GW-7243D

DeviceNet Slave/ Modbus Master Gateway

User's Manual

Warranty

All products manufactured by ICP DAS are warranted against 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 consequent to 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

Copyright 2009 by ICP DAS Co., LTD. All rights are reserved worldwide.

Trademark

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

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table of Contents

1. General Information 4
1.1 Overview 4
1.2 DeviceNet Applications 5
1.3 Hardware Specification 6
1.4 DeviceNet Features 7
1.5 Modbus TCP Features 7
1.6 Modbus RTU/ASCII Features 7
1.7 GW-7243D Utility Features 8
1.8 Modbus Commands Supported 8
1.9 System Structure 9

2. Hardware 10
2.1 Pin Assignment 10
2.1.1 RS-232 & RS-485 & Power supply interface 11
2.1.2 Connect to DeviceNet devices 12
2.1.3 Ethernet connection 13
2.2 Terminator resistor settings 14
2.3 LED Indication 16
2.3.1 Power LED 16
2.3.2 DeviceNet Indicator Leds 17
2.3.2.1 MS LED 17
2.3.2.2 NS LED 17
2.3.2.3 IO LED 18
2.3.3 Five 7-Segment LED Displays 19

3. DeviceNet System 21
3.1 DeviceNet network Introduction 21
3.2 Predefined Master/Slave Connection Messages 25
3.2.1 Explicit Response/Request Messages 25
3.2.2 I/O Poll Command/Response Messages 26
3.3 EDS file 27

4. DeviceNet Profile Area 28
4.1 Introduction to the DeviceNet Objects of GW-7243D 28
4.2 DeviceNet Statement of Compliance 29
4.3 List of the GW-7243D's DeviceNet Object 30
4.4 Identity Object (Class ID: 0x01) 31

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.5 Message Router Object (Class ID: 0x02) 32
4.6 DeviceNet Object (Class ID: 0x03) 33
4.7 Connection Object (Class ID: 0x05) 34
4.8 GW-7243D Module Object (Class ID: 0x64) 37
4.9 COM1 Modbus RTU/ASCII Object (Class : 0x65) 39
4.10 COM2 Modbus RTU/ASCII Object (Class ID: 0x66) 41
4.11 Modbus TCP Server 01 Object (Class ID: 0x67) 43
4.12 Modbus TCP Server 02 Object (Class ID: 0x68) 45
4.13 Modbus TCP Server 03 Object (Class ID: 0x69) 47
4.14 Modbus TCP Server 04 Object (Class ID: 0x6A) 49

5. Configuration & Getting Started 51
5.1 Utility Overview 51
5.2 Install & Uninstall the GW-7243D Utility 52
5.2.1 Install GW-7243D Utility 52
5.2.2 Uninstall GW-7243D Utility 55
5.3 Configure GW-7243D by using Utility tool 58
5.3.1 Steps to configure the GW-7243D 58
5.3.2 Import/export all configuration to/from EEPROM 65
5.3.3 Restore to Factory Setting 67

6. Modbus Commands 68
6.1 “Read Coil Status” Command (0x01) 70
6.2 “Read Input Status” Command (0x02) 70
6.3 “Read Holding Registers” Command (0x03) 71
6.4 “Read Input Registers” Command (0x04) 71
6.5 “Force Single Coil” Command (0x05) 72
6.6 “Write Single Register” Command (0x06) 72
6.7 “Force Multiple Coils” Command (0x0F) 73
6.8 “Preset Multiple Registers” Command (0x10) 74
6.9 Exception Responses 75

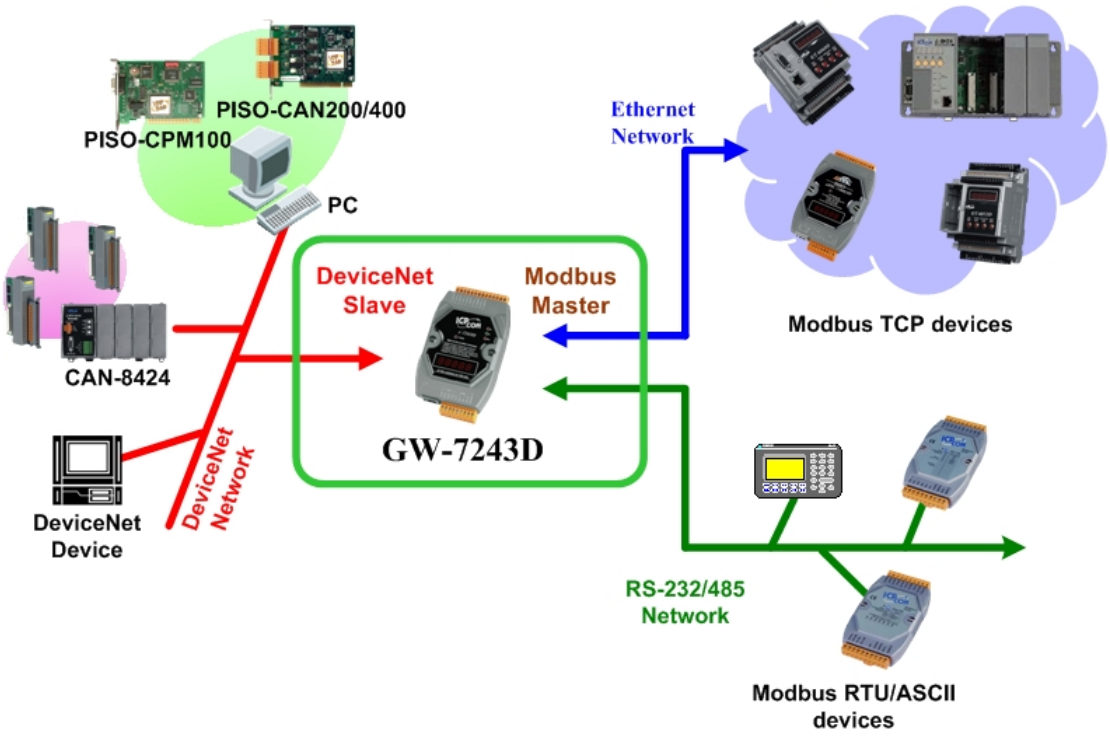
Appendix A: General Status Table 76

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1. General Information

1.1 Overview

The GW-7243D is one of DeviceNet products in ICP DAS and it stands as a DeviceNet slave to Modbus TCP/RTU/ASCII master Gateway device. It allows a master located on a DeviceNet network to enter a dialogue with slave devices on the Modbus TCP/RTU/ASCII network. In DeviceNet network, it functions as a “Group 2 Only Server” device. In Modbus network, GW-7243D represents a master device and sends request message to access the Modbus TCP/RTU/ASCII slave device by DeviceNet object definition. In order to simplify the protocol converting mechanism, we also provide the GW-7243D Utility tool for users to configure the device parameters and build EDS file for the DeviceNet slave device. Users can easily apply Modbus TCP/RTU/ASCII devices in DeviceNet applications through the GW-7243D. The application architecture is depicted as following figure. Users can connect the Modbus TCP/RTU/ASCII devices to the DeviceNet network via the GW-7243D.



LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.2 DeviceNet Applications

DeviceNet is the standardized network application layer optimized for factory automation. It is mainly used in low- and mid-volume automation systems. Some users have implemented DeviceNet protocol in machine control systems. The main DeviceNet application fields are demonstrated the following area. (For more information, please refer to www.odva.org):

• Production cell builds and tests CPUs	• Dinnerware production
• Beer brewery	• HVAC module production
• Equipment for food packing	• HVAC module production
• Fiberglass twist machine	• Trawler automation system
• Sponge production plant	• LCD manufacturing plant
• Sponge production plant	• Rolling steel door production
• Overhead storage bin production	• Bottling line
• Pocket-bread bakery	• Tight manufacturing



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.3 Hardware Specification

CPU	80186, 80 MHz or compatible
Flash	512 KB
SRAM	512 KB
EERROM	16 KB
Ethernet	RJ-45 x 1, 10/100Base-TX (Auto-negotiating, auto MDI/MDI-X, LED indicators)
Real Time Clock	Year-2000 compliance; seconds, minutes, hours, date of the month, year, valid up from 1980 to 2079
CAN Signal Support	CAN_H, CAN_L
Connecter	5-pin screw terminal connecter
Power Consumption	2.5 W
COM1	RS-232: TxD, RxD, RTS, CTS, GND
COM2	RS-485: DATA+, DATA-
COM Speed	115200 bps max
CAN Controller	Phillip SJA1000T CAN Controller
CAN Transceiver	Phillip 82C250 CAN Transceiver
LED Directors	MS, NS, IO
Display	7-segment LED: 5 digits
Isolated	1kV isolation on the CAN side
Required Supply Voltage	+10 ~ +30 V_{DC} (non-regulated)
Operating Temperature	-25 ~ +75°C
Storage Temperature	-40 ~ +85°C
Humidity	5 ~ 95% RH, non-condensing
Dimensions	122mm x 72mm x 33mm

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.4 DeviceNet Features

- Comply with DeviceNet specification Volume I/II, Release 2.0
- “Group 2 Only Server” DeviceNet subscriber
- On-line change baud rate and MAC ID of CAN
- MS̀, NS and IO LED indicators
- Support Offline Connection Set and Device Shutdown Message
- Connection supported:
  one “Explicit Connection”
  one “Polled Command/Response” connection

1.5 Modbus TCP Features

- Allow maximum 4 Modbus TCP devices.
- Support maximum 5 Modbus TCP commands for each Modbus TCP device.
- Support Modbus function codes: 0x01，0x02，0x03，0x04，0x05，0x06，0x0F and 0x10.
- Provide maximum 1920 channels DI/DO for a Modbus TCP command
- Provide maximum 120 channels AI/AO for a Modbus TCP command
- Allow maximum 2048 DI channels, 2048 DO channels, 1024 AI channels and 1024 AO channels for each Modbus TCP device.

1.6 Modbus RTU/ASCII Features

- Support both Modbus RTU and ASCII protocol for each COM port
- Allow maximum 10 Modbus RTU/ASCII commands for each COM port.
- Support Modbus function codes: 0x01，0x02，0x03，0x04，0x05，0x06，0x0F and 0x10.
- Baud rate: 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 bps
- Data bits: 7 or 8 bits
- Parity bits: None, even or odd
- Stop bits: 1 or 2 bits
- Provide maximum 1920 channels DI/DO for a Modbus RTU/ASCII command
- Provide maximum 120 channels AI/AO for a Modbus RTU/ASCII command
- Allow maximum 2048 DI channels, 2048 DO channels, 1024 AI channels and 1024 AO channels for each COM port.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.7 GW-7243D Utility Features

- Set IP/Gateway/Mask of GW-7243D
- Configure DeviceNet parameters, such as node ID, baud rate, and Poll I/O connection path.
- Configure COM port communication parameters
- Set Modbus TCP/RTU/ASCII protocol communication parameters
- Show Modbus TCP/RTU/ASCII protocol communication parameters
- Show DeviceNet Application Object configuration
- Produce EDS file dynamically

1.8 Modbus Commands Supported

The GW-7243D supports some kinds of Modbus commands. When users want to use Modbus TCP/RTU/ASCII devices on the DeviceNet network, they must notice that what kind of the function codes their Modbus TCP/RTU/ASCII devices support. The following table shows the function codes of Modbus supported by GW-7243D. For more details please refer to chapter 6.

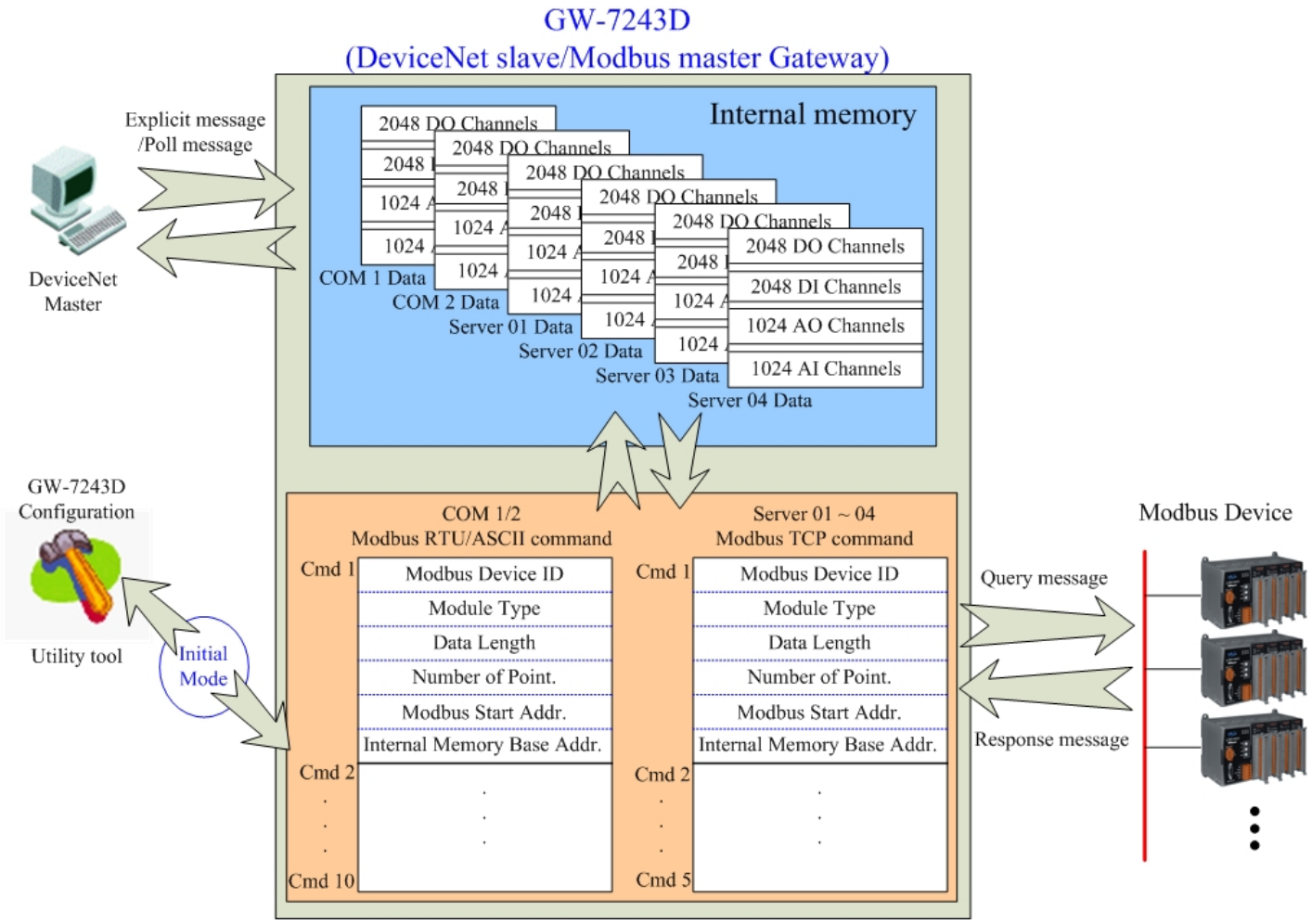
Function codes of Modbus supported by GW-7243D

Function Code	Modbus Command
0x01	Read coil status
0x02	Read input status
0x03	Read holding registers
0x04	Read input registers
0x05	Write single coil
0x06	Write single register
0x0F	Force multiple coils
0x10	Preset multiple registers

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.9 System Structure

The following architecture is the software structure of GW-7243D.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2. Hardware

