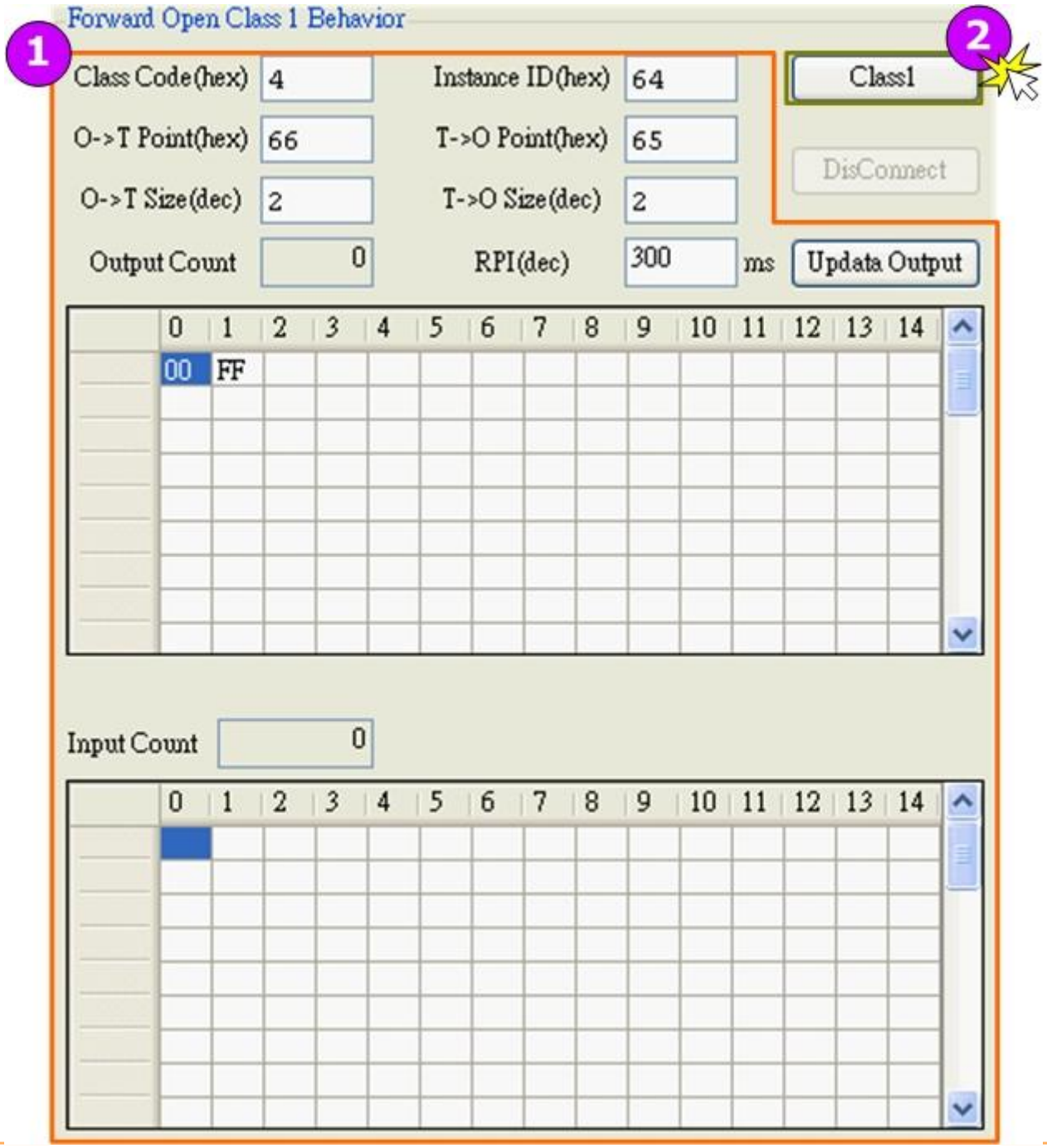
4.3.2 Forward Open Class 1 Behavior

Use this field to apply the Forward Open service to build the CIP class 1 connection to communicate with the GW-7472.

Step1: Enter the **Class Code**, **Instance ID**, **O->T Point**, **O->T Point**, **O->T Size**, **T->O Size**, and **RPI** parameters.

Step2: Click the "**Class1**" button to communicate with the GW-7472.

Step3: Click the "**DisConnect**" button to stop to communicate with the GW-7472.



4.4 Firmware Update

The GW-7472 supports firmware update through the Ethernet network with the BOOTP/TFTP protocol. Generally, the firmware is not necessary to update when it works well. If there are some bugs in the firmware of your GW-7472 or you need new functions which don't support in your GW-7472, the firmware update is necessary. If the firmware update procedure is broken unfortunately, please try it again.

Before updating the firmware, you have to set the "Init Switch" to "Init" position and then re-power on the GW-7472. Since the flash becomes writable, we can update the firmware through the Ethernet network.

Mode	Firmware Running	Flash Protection	Firmware Update	Configuration
Init	No	No	Yes	Factory
Run	Yes	Yes	No	User-Defined

Note:

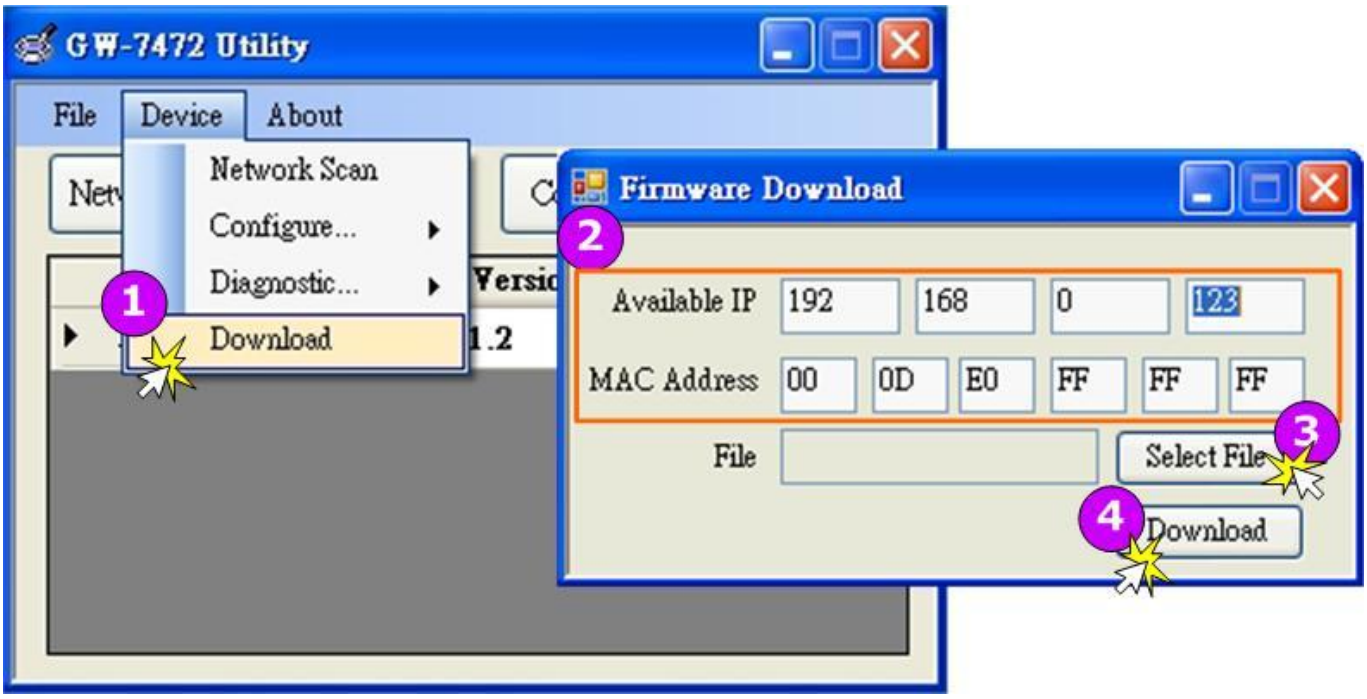
1. **Well configure the network settings of your PC. Or the update procedures through the Ethernet network may not work correctly.**
2. **The program (TFTP server) may not run correctly if there is another TFTP server running on the same PC.**
3. **The BOOTP and TFTP protocols use the Ethernet UDP port 67, 68 and 69. Please confirm that the firewall of the Windows system or anti-virus software can pass these UDP ports.**

Step1: Click the “**Download**” item to open the “Firmware Download” dialog.

Step2: Enter the MAC address of the GW-7472 and an available IP address which will be temporally assigned to the GW-7472 via the BOOTP protocol. After finishing the firmware update, this IP address is useless.

Step3: Select the firmware which will be updated.

Step4: Click the “**Download**” button to start the update procedure.



Available IP:

This parameter is an available IP address on the Ethernet network. During the update procedure, the GW-7472 will use this IP address. You can also assign the IP address which is used in the run mode of the GW-7472. Contact your network administrator for more information about an available IP address.

MAC Address:

This parameter is the MAC address of the GW-7472. You can get it from the Utility tool. Please refer to section “**4.2.1 Network Settings**”

Select File:

The folder path of the new firmware can’t include the character “ “(the space character). Or the update procedure may be broken.

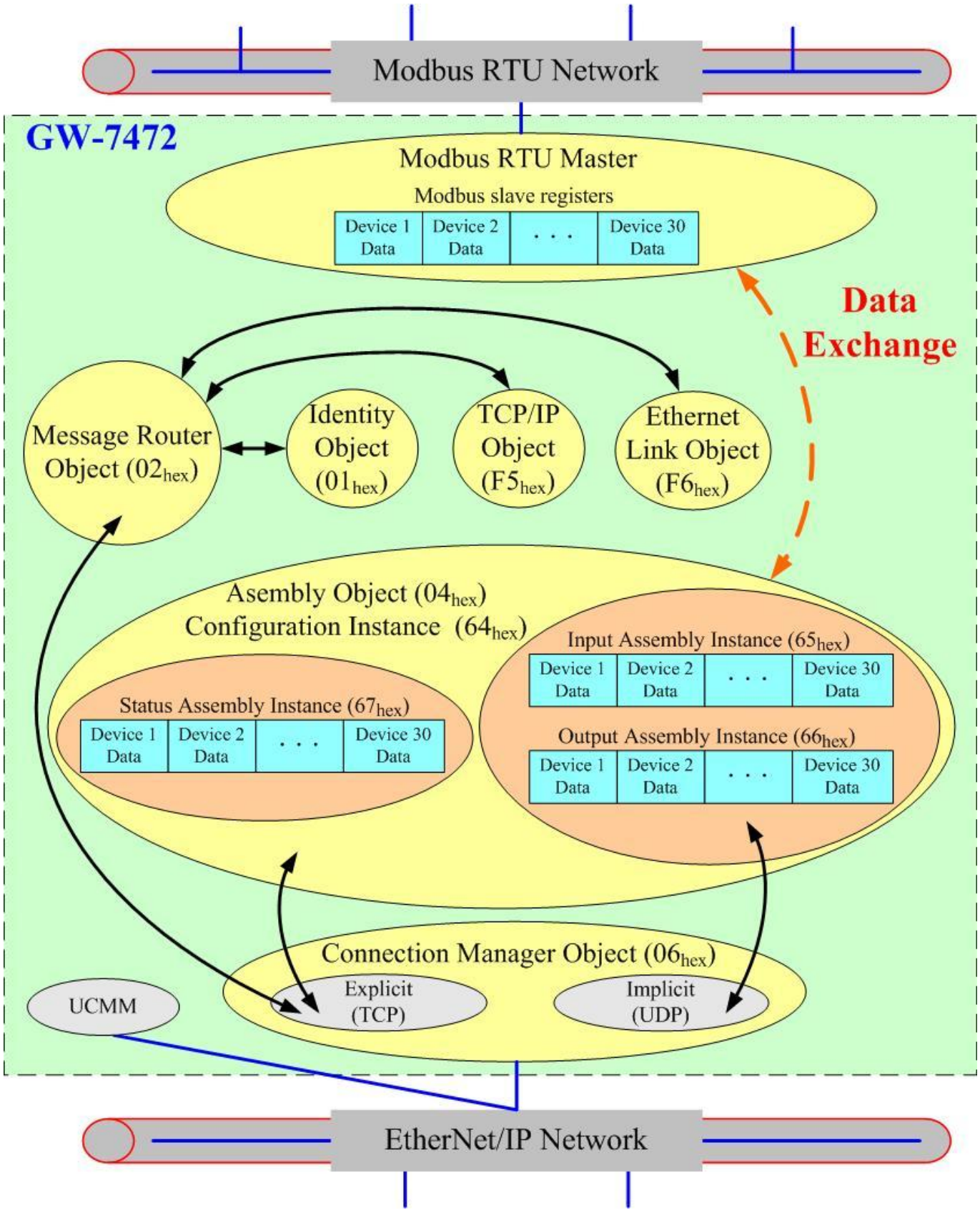
5. R/W Modbus RTU devices from EtherNet/IP