2.1 Pin Assignment

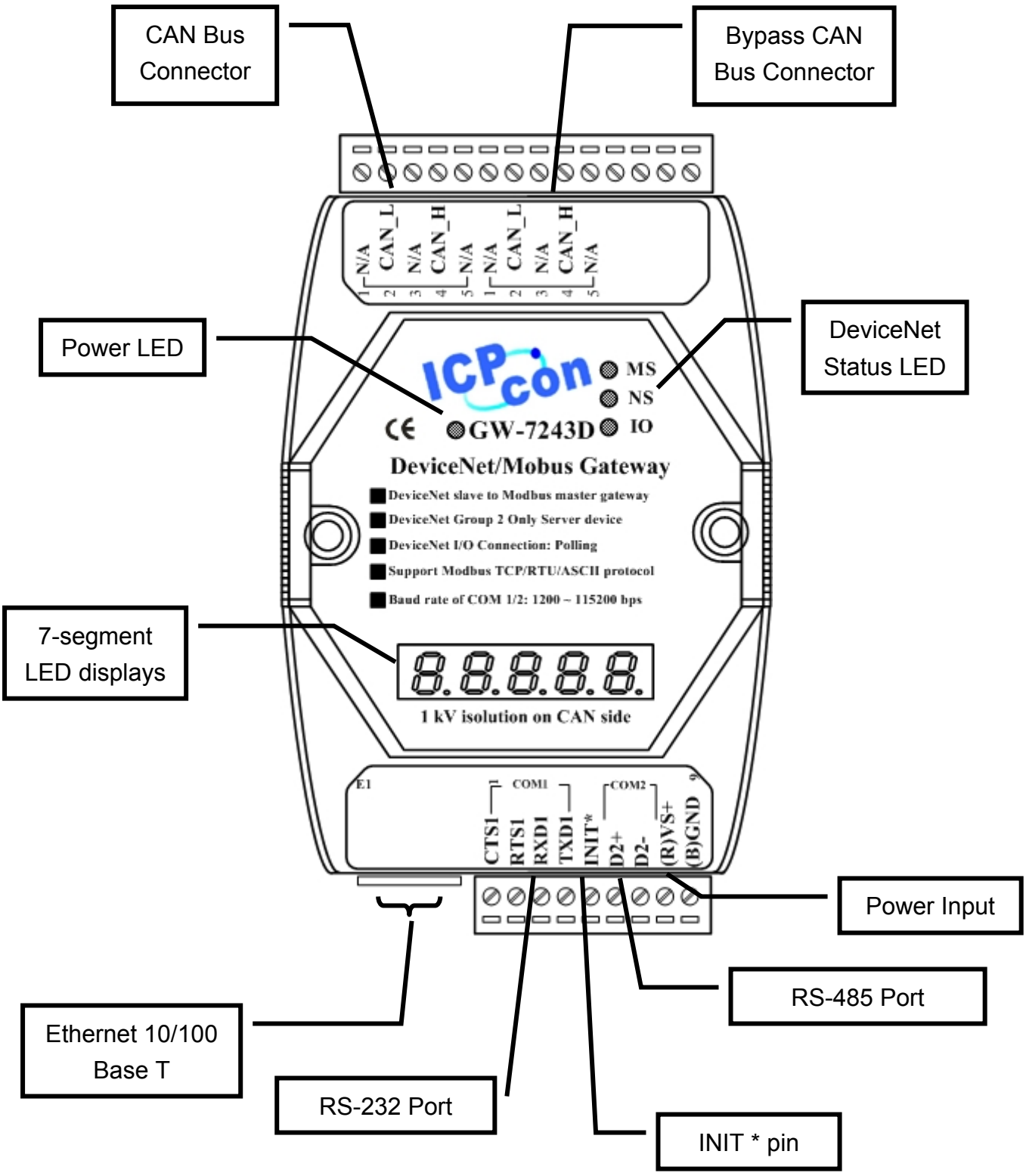
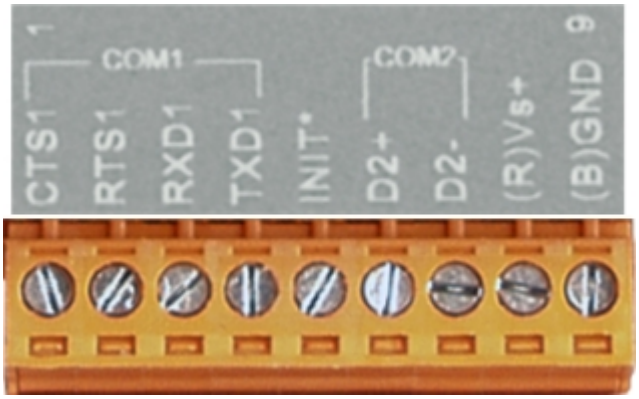


Figure 2-1 Hardware Structure of the GW-7243D

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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

The GW-7243D provides one RS-232 interface and one RS-485 interface. The GND-signal of COM1 is shared with pin-9, GND. The pin assignment is shown in table 2-1.



Pin	Name	Description
1	CTS1	CTS pin of COM1 (RS-232)
2	RTS1	RTS pin of COM1 (RS-232)
3	RXD1	RXD pin of COM1 (RS-232)
4	TXD1	TXD pin of COM1 (RS-232)
5	INIT*	Initial pin for enable/disable AUTOEXEC.BAT
6	D2+	Data+ pin of COM2 (RS-485)
7	D2-	Data- pin 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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.1.2 Connect to DeviceNet devices

In order to provide an easy CAN bus wiring, the GW-7243D supplies one CAN port with two CAN bus connector interfaces. Each connecter built on the GW-7243D looks like as figure 2-2 and table 2-2.

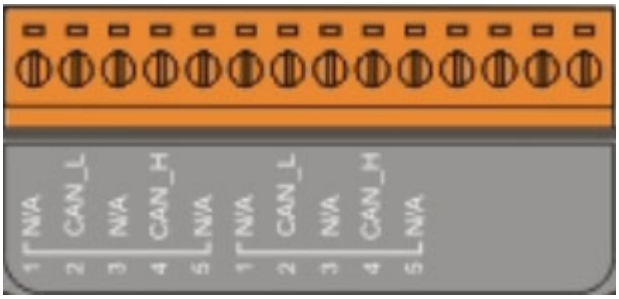


Figure 2-2 CAN bus connector

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 2-3.

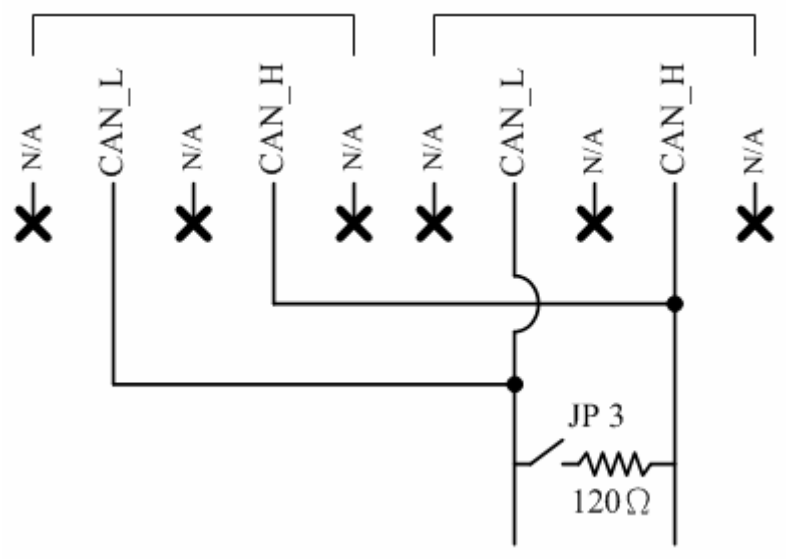
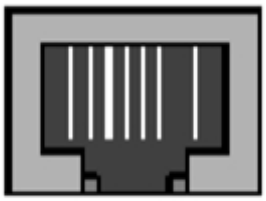


Figure 2-3 Electronic circuit of CAN bus connector

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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-7243D, 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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2 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 2-4.

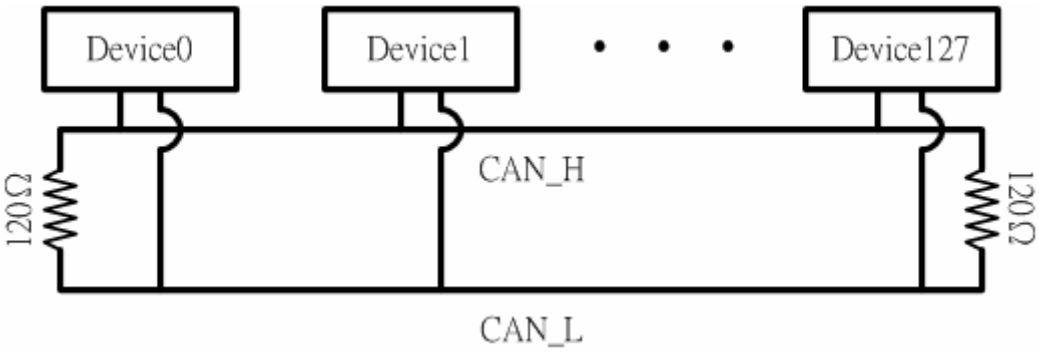


Figure 2-4: Terminator resistor

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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Therefore, the GW-7243D 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-7243D cover and use the JP3 jumper to activate the 120Ω terminator resistor built in the system, as in the figure 2-5. Note that the default setting is active. And about the J3 jumper setting, please refer the table 2-5.

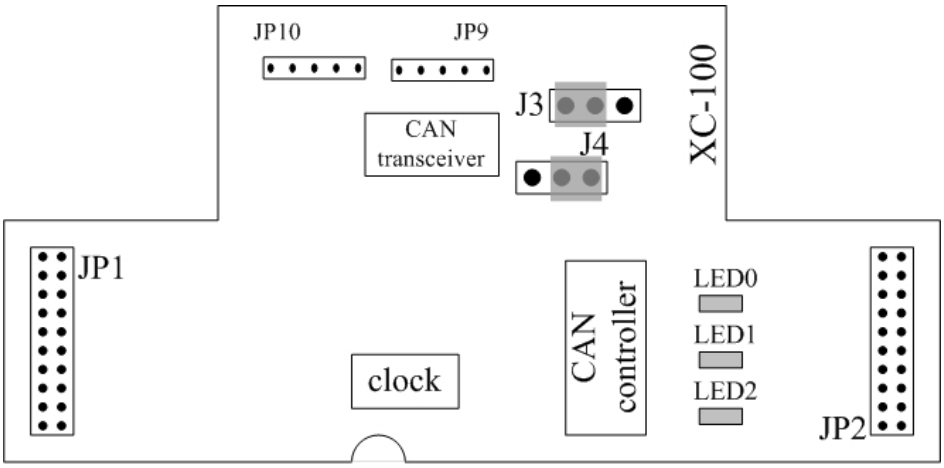


Figure 2-5 XC100 I/O expansion board LAYOUT

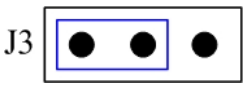
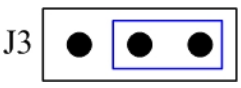
Apply the termination resistor(120Ω)	**Don't apply the termination resistor**
	

Table 2-5 J3 Jumper Selection

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.3 LED Indication

The GW-7243D acts as a DeviceNet slave to Modbus TCP/RTU/ASCII gateway. It provides some Leds to indicate what situation the GW-7243D is in.

2.3.1 Power LED

After connecting the GW-7243D with the electronic power (the range of input voltage is 10 ~ 30 V_{DC}). The Power LED will be turn on. If the Power LED is off after giving the proper voltage, please check the power and load of power supply firstly. If the situation is not improved, please communicate your local distributor to solve this problem. The corresponding conditions are given in table 2-6 and the location is shown in Figure 2.1.

Table 2-6 Power led conditions

Condition	**Status**	**Description**
Off	No power or module fail	No power supply or module failure.
Solid red	Normal	Device is working

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.3.2 DeviceNet Indicator Leds

The GW-7243D 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 IO LED (it is red). The Indicators assist maintenance personnel in quickly identifying a problem unit. The LED test is to be performed at power–up. When the DeviceNet communication events occur, these indicators will be triggered to glitter with different conditions.

2.3.2.1 MS LED

This LED provides device status and indicates whether or not the device is operating properly. Table 2-7 shows the conditions of MS status. Therefore, when the device is operated normally, the MS-LED must be turned off.

Table 2-7 MS led conditions

Condition	**Status**	**Description**
Red	Critical fault	Device has unrecoverable fault
Flashing red	Non-critical fault	Device has recoverable fault; to recover: Reconfigure device Reset device Perform error recovery

2.3.2.2 NS LED

This LED indicates the status of the communication link for the device. Table 2-8 shows the conditions of NS status. Therefore, when the device is correctly communicating in the network, the NS-LED is normally turned on.

Table 2-8 NS led conditions

Condition	**Status**	**Description**
Off	Off line	DeviceNet is off line
Flashing green	On line	DeviceNet is on line, but not communicating
Init solid green	Link failed	The device has detected an error that has rendered it incapable of communicating on the link; for example, detected a duplicate node address or network configuration error
Solid green	On line, communicating	DeviceNet is on communication

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.3.2.3 IO LED

This LED provides the status of the data access of the DeviceNet network. When DeviceNet master get/set the data of IO and configuration from Class ID 0x64 to Class ID 0x6A of GW-7243D, the LED would be flashed. Table 2-9 shows the different conditions for IO LED. Therefore, when the device IO function is working, the IO-LED should be flashed.

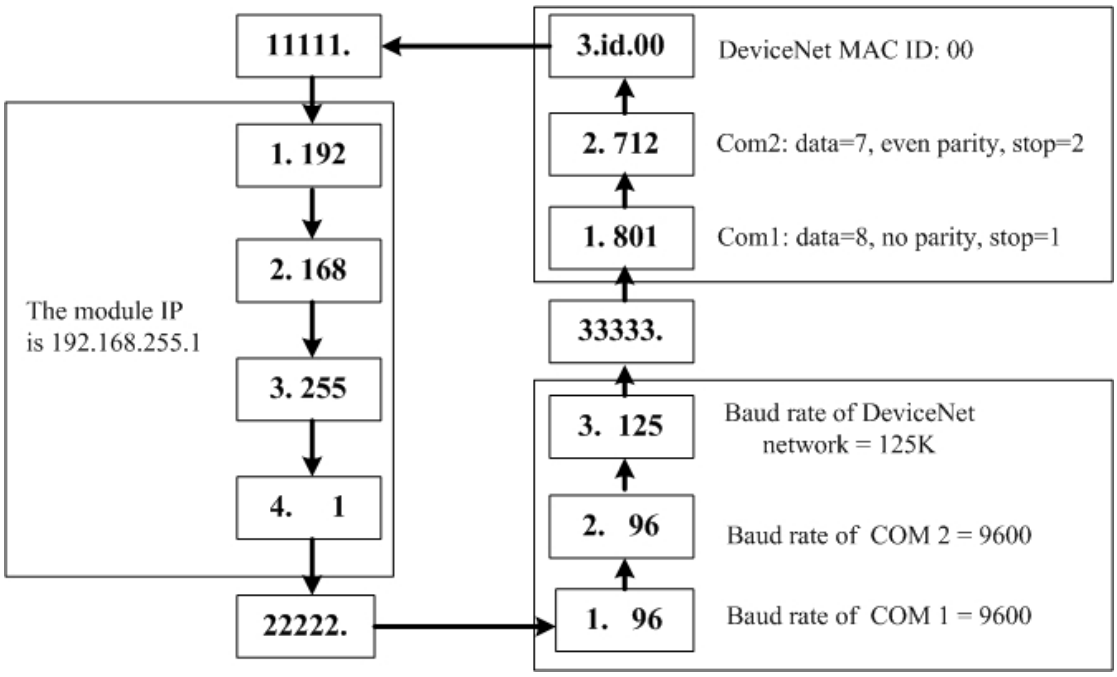
Table 2-9 IO led conditions

Condition	**Status**	**Description**
Off	No data	No data is being transmitted or received
Flashing red	Communicating	Data is being transmitted or received

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.3.3 Five 7-Segment LED Displays

The GW-7243D provides five 7-Segment LED displays and it scrolls to show the module information of GW-7243D.