Since the GW-7472 provides the functions of an EtherNet/IP adapter and Modbus RTU master, there are some mechanisms for data-exchanging between EtherNet/IP objects and the Modbus RTU registers. This section describes the EtherNet/IP Object Model of the GW-7472 and how to read/write the GW-7472 EtherNet/IP object data mapping to the registers of Modbus slaves by using the EtherNet/IP Explicit and Implicit Message.

5.1 Object Model

The Object Model for the GW-7472 is shown in the following figure. Inside the GW-7472, there are one Modbus RTU master and an EtherNet/IP adapter. When booting up, the GW-7472 scans all of the input registers of Modbus RTU slaves and updates all of the output registers of Modbus RTU slaves. At the same time, the Modbus RTU master exchanges the input data and Modbus connection conditions with the objects of the EtherNet/IP adapter.

The EtherNet/IP adapter of the GW-7472 provides six kinds of objects. Each object has its characteristic, service and instances. The Connection Manager Object is applied for building a connection before using the Explicit Messages and the Implicit Messages. The Message Router Object is used to route the message to other objects of the EtherNet/IP adapter. The Assembly Object, Identity Object, TCP/IP Object, and Ethernet Link Object are used to record the I/O information, device information, TCP/IP configuration, Ethernet link-specific status information respectively. After receiving an EtherNet/IP message, the GW-7472 will distinguish what the message type it is. The Explicit Message can direct access the Assembly Object or access other objects via the Message Router Object. The Implicit Message can only access I/O data of the Assembly Object. The UCMM Message is used to access all of the objects without building a connection. When the UCMM Message is got by the GW-7472, the message is passed to the Message Router Object for routing. When the EtherNet/IP scanner communicates with the EtherNet/IP adapter of the GW-7472, the GW-7472 replies the corresponding information. At the same time, the EtherNet/IP adapter exchanges the output data with the Modbus master.


Modbus RTU Network
GW-7472
Modbus RTU Master
Modbus slave registers
Device 1 Data
Device 2 Data
. . .
Device 30 Data
Data Exchange
Message Router Object (02hex)
Identity Object (01hex)
TCP/IP Object (F5hex)
Ethernet Link Object (F6hex)
Asembly Object (04hex)
Configuration Instance (64hex)
Input Assembly Instance (65hex)
Status Assembly Instance (67hex)
Output Assembly Instance (66hex)
Connection Manager Object (06hex)
UCMM
Explicit (TCP)
Implicit (UDP)
EtherNet/IP Network

5.2 Explicit Message

Explicit Messages are applied for accessing all of the objects in the object model. The specific instances and attributes for each Object Class are presented in “Appendix A: EtherNet/IP Object Model”. Before using Explicit Messages, you must use the Forward Open service of the Connection Manager Object to build a connection between the EtherNet/IP scanner and the GW-7472. Afterwards, the Explicit Message can be used.

5.4 Implicit Message

Implicit Messages are applied only for accessing the Input Instance 65_{hex} and Output Instance 66_{hex} of the Assembly Object in the object model. Before using Implicit Messages, you must use the Forward Open service of the Connection Manager Object to build a connection between the EtherNet/IP scanner and the GW-7472. Afterwards, the Implicit Message can be used.