The important information of GW-7243D can be divided as follows:

- Group-ID 11111: IP information of this GW-7243D
- Group-ID 22222: baud rate of all ports
- Group-ID 33333: configuration of all ports

The IP information format of GW-7243D is given as follows:

- Group-ID of 5-digit LED: 11111.
- LED-1: indicator, can be 1 or 2 or 3 or 4
- LED-2 ~ LED-5: IP of GW-7243D

The LED will show Group-ID first, and then show its IP as the above diagram indicates. If users change IP, the value shown will change immediately. The default shipping IP = 192.168.255.1 ➔ the LED show sequence is given as above diagram.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

The baud-rate format of COM ports are given as follows:

- Group-ID of 5-digit LED: 22222.
- LED-1: COM port number
- LED-2 ~ LED-5: value of (baud rate/100)

The baud-rate format of CAN port is given as follows:

- LED-1: indicator, 3, DeviceNet baud rate.
- LED-2 ~ LED-5: value of (baud rate/1000)

The COM port No. is shown in LED-1 and its baud rate is shown in the LED-2 ~ LED-5. The COM port baud rate = (value of LED-2 ~ LED-5)*100. Therefore, the display “1. 96” means baud rate of COM1 is 9600 bps; the display “2.1152” means baud rate of COM2 is 115200 bps.

The DeviceNet baud rate is equal to (value of LED-2 ~ LED-5)*1000. Therefore, the display “3. 125” means that baud rate of DeviceNet network is 125Kbps; the display “3. 500” means that the baud rate of DeviceNet network is 500Kbps. All baud rate of GW-7243D’s port will be shown one by one.

The configuration of COM ports are given as follows:

- Group-ID of 5-digit LED: 33333.
- LED-1: COM port number
- LED-3: data bit, 7 or 8
- LED-4: parity bit, 0=no parity, 1=even parity, 2=odd parity
- LED-5: stop bit, 1 or 2

The configuration of CAN port is given as follows:

- LED-2/ LED-3: fix string, “id.”.
- LED-4/ LED-5: DeviceNet MAC ID of this module, default 01

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3. DeviceNet System

3.1 DeviceNet network Introduction

DeviceNet is one of the kinds of the network protocols based on the CAN bus which are mainly used for machine control in embedded network, such as in textile machinery, printing machines, injection molding machinery, or packaging machines. DeviceNet is a low level network that provides connections between simple industrial devices (sensors, actuators) and higher-level devices (controllers). It allows direct peer to peer data exchange between nodes in an organized and, if necessary, deterministic manner. The network management functions specified in DeviceNet simplifies project design, implementation and diagnosis by providing standard mechanisms for network start-up and error management. DeviceNet defines a connection-based scheme to facilitate all application communications. A DeviceNet connection provides a communication path between multiple endpoints. The endpoints of a connection are applications that need to share data. The figure 3-11 shows the DeviceNet layer in the control and information layers.

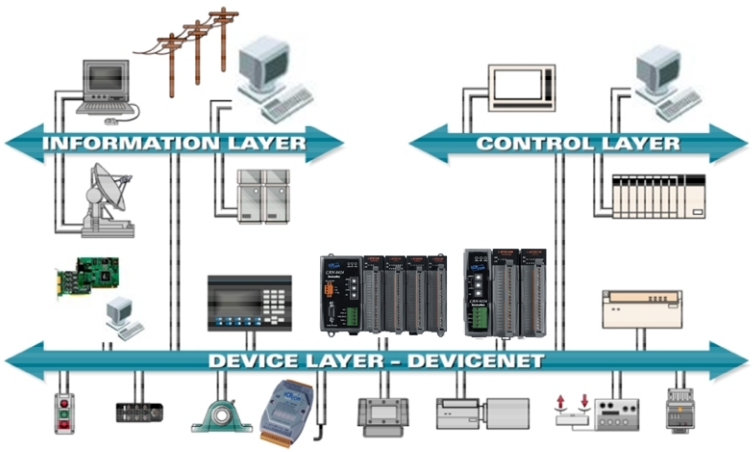


Figure 3-1 DeviceNet layer

The DeviceNet Communication Protocol is based on the concept of connections. One must establish a connection with a device in order to exclude information with that device. To establish a connection, each gateway implements Predefined Master/Slave Connection Set through the DeviceNet network. After establishing the explicit connections, the connection is then used to move information from one node to the other. Once IO connections

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

have been established, I/O data may be moved among devices in the network.

The 11-bit CAN identifier is used to identify the connection. DeviceNet defines four separate groups of 11-bit CAN identifiers: Group 1, Group 2, Group 3, and Group 4 described in the Table 3-1. With respect to Connection Based Messages, the Connection ID is placed within the CAN Identifier Field. With this in mind, the below figure also describes the components for a DeviceNet Connection ID. Because of the arbitration scheme defined by CAN, Group 1 messages have a higher priority than group 2 messages and group 2 messages have higher priority than group 3 messages and so on. This prioritization must be taken into consideration when establishing connections.

Table 3-1 DeviceNet CAN Identifier Field

IDENTIFIER BITS											IDENTITY USAGE	HEX RANGE
10	9	8	7	6	5	4	3	2	1	0		
0	Group 1 Message ID				Source MAC ID						Group 1 Messages	000 – 3ff
1	0	MAC ID						Group 2 Message ID			Group 2 Messages	400 – 5ff
1	1	Group 3 Message ID				Source MAC ID					Group 3 Messages	600-7bf
1	1	1	1	1	Group 4 Message ID						Group 4 Messages	7c0–7ef

The GW-7243D provides the Predefined Master/slave Connection Set for users to establish connections. The Predefined Master/Slave Connection Set is a set of Connections that facilitate communications typically seen in a Master/Slave relationship. Many of the steps involved in the creation and configuration of an Application connection have been removed within the Predefined Master/Slave Connection Set definition. This, in turn, presents the means by which a communication environment can be established using less network and device resources. The CAN Identifier Fields associated with the Predefined Master/Slave Connection Set are shown in the table 3-2. The table defines the Identifiers that are to be used with all connection based message involved in the Predefined Master/Slave Connection Set and, as such, it also illustrates the produced_connection_id and consumed_connection_id attributes associated with Predefined Master/Slave Connection Objects.

Note: The Master is the device that gathers and distributes I/O data for the process controller. Slaves are the devices from which the Master gathers I/O data and to which the Master distributes I/O data.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 3-2 Predefined Master/Slave Connection Set Identify Fields

IDENTIFIER BITS											IDENTITY USAGE	HEX RANGE
10	9	8	7	6	5	4	3	2	1	0		
0	Group 1 Message ID				Source MAC ID						Group 1 Messages	000 – 3ff
0	1	1	0	0	Source MAC ID						Slave's Multicast Poll Response	
0	1	1	0	1	Source MAC ID						Slave's I/O Change of State or Cyclic Message	
0	1	1	1	0	Source MAC ID						Slave's I/O Bit–Strobe Response Message	
0	1	1	1	1	Source MAC ID						Slave's I/O Poll Response or Change of State/Cyclic Acknowledge Message	
1	0	MAC ID						Group 2 Message ID			Group 2 Messages	400 – 5ff
1	0	Source MAC ID						0	0	0	Master's I/O Bit–Strobe Command Message	
1	0	Multicast MAC ID						0	0	1	Master's I/O Multicast Poll Command Message	
1	0	Destination MAC ID						0	1	0	Master's Change of State or Cyclic Acknowledge Message	
1	0	Source MAC ID						0	1	1	Slave's Explicit/ Unconnected Response Messages/ Device Heartbeat Message/ Device Shutdown Message	
1	0	Destination MAC ID						1	0	0	Master's Explicit Request Messages	
1	0	Destination MAC ID						1	0	1	Master's I/O Poll Command/Change of State/Cyclic Message	
1	0	Destination MAC ID						1	1	0	Group 2 Only Unconnected Explicit Request Messages (reserved)	
1	0	Destination MAC ID						1	1	1	Duplicate MAC ID Check Messages	

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

A device within a DeviceNet network is represented by the below object model. The object model provides a template for organizing and implementing the Attributes (data), Services (methods or procedures) and behaviors of the components within a DeviceNet product. The figure 3-2 depicts the object model for GW-7243D (Group 2 Only Server).