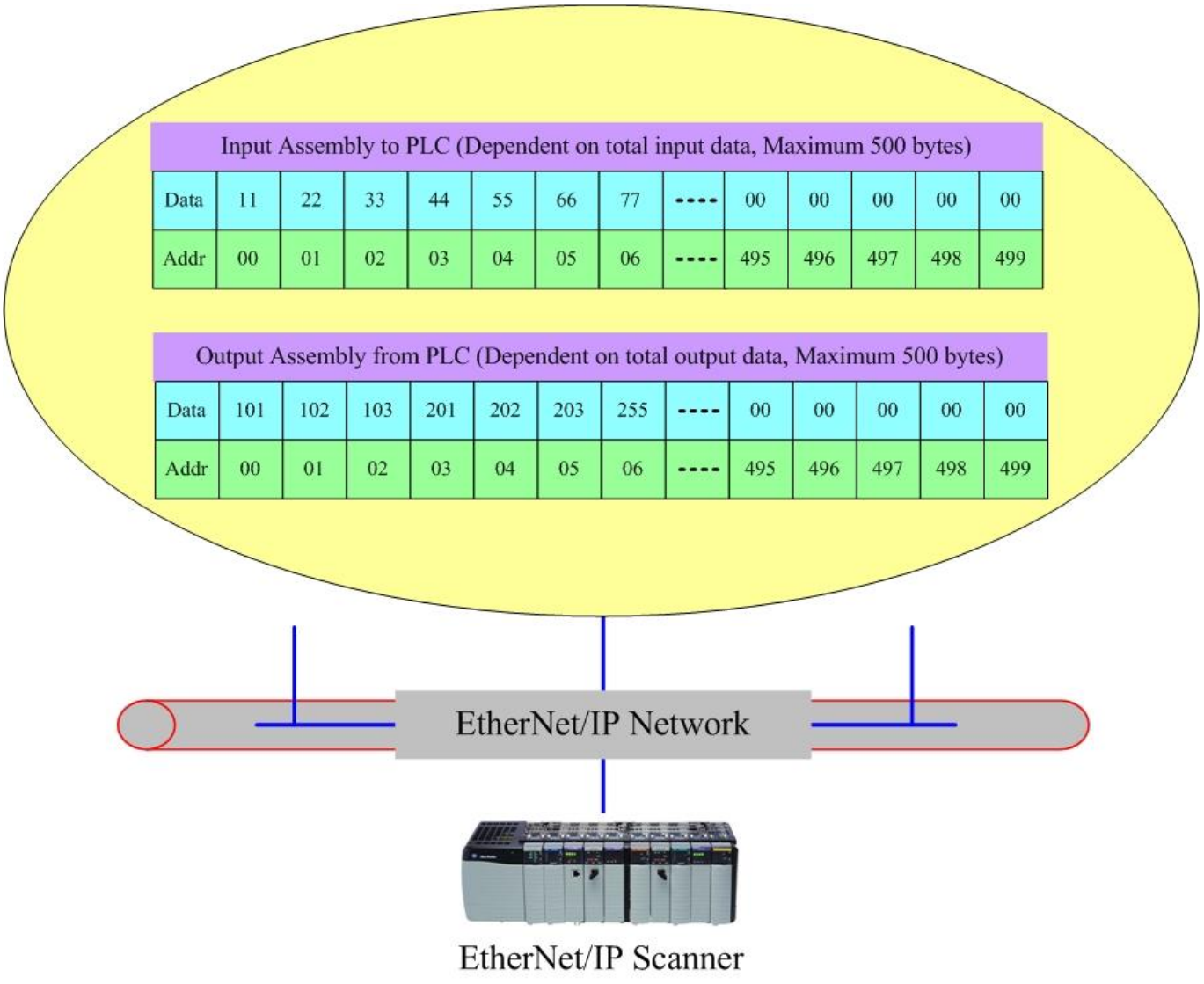
5.5 UCMM

The UCMM are also applied for accessing all of the objects in the object model. The main feature of the UCMM is that you can send the UCMM without building a connection. It is a simple method for EtherNet/IP to get the information of all objects. However, because of using UCMM without building a connection, the reliability of the message transmission is worse than the Explicit Message.

5.6 Assembly Object

The GW-7472 supports one input instance, one output instance and one command status instance in Assembly Object. Each of these instances is mapping to the register data of the Modbus slaves. After you use the Utility to configure the GW-7472, the mapping information between the registers of the Modbus slaves and the instances of the Assembly Object is created by following the configuration order of the Modbus commands defined by the Utility. A

GW-7472 allows setting maximum 30 Modbus commands to get the register data of the Modbus slaves. The unit of the input instance and output instance is BYTE. Therefore, no matter the data format is Coil (1 bit) or WORD, all of the register data of the Modbus slaves will be assigned to the instances of the Assembly Object by using BYTE format. While creating a mapping table, the data in the same Modbus command will be put together and be mapping to some section of the instance by using integral number of bytes. The input register data of the first Modbus command defined by the Utility are mapping to the most front end of the input instance. The input register data of the following Modbus command are mapping to the following section of the input instance. The situation is the same at the mapping of the output instance. The maximum data size of the input instance and output instance are 500 bytes respectively. The following figure shows the general concept of the mapping information of the input instance and output instance. For details about the input, output and status instances, please refer to the "Appendix A (4. Assembly Object (04_{hex}))".



6. Supported Modbus Communication

Function Code (in hex)	Explanation
01	Read output status
02	Read input status
03	Read multiple data registers
04	Read input registers
0F	Write multiple bits
10	Write multiple data register

Appendix A: EtherNet/IP Object Model

1. Device Object Model

The Device Object Model is the logical organization of attributes, classes and services supported by a device. Objects are composed of attributes and services. There are three types of objects in any CIP device: Required Objects, Application Objects and Vendor Specific Objects.

Required Objects are object classes that must be supported by all devices on EtherNet/IP.

Applications Objects are classes that must be supported by all devices using the same profile. An example of a profile is a Discrete I/O device or an AC Drive. This ensures that devices from different vendors but with the same profile have a common interface to EtherNet/IP Client devices. For example, every AC Drive device must have a motor object among its required application objects. The attribute numbers for the maximum motor frequency and other motor data are predefined by the AC Drive profile to simply access to any device supporting the AC Drive profile.

Vendor Specific Objects are classes that add attributes and services that don't fit into the Required or Application Objects.

The required objects of the GW-7472 are list as below:

- Identity Object (0x01)
- Message Router Object (0x02)
- Assembly Object (0x04)
- Connection Manager Object (0x06)
- TCP Object (0xF5)
- Ethernet Link Object (0xF6)

2. Identity Object (01_{hex})

The Identify Object provides read only data that describes the general information about the device. The information may be the EtherNet/IP Vendor number, the major and minor revision and the serial number of the device. Your EtherNet/IP scanner has no direct control of any attributes in this object.

Class Attributes (Instance ID = 0_{hex})

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	01_{dec}	Get
2	Max Instance	UINT	01_{dec}	Get
3	Number of Instances	UINT	01_{dec}	Get
6	Max Class Attributes ID Number	UINT	07_{dec}	Get
7	Max Instance Attributes ID Number	UINT	07_{dec}	Get

Instance Attributes (Instance ID = 1_{hex})

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Vendor ID	UINT	803_{dec}	Get
2	Device Type	UINT	$0C_{hex}$	Get
3	Product Code	UINT	256_{dec}	Get
4	Product Major Revision Product Minor Revision	USINT USINT	01_{dec} 00_{dec}	Get
5	Status	WORD	00_{dec}	Get
6	Serial Number	UDINT	Unique 32 bit value	Get
7	Product Name **Structure of:** Product Name Size Product Name String	 SHORT STRING	 08_{dec} “GW-7472”	Get

Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
$0E_{hex}$	Yes	Yes	Get_Attribute_Single
01_{hex}	Yes (1,2,6,7)	Yes (1,2,3,4,5,6,7)	Get_Attributes_All
05_{hex}	No	Yes	Reset

3. Message Router Object (02_{hex})

The Message Router Object is used for routing the Explicit Message or UCMM to access the instance of the object with specific Class ID, Instance ID and Attribute ID. It provides two kinds of services for accessing any objects in the GW-7472.