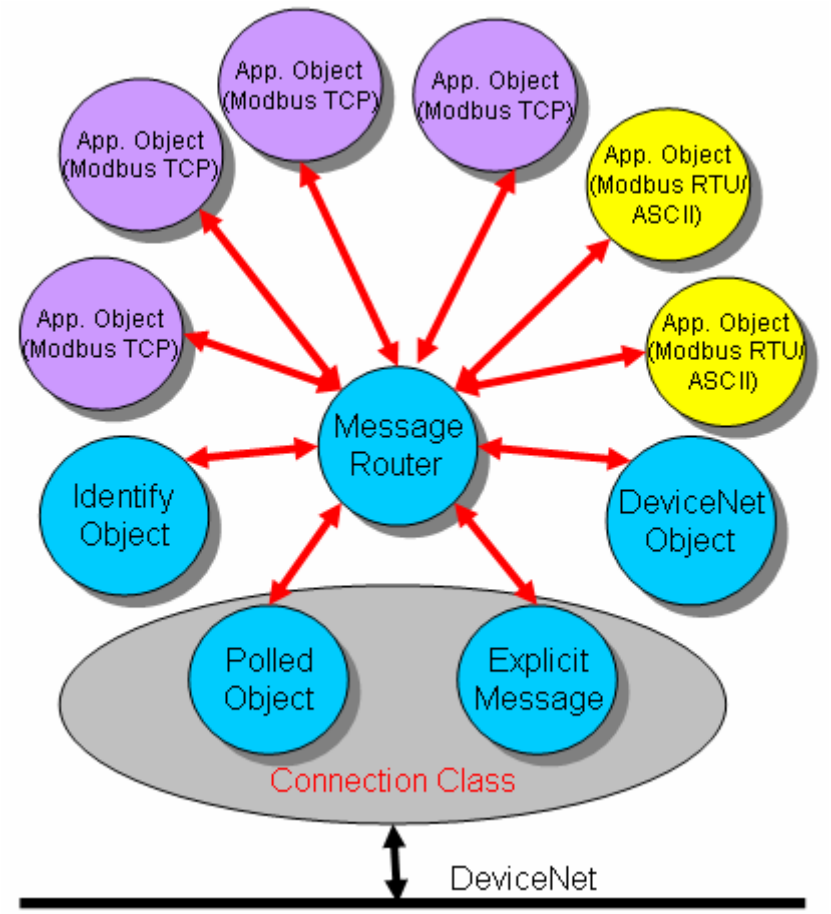


Figure 3-2 Object model of GW-7243D

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3.2 Predefined Master/Slave Connection Messages

The GW-7243D provides “Predefined Master Slave Connection Set” device. Users must understand these connection set to know how to operate the device.

With the Predefined Master Slave Connection Set, DeviceNet allows devices with fewer resources to take part in DeviceNet network communication. For this reason a set of identifiers has been reserved within the Message Group 2 to simplify the movement of I/O and configuration data typically seen in Master/Slave relationships. The steps, which are necessary to create and configure a connection between devices, have been removed within the Predefined Set. The Predefined Master/Slave Connection Set allows for the establishment of a DeviceNet communication environment using less network and Device resources. The Predefined Connection Set contains one Explicit Messaging Connection and allows several different I/O Connections which include a Bit Strobe Command/Response, Poll Command/Response, Change of State and Cyclic. The following type of messages are processed by a DeviceNet Slave

3.2.1 Explicit Response/Request Messages

Explicit Request Messages are used to perform operations such as reading and writing attributes. Explicit response Messages indicate the results of the slaves answer to attempt to service an Explicit Request massage. Within a Slave Explicit Request and Response messages are received/transmitted by a single Connection Object. The architecture is as figure 3-3.

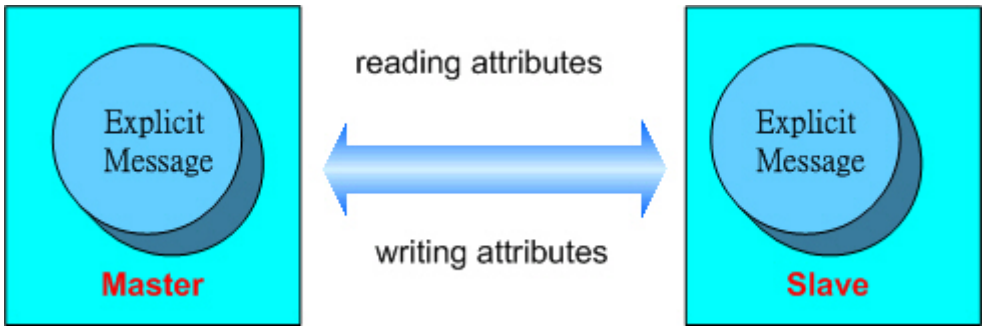


Figure 3-3 Architecture of Explicit message

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3.2.2 I/O Poll Command/Response Messages

The Poll Command Message is a command that is transmitted by the Master. A Poll Command is directed towards a single, specific Slave (point-to-point connection). A Master must transmit a separate Poll command message for each one of its Slaves that will be polled. The Poll Response Message is an I/O message that the Slave transmits back to the Master when a Poll Command is received. Within a Slave the two messages are received or transmitted by a single Connection Object. The architecture is as figure 3-4.

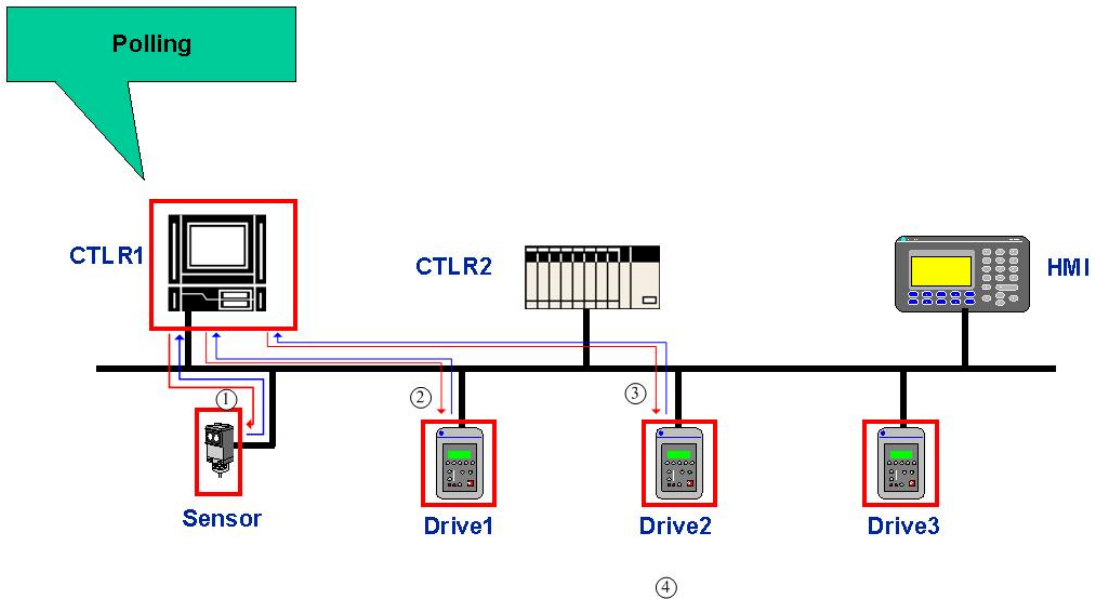


Figure 3-4 Architecture of IO poll message

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

3.3 EDS file

An Electronic Data Sheet is an external disk that contains information about configurable attributes for devices, including the object addresses of each parameter. The following figure shows that the configuration tool uses the EDS file to configure those Modbus TCP/RTU/ASCII devices. The application objects in these devices represent the destination addresses for the configuration data. These addresses are encoded in the EDS. ICP DAS provides users with GW-7243D Utility software to create the suitable EDS file. The EDS file system architecture is as figure 3-5.

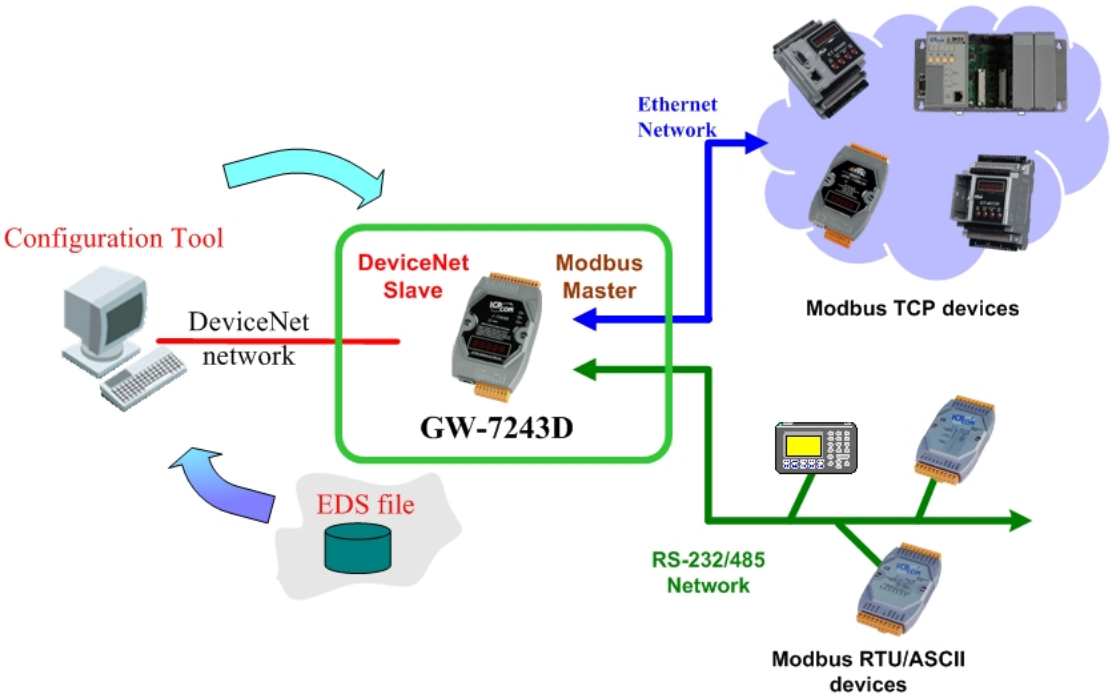


Figure 3-5 Architecture of EDS file

EDS provides information about the device's configuration data in terms of the following:

- context
- content
- format

The information in an EDS allows configuration tools to provide informative screens that guide a user through the steps necessary to configure a device. ICP DAS provides CAN utilities, so that users can setup their own EDS file. You can use the EDS file in the DeviceNet Master to access DeviceNet Slave devices.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4. DeviceNet Profile Area

This chapter is for users who want to understand more detailed information related to the GW-7243D device when using the DeviceNet protocol. This section documents the detailed functions for each object class that is implemented in the DeviceNet network

4.1 Introduction to the DeviceNet Objects of GW-7243D

The GW-7243D has been developed in accordance with the ***Object Modeling*** from the DeviceNet protocol. This model leads to a method used for addressing the GW-7243D's data made up of four separate values: MAC ID、Class ID、Instance ID and Attribute ID. An address made up in this way is known as a "**Path**". The Connection by Explicit Messaging, for example, uses paths of this sort to exchange data from one node to another on a DeviceNet network. See table 4-1 to know the address field of GW-7243D.

Table 4-1 Address field of GW-7243D

Address	Min. - Max.	Description
Node	0-63	This field allows you to address one subscriber out of the series of subscribers on a DeviceNet network using its **MAC ID**.
Class	1-65535	All objects sharing the same characteristics belong to the same class, characterized by its **Class ID**.
Instance	0-65535	All instances from one class share the same **attributes** but each of them has its own set of values for these attributes.
Attribute	1-255	It is assigned some sort of value (byte, unsigned integer, character string, etc.) in order to supply information about the subscriber's status or to make settings on the subscriber's behaviors

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.2 DeviceNet Statement of Compliance

General Device Data

Conforms to DeviceNet Specification	Volume I, II Release 2.0
Product Name	ICPDAS-GW-7243D
Vendor Name	ICP DAS Co., LTD
Device Profile	Communications Adapter
Product Code	5

DeviceNet Physical Conformance Data

Network Power Consumption (Max)	Open-Hardwired
Isolated Physical Layer	Yes
LED Supported	Yes
MAC ID Setting	Software
Device MAC ID	Software (Default is 0x01)
Communication Rate Setting	Software (Default is 125k bits/s)
Predefined Master/Slave Connection Set	Group 2 Only Server
Connection supported	One “Explicit Connection” One “Polled Command/Response” Connection

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.3 List of the GW-7243D's DeviceNet Object

The GW-7243D supports the following DeviceNet object classes.

Object Type	Class ID	Instance number	Interfaces
Identity	01 (0x01)	1	Message Router
Message Router	02 (0x02)	1	Explicit message connection
DeviceNet	03 (0x03)	1	Message Router
Connection	05 (0x05)	2	Poll IO connection and Explicit message connection
GW-7243D module	100 (0x64)	3	Message Router
COM1 Modbus RTU/ASCII	101 (0x65)	10 (max)	Message Router
COM2 Modbus RTU/ASCII	102 (0x66)	10 (max)	Message Router
TCP server 01	103(0x67)	5(max)	Message Router
TCP server 02	104(0x68)	5(max)	Message Router
TCP server 03	105(0x69)	5(max)	Message Router
TCP server 04	106(0x6A)	5(max)	Message Router

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.4 Identity Object (Class ID: 0x01)

This object provides the identification and general information about the device. It is described in chapter 6-2 of volume II of the DeviceNet specifications.

Class Attribute

Attribute ID	Attribute name	Data Type	Method	Value
0x01	Revision	UINT	Get	-
0x02	Max Instance	UINT	Get	1

Class Service

Service Code	Service Name	Need
0x0E	Get_Attribute_Single	Required

Instance Attribute

Attribute ID	Description	Data Type	Method	Value
0x01	Vendor UINT	UINT	Get	803
0x02	Device type	UINT	Get	12
0x03	Product code	UINT	Get	5
0x04	Vendor Revision “major.minor”	USINT, USINT	Get	Depended on firmware version
0x05	Status	UINT	Get	(16-bit register)
0x06	Serial number	UDINT	Get	(variable)
0x07	Product name	Short_String	Get	“ICPDAS-GW-7243D”
0x0A	Heartbeat Interval	USINT	Get/Set	0-65535

Instance Service

Service Code	Service Name	Need
0x0E	Get_Attribute_Single	Required
0x10	Set_Attribute_Single	Required
0x05	Reset	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.5 Message Router Object (Class ID: 0x02)

The “Message Router” object is the element through which all objects of the “Explicit messages” type pass, so that they can be routed to the objects they are intended for. This object is described in chapter 6-3 of volume II of the DeviceNet specifications.

Class Attribute

Attribute ID	Attribute name	Data Type	Method	Value
0x01	Revision	UINT	Get	-

Class Service

Service Code	Service Name	Need
0x0E	Get_Attribute_Single	Required

Instance Attributes:
This instance has no attributes.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.6 DeviceNet Object (Class ID: 0x03)

The DeviceNet Object is used to provide the configuration and status of a physical attachment on the DeviceNet network. It is described in chapter 5-5 of volume I of the DeviceNet specifications. The GW-7243D is a “Group 2 Only Server” type subscriber (see chapter 7-9 of volume I of the DeviceNet specifications).

Class Attribute

Attribute ID	Attribute name	Data Type	Method	Value
0x01	Revision	UINT	Get	-

Class Service

Service Code	Service name	Need
0x0E	Get_Attribute_Single	Required

Instance Attribute

Attribute ID	Description	Data Type	Method	Value
0x01	MAC ID	USINT	Get/Set	0~63
0x02	Baud rate	USINT	Get/Set	0~2
0x03	BOI	USINT	Get/Set	0
0x04	Bus-off counter	USINT	Get/Set	0
0x05	Allocation information	BYTE	Get/Set	(variable)

Instance Service

Service Code	Service name	Need
0x0E	Get_Attribute_Single	Optional
0x10	Set_Attribute_Single	Optional
0x4B	Allocation Master/Slave Connection Set	Optional
0x4C	Release Master/Slave Connection Set	Optional

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.7 Connection Object (Class ID: 0x05)

This section presents the externally visible characteristics of the connection objects associated with the Predefined Master/Slave Connection Set within slave devices. With the GW-7243D, the “Connection” object has up to two instances (Instance ID 0x01 to 0x02). Each of these instances represents one of the two ends of a virtual connection established between two nodes on the DeviceNet network. Each instance of this object belongs to one of the two following types of connection: Explicit connection, allowing *Explicit Messages* to be sent, or *Poll I/O Connections*. This object is described in chapter 5-4 of volume I of the DeviceNet specifications.