Class Attributes (Instance ID = 0_{hex})

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	01_{dec}	Get
2	Max Instance	UINT	01_{dec}	Get
3	Number of Instances	UINT	01_{dec}	Get
6	Max Class Attributes ID Number	UINT	07_{dec}	Get
7	Max Instance Attributes ID Number	UINT	00_{dec}	Get

Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
$0E_{hex}$	Yes	No	Get_Attribute_Single
01_{hex}	Yes	No	Get_Attributes_All

4. Assembly Object (04_{hex})

An EtherNet/IP Assembly Object assembles data from other objects into input and output packages that are exchanged with the EtherNet/IP scanner. Input objects refer to the collection of data items that are transferred from the server (the GW-7472) to the Client (maybe the EtherNet/IP scanner). Output refers to the collection of data items that are transferred from the client (maybe the EtherNet/IP scanner) to the server (the GW-7472).

The GW-7472 provides Input/Output/Status Assembly for transferring data and status from the Modbus RTU network to the EtherNet/IP scanner. Generally, before using an EtherNet/IP scanner, you need to configure what object and instance you are interesting. Therefore, the following table must be applied to confirm the Class ID, Instance ID and Attribute ID when using the configuration tool of the EtherNet/IP scanner. If the configuration tool of the EtherNet/IP scanner supports the EDS loader, you can get the EDS file of the GW-7472 by using Utility. Please refer to the section 4.2.4 for more details.

INPUT/OUPUT ASSEMBLY

Input Instance: 65_{hex}

Output Instance: 66_{hex}

The input/output instance stores the Modbus Register data for the access of the EtherNet/IP scanner. The register data for all the Modbus nodes are packed into a maximum 500-byte data of the EtherNet/IP package. The size of the input/output instance is dependent on all the Modbus Read register data assigned by the Utility.

Class Attributes (Instance ID = 0_{hex})

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	02_{dec}	Get
2	Max Instance	UINT	03_{dec}	Get
3	Number of Instances	UINT	03_{dec}	Get
6	Max Class Attributes ID Number	UINT	07_{dec}	Get
7	Max Instance Attributes ID Number	UINT	03_{dec}	Get

Instance 64_{hex} Attributes (Configuration Instance)

Most EtherNet/IP scanner contains a configuration path when opening an Implicit Message connection to the GW-7472. Through the configuration path, the EtherNet/IP scanner can exchange the input and output data of the GW-7472. Therefore, there is no data needed in the Configuration Instance.

Instance 65_{hex} Attributes (Input Instance)

Attribute ID	Name	Data Type	Default Data Value	Access Rule
3	Serial Read Data **Structure of** Node Read Data 1 … Node Read Data n	BYTE [maximum 500]	All 0's	Get

The input data size is based on the arrangement of the input registers of the Modbus slaves configured by the GW-7472 Utility. The Input Instance data are packaged by following the command order defined in Utility.

Instance 66_{hex} Attributes (Output Instance)

Attribute ID	Name	Data Type	Default Data Value	Access Rule
3	Serial Data **Structure of** Node Read Data 1 … Node Read Data n	BYTE [maximum 500]	All 0's	Get/Set

The output data size is based on the arrangement of the output registers of the Modbus slaves configured by the GW-7472 Utility. The Output Instance data are packaged by following the command order defined in Utility.

Instance 67_{hex} Attributes (Command Status Instance)

Attribute ID	Name	Data Type	Default Data Value	Access Rule
3	Serial Data **Structure of** 1st Command status 2nd Command status … 30th Command status	BYTE [fixed to 30]	All 0's	Get/Set

Command Status (in hex)	Explanation
00	No Error
01	Illegal device ID
02	Illegal function code
03	Illegal data address
04	Receiving an Invalid command
05	CRC checking error
06	Timeout error occurred

Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
$0E_{hex}$	Yes	Yes	Get_Attribute_Single
10_{hex}	No	Yes	Set_Attribute_Single

5. Connection Manager Object (06_{hex})

The Connection Manager Object allocates and manages the internal resources associated with both Implicit and Explicit Messaging Connections. The specific instance generated by the Connection Manager Object is referred to as a Connection instance or a Connection Object.

Class Attributes (Instance ID = 0_{hex})

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	01_{dec}	Get
2	Max Instance	UINT	01_{dec}	Get
3	Number of Instances	UINT	01_{dec}	Get
6	Max Class Attributes ID Number	UINT	07_{dec}	Get
7	Max Instance Attributes ID Number	UINT	00_{dec}	Get

Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
$0E_{hex}$	Yes	No	Get_Attribute_Single
01_{hex}	Yes	No	Get_Attributes_All
$4E_{hex}$	No	Yes	Forward_Close
54_{hex}	No	Yes	Forward_Open

6. TCP/IP Interface Object ($F5_{hex}$)

The TCP/IP Interface Object contains read-only data that describes the TCP/IP connection parameters between the Gateway and the EtherNet/IP scanner. The configurable items include the GW-7472's IP address, network mask and gateway address. You can't directly control any attributes of this object.

Class Attributes (Instance ID = 0_{hex})

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	01_{dec}	Get
2	Max Instance	UINT	01_{dec}	Get
3	Number of Instances	UINT	01_{dec}	Get
6	Max Class Attributes ID Number	UINT	07_{dec}	Get
7	Max Instance Attributes ID Number	UINT	06_{dec}	Get

Instance Attributes (Instance ID = 1_{hex})

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Status[1]	UINT	01_{dec}	Get
2	Configuration Capability[2]	UINT	04_{dec}	Get
3	Configuration Control[3]	UINT	00_{dec}	Get
4	Physical Link Object[4] **Structure of:** Path Size Path	 UINT Padded EPATH	 02_{dec} $20F6_{hex}$, 2401_{hex}	Get
5	Interface Configuration[5] **Structure of:** IP Address Network Mask Gateway Address Name Server Name Server 2 Domain Name Size Domain Name	 UINT UINT UINT UINT UINT UINT UINT	 192 168 255 1_{dec} 255 255 0 0_{dec} 192 168 0 1_{dec} 0 0 0 0	Get
6	Host Name[6] **Structure of:** Host Name Size Host Name	 UINT String	 0 0	Get

Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
$0E_{hex}$	Yes	No	Get_Attribute_Single

[1] Section 5-3.2.2.1 of “Volume 2: EtherNet/IP Adaptation of CIP” from ODVA for more details on this attribute.

[2] Section 5-3.2.2.2 of “Volume 2: EtherNet/IP Adaptation of CIP” from ODVA for more details on this attribute.

[3] Section 5-3.2.2.3 of “Volume 2: EtherNet/IP Adaptation of CIP” from ODVA for more details on this attribute.

[4] Section 5-3.2.2.4 of “Volume 2: EtherNet/IP Adaptation of CIP” from ODVA for more details on this attribute.

[5] Section 5-3.2.2.5 of “Volume 2: EtherNet/IP Adaptation of CIP” from ODVA for more details on this attribute.

[6] Section 5-3.2.2.6 of “Volume 2: EtherNet/IP Adaptation of CIP” from ODVA for more details on this attribute.

7. Ethernet Link Object ($F6_{hex}$)

The Ethernet Link Object contains read-only data that describes the status of the physical Ethernet link. You can't directly control any attributes of this object.

Class Attributes (Instance ID = 0_{hex})

Attribute ID	Name	Data Type	Data Value	Access Rule
1	Revision	UINT	01_{dec}	Get
2	Max Instance	UINT	01_{dec}	Get
3	Number of Instances	UINT	01_{dec}	Get
6	Max Class Attributes ID Number	UINT	07_{dec}	Get
7	Max Instance Attributes ID Number	UINT	03_{dec}	Get

Instance Attributes (Instance ID = 1_{hex})

Attribute ID	Name	Data Type	Default Data Value	Access Rule
1	Interface Speed[7]	UDINT	100_{dec}	Get
2	Interface Flags[8]	DWORD	03_{dec}	Get
3	Physical Address[9]	ARRAYof 6 USINTs	00 0D E0 xx xx xx_{hex}	Get

Common Services

Service Code	Implemented for		Service Name
	Class	Instance	
$0E_{hex}$	Yes	Yes	Get_Attribute_Single
01_{hex}	Yes	Yes	Get_Attributes_All

[7] Section 5-4.2.2.1 of "Volume 2: EtherNet/IP Adaptation of CIP" from ODVA for more details on this attribute.
[8] Section 5-4.2.2.2 of "Volume 2: EtherNet/IP Adaptation of CIP" from ODVA for more details on this attribute.
[9] Section 5-4.2.2.3 of "Volume 2: EtherNet/IP Adaptation of CIP" from ODVA for more details on this attribute.

8. CIP General Status Code

Reference Volume 1: CIP Common Specification Appendix B

General Status Code (in hex)	Status Name	Description of Status
00	Success	Service was successfully performed by the object specified.
01	Connection failure	A connection related service failed along the connection path.
02	Resource unavailable	Resources needed for the object to perform the requested service were unavailable
04	Path segment error	The path segment identifier or the segment syntax was not understood by the processing node. Path processing shall stop when a path segment error is encountered.
05	Path destination unknown	The path is referencing an object class, instance or structure element that is not known or is not contained in the processing node. Path processing shall stop when a path destination unknown error is encountered.
08	Service not supported	The requested service was not implemented or was not defined for this Object Class/Instance.
09	Invalid attribute value	Invalid attribute data detected
0E	Attribute not settable	A request to modify a non-modifiable attribute was received.
13	Not enough data	The service did not supply enough data to perform the specified operation.
14	Attribute not supported	The attribute specified in the request is not supported
15	Too much data	The service supplied more data than was expected
16	Object does not exist	The object specified does not exist in the device.
20	Invalid parameter	A parameter associated with the request was invalid. This code is used when a parameter does not meet the requirements of this specification and/or the requirements defined in an Application Object Specification.
26	Path Size Invalid	The size of the path which was sent with the Service Request is either not large enough to allow the Request to be routed to an object or too much routing data was included.

9. Connection Manager Service Request Error Codes

Reference Volume 1: CIP Common Specification Table 3-5.29

General Status (in hex)	Extended Status (in hex)	Explanation and Description
00		Service completed successfully
01	100	CONNECTION IN USE OR DUPLICATE FORWARD OPEN This extended status code shall be returned when an originator is trying to make a connection to a target with which the originator may have already established a connection
01	103	TRANSPORT CLASS AND TRIGGER COMBINATION NOT SUPPORTED A transport class and trigger combination has been specified which is not supported by the target. Routers shall not fail the connection based on the transport class and trigger combination. Only targets shall return this extended status code.
01	106	OWNERSHIP CONFLICT The connection cannot be established since another connection already "owns" some of the resources required for this connection.
01	107	CONNECTION NOT FOUNT AT TARGET APPLICATION This extended status code shall be returned by the close connection request, where the connection which is to be closed is not active at the target node.
01	108	INVALID NETWORK CONNECTION PARAMETER This extended status code shall be returned as the result of specifying a connection type, connection priority, redundant owner or fixed / variable that is not supported by the target application. Only a target node shall return this extended status code.
01	109	INVALID CONNECTION SIZE This extended status code is returned when the target or router does not support the specified connection size.
01	113	CONNECTION MANAGER CANNOT SUPPORT ANY MORE CONNECTIONS
01	114	VENDOR ID OR PRODUCT CODE MISMATCH The Product Code or Vendor Id specified in the electronic key logical segment does not match the Product Code or Vendor Id of in the target device.
01	115	PRODUCT TYPE MISMATCH The Product Type specified in the electronic key logical segment does not match the Product Type of in the target device.
01	116	REVISION MISMATCH The major and minor revision specified in the electronic key logical segment does not correspond to a valid revision of the target device.
01	117	INVALID CONNECTIO POINT The connection point specified in the connection path does not correspond to a valid connection point for the target application.
01	118	INVALID CONFIGURATION FORMAT An instance number specified for the configuration data does not correspond to a configuration instance.
01	119	CONNECTION REQUEST FAILS SONCE THERE IS NO CONTROLLING CONNECTION CURRENTLY OPEN The extended status code shall be returned when an attempt is made to establish an echo (i.e. listen only) connection to a connection which has no controlling connection (i.e. owner).
01	11A	TARGET APPLICATION CANNOT SUPPORT ANY MORE CONNECTIONS The maximum number of connections supported by this instance of the Target

		Application has been exceeded.
01	205	PARAMETER ERROR IN UNCONNECTED SEND SERVICE One of the parameters in the unconnected send service was in error.
01	315	INVALID SEGMENT IN CONNECTION PATH Invalid Segment Type or Segment Value in Connection Path This extended status code is the result of a device being unable to decode the connection path. This could be caused by an unrecognized path type, a segment type occurring unexpectedly, or a myriad of other problems in the connection path.

Appendix B: Glossary

1. ARP (Address Resolution Protocol)

Consider two machines A and B that share a physical network. Each has an assigned IP address IP_A and IP_B, and a MAC address the MAC_A and MAC_B. The goal is to devise low-level software that hides MAC addresses and allows higher-level programs to work only with the IP addresses. Ultimately, however, communication must be carried out by the physical networks using whatever MAC address scheme the hardware supplies.

Suppose machine A wants to send a packet to machine B across a physical network to which they are both attached, but A only has the Internet address for B, IP_B. The question arises: how does A map that address to the MAC address for B, MAC_B?

ARP provides a method of dynamically mapping 32-bit IP address to the corresponding 48-bit MAC address. The term dynamic is used since it happens automatically and is normally not a concern for either the application user or the system administrator.

2. Clients and Servers

The client-server paradigm uses the direction of initiation to categorize whether a program is a client or server. In general, an application program that initiates peer to peer communication is called a client. End users usually invoke client programs when they use network services.

Most client programs consist of conventional application program develop tools. Each time a client program is executed; it contacts a server, sends a request and waits for a response. When the response arrives, the client program continues processing. Client programs are often easier to develop than servers, and usually require no special system privileges to operate.

By comparison, a server is any program that waits for incoming requests from a client program. The server receives a request from a client, performs the necessary computation and returns the result to the client.

3. Ethernet

The term Ethernet generally refers to a standard published in 1982 by Digital Equipment Corp., Intel Corp. and Xerox Corp. Ethernet is the most popular physical layer local area network (LAN) technology today. Ethernet is a best-effort delivery system that uses CSMA/CD technology. It recognizes hosts using 48-bit MAC address.

4. Firmware

Firmware is an alterable program located or stored in the semi-permanent storage area, e.g., ROM, EEPROM, or Flash memory.

5. Gateway

Computers that interconnect two networks and pass packets from one to the other are called Internet Gateways or Internet Routers. Gateways route packets that are based on the destination network, not on the destination host.

6. ICMP (Internet Control Messages Protocol)

No system works correctly all the time. ICMP provides a method of communicating between the Internet Protocol software on one machine and the Internet Protocol software on another. It allows gateways to send error or control messages to other gateways or allows a host to know what is wrong with the network communication.

7. Internet

Physically, the Internet is a collection of packet switching networks interconnected by gateways along with TCP/IP protocol that allows them to perform logically as a single, large and virtual network. The Internet recognizes hosts using 32-bit IP address.

8. IP (Internet Protocol) address

Every interface on an Internet must have a unique IP address (also called an Internet address). These addresses are 32-bit numbers. They are normally written as four decimal numbers, one for each byte of the address such as “192.168.41.1”. This is called dotted-decimal notation.

9. MAC (Media Access Control) address

To allow a computer to determine which packets are meant for it, each computer attached to an Ethernet is assigned a 48-bit integer known as its MAC address (also called an Ethernet address, hardware address or physical address). They are normally written as eight hexadecimal numbers such as "**00:71:88:af:12:3e:0f:01**". Ethernet hardware manufacturers purchase blocks of MAC addresses and assign them in sequence as they manufacture the Ethernet interface hardware. Thus, no two hardware interfaces have the same MAC address.

10. Packet

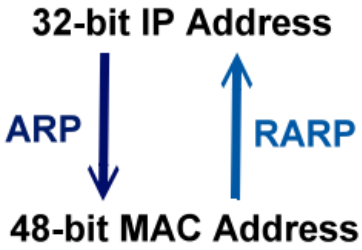
A packet is the unit of data sent across a physical network. It consists of a series of bits containing data and control information, including the source and the destination node (host) address, and is formatted for transmission from one node to another.

11. Ping

Ping sends an ICMP echo request message to a host, expecting an ICMP echo reply to be returned. Normally, if a host cannot be pinged, you won't be able to use Telnet or FTP to connect to the host. Conversely, if Telnet or FTP cannot be used to connect to a host, Ping is often the starting point to determine what the problem is.

12. RARP (Reverse Address Resolution Protocol)

RARP provides a method of dynamically mapping 48-bit MAC address to the corresponding 32-bit IP address.



13. Socket

Each TCP segment contains the source and destination port number that can be used to identify the sending and receiving application. These two values, along with the source and destination IP address in the IP header, uniquely identify each connection.

The combination of an IP address and a port number is called a socket.

14. Subnet Mask

Subnet mask is often simply called the mask. Given its own IP address and its subnet mask, a host can determine if a TCP/IP packet is destined for a host that is (1) on its own subnet, or (2) on a different network. If (1), the packet will be delivered directly; otherwise if, will be delivered via gateways or routers.

15. TCP (Transmission Control Protocol)

TCP provides a reliable flow of data between two hosts. It is associated with tasks such as dividing the data passed to it from applications into appropriately sized chunks for the network layer below, acknowledging received packets, setting timeouts to make certain that the other end acknowledges packets that are sent, and so on.

16. TCP/IP

The transmission Control Protocol (TCP) and the Internet Protocol (IP) are the standard network protocols. They are almost always implemented and used together and called TCP/IP. TCP/IP can be used to communicate across any set of interconnected networks.

17. UDP (User Datagram Protocol)

UDP provides a much simpler service to the application layer. It just sends packets of data from one host to the other. But there is no guarantee that the packets will reach the destination host.

Appendix C: FAQ

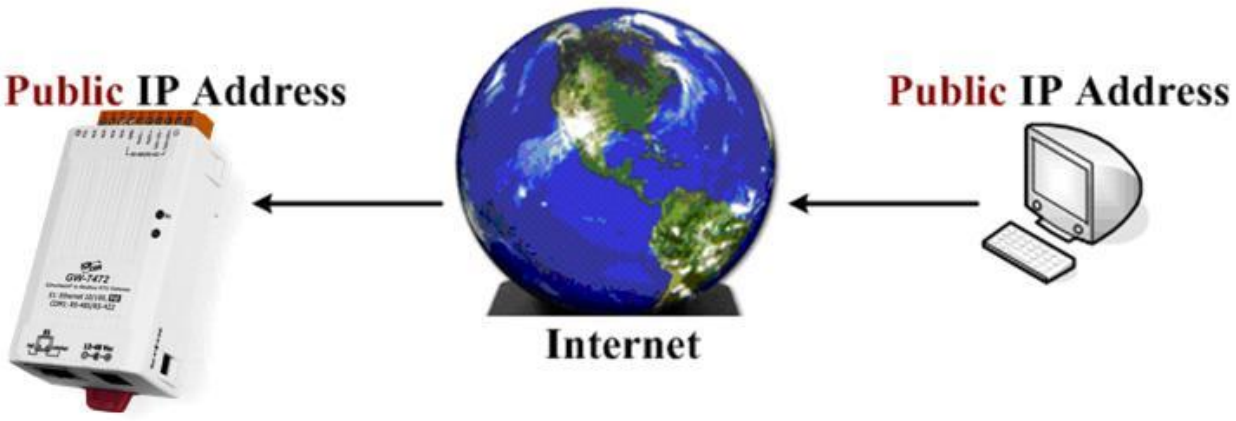
1. Why does the GW-7472 series module fail on a (public) Internet connection?

The default IP address of the GW-7472 is 192.168.255.1, which can be only used on a private Internet connection. A private network packet will not be routed via a (public) Internet connection, which is the reason why the GW-7472 failed on the Internet.

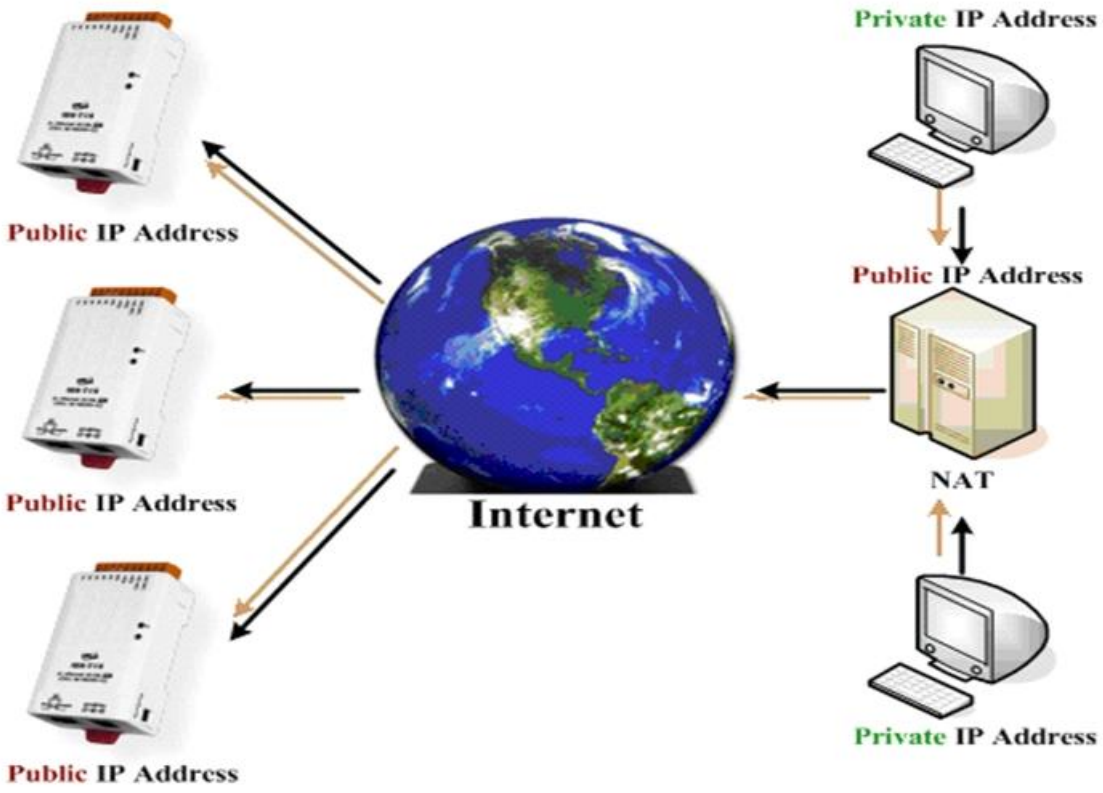
The IANA has reserved three address spaces for private internets (RFC1918).

10.0.0.0 - 10.255.255.255 (10/8 prefix)
172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

The GW-7472 can be operated on the Internet by using a legal public IP address. This address can be obtained from your ISP or network administrator.



A private internet client may communicate with a public Internet server (GW-7472s) only if the NAT service for the client is available.



Note!!

IANA	Internet Assigned Numbers Authority
RFC	Request for Comments
ISP	Internet Service Providers
NAT	Network Address Translator

Copyright @ 2010 by ICP DAS Co., Ltd. All Rights Reserved.