Here is a brief description of the two instances of the GW-7243D’s “Connection” object.

Instance ID	Type of connection	Connection name
0x01	Explicit Messaging	Explicit Connection
0x02	I/O Connection	Polled Command/Response Connection

Class Attribute

Attribute ID	Attribute Name	Data Type	Method	Value
0x01	Revision	UINT	Get	-

Class Service

Service Code	Service Name	Need
0x0E	Get_Attribute_Single	Required

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Instance (0x01) Attribute: Explicit Connection

Attribute ID	Attribute Name	Data Type	Method	Value
0x01	State	USINT	Get	0x00 to 0x05
0x02	Instance type	USINT	Get	0x00
0x03	Transport class trigger	BYTE	Get	0x83
0x04	Produced connection id	UINT	Get	Table 3-2
0x05	Consumed connection id	UINT	Get	Table 3-2
0x06	Initial comm. characteristics	BYTE	Get	0x21
0x07	Produced connection size	UINT	Get	0x84
0x08	Consumed connection size	UINT	Get	0x84
0x09	Expected packet rate	UINT	Get/Set	0x09c4
0x0C	Watchdog timeout action	USINT	Get/Set	0x01
0x0D	Produced connection path length	UINT	Get	0x00
0x0E	Produced connection path	EPATH	Get/Set	(empty path)
0x0F	Consumed connection path length	UINT	Get	0x00
0x10	Consumed connection path	EPATH	Get/Set	(empty path)
0x11	Production inhibit time	UINT	Get/Set	0x00

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Instance (0x02) Attribute: Polled Command/Response Connection

Attribute ID	Attribute Name	Data Type	Method	Value
0x01	State	USINT	Get	0x00 to 0x04
0x02	Instance type	USINT	Get	0x01
0x03	Transport class trigger	BYTE	Get	0x83
0x04	Produced connection id	UINT	Get	Table 3-2
0x05	Consumed connection id	UINT	Get	Table 3-2
0x06	Initial comm. characteristics	BYTE	Get	0x01
0x07	Produced connection size	UINT	Get	(size of the input data)
0x08	Consumed connection size	UINT	Get	(size of the output data)
0x09	Expected packet rate	UINT	Get/Set	0x00
0x0C	Watchdog timeout action	USINT	Get	0x00
0x0D	Produced connection path length	UINT	Get	0x06
0x0E	Produced connection path	EPATH	Get/Set	(area path)
0x0F	Consumed connection path length	UINT	Get	0x06
0x10	Consumed connection path	EPATH	Get/Set	(area path)
0x11	Production inhibit time	UINT	Get/Set	0x00

Instance Service

Service Code	Service Name	Need
0x05	Reset	Option
0x09	Delete	Option
0x0E	Get_Attribute_Single	Conditional
0x10	Set_Attribute_Single	Conditional

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.8 GW-7243D Module Object (Class ID: 0x64)

This object defines the communication parameters of the GW-7243D. There are three instances. These instances describe the parameters of the Ethernet, COM 1 and COM 2 ports. The DeviceNet master can read these configurations from this object. The definitions of this object are shown as the following tables.

Class Attribute

Attribute ID	Attribute name	Data Type	Method	Value
0x01	Revision	UINT	Get	-
0x02	Max Instance	UINT	Get	3

Class Service

Service Code	Service Name	Need
0x0E	Get_Attribute_Single	Required

Instance Attribute (ID = 0x00)

Attribute ID	Description	Data Type	Method	Value
0x01	Application Object Version	UINT	Get	-
0x02	Application Object Max. Instance Count	UINT	Get	3

Instance Attribute (ID = 0x01) (for GW-7243D's Ethernet Parameters)

Attribute ID	Description	Data Type	Method	Value
0x01	GW-7243D IP	UDINT	Get/Set	Determined by user defined
0x02	GW-7243D Gateway	UDINT	Get/Set	Determined by user defined
0x03	GW-7243D Mask	UDINT	Get/Set	Determined by user defined

Instance Attribute (ID = 0x02) (for GW-7243D's COM 1 Parameters)

Attribute ID	Description	Data Type	Method	Value
0x01	GW-7243D COM 1 Baud Rate	UDINT	Get/Set	1200 ~ 115200
0x02	GW-7243D COM 1	USINT	Get/Set	7 or 8

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

	Data Bits			
0x03	GW-7243D COM 1 Parity Bit	USINT	Get/Set	0 : None 1 : Even 2 : Odd
0x04	GW-7243D COM 1 Stop Bits	USINT	Get/Set	1 or 2

Instance Attribute (ID = 0x03) (for GW-7243D's COM 2 Parameters)

Attribute ID	Description	Data Type	Method	Value
0x01	GW-7243D COM 2 Baud Rate	UDINT	Get/Set	1200 ~ 115200
0x02	GW-7243D COM 2 Data Bits	USINT	Get/Set	7 or 8
0x03	GW-7243D COM 2 Parity Bit	USINT	Get/Set	0 : None 1 : Even 2 : Odd
0x04	GW-7243D COM 2 Stop Bits	USINT	Get/Set	1 or 2

Instance Service

Service Code	Service name	Need
0x0E	Get_Attribute_Single	Required
0x10	Set_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.9 COM1 Modbus RTU/ASCII Object (Class : 0x65)

This object defines the command parameters of the Modbus RTU/ASCII which is used to communicate to the Modbus slave via COM 1 port. There are ten instances for ten Modbus RTU/ASCII commands. The DeviceNet master can read the configuration and IO data from this object. The definition of this object is shown as the following tables.

Class Attribute (ID = 0x00) (for Connection Status / Parameters)

Attribute ID	Description	Data Type	Method	Value
0x01	Application Objection Version	UINT	Get	-
0x02	Application Objection Max. Instance	UINT	Get	0x0A
0x03	Connection Mode	USINT	Get	0 : None 1 : RTU 2 : ASCII
0x04	Command Timeout	UDINT	Get	Determined by user defined

Class Service

Service Code	Service Name	Need
0x0E	Get_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Instance Attribute (ID = 0x01 ~ 0x0A) (Modbus RTU/ASCII COM1 Commands)

Attribute ID	Description	Data Type	Method	Value
0x01	Command Valid Flag	USINT	Get	0 : No command 1 : Command valid
0x02	Net ID	USINT	Get	Determined by user defined
0x03	Function Code	USINT	Get	Determined by user defined
0x04	Map Memory Address	UINT	Get	Determined by user defined
0x05	Modbus Address	UINT	Get	Determined by user defined
0x06	I/O Channel Count	UINT	Get	Determined by user defined
0x07	Command Error Response	UINT	Get	Refer to Appendix A
0x0A	Input Discrete Data	USINT […]	Get	(array of values)
0x0B	Coils Status Data	USINT […]	Get/Set	(array of values)
0x0C	Input Register Data	UINT […]	Get	(array of values)
0x0D	Registers Data	UINT […]	Get/Set	(array of values)

Instance Service

Service Code	Service name	Need
0x0E	Get_Attribute_Single	Required
0x10	Set_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.10 COM2 Modbus RTU/ASCII Object (Class ID: 0x66)

This object defines the command parameters of the Modbus RTU/ASCII which is used to communicate to the Modbus slave via COM 2 port. There are ten instances for ten Modbus RTU/ASCII commands. The DeviceNet master can read the configuration and IO data from this object. The definition of this object is shown as the following tables.

Class Attribute (ID = 0x00) (for Connection Status / Parameters)

Attribute ID	Description	Data Type	Method	Value
0x01	Application Objection Version	UINT	Get	-
0x02	Application Objection Max. Instance	UINT	Get	0x0A
0x03	Connection Mode	USINT	Get	0 : None 1 : RTU 2 : ASCII
0x04	Command Timeout	UDINT	Get	Determined by user defined

Class Service

Service Code	Service Name	Need
0x0E	Get_Attribute_Single	Required

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Instance Attribute (ID = 0x01 ~ 0x0A) (Modbus RTU/ASCII COM2 Commands)

Attribute ID	Description	Data Type	Method	Value
0x01	Command Valid Flag	USINT	Get	0 : No command 1 : Command valid
0x02	Net ID	USINT	Get	Determined by user defined
0x03	Function Code	USINT	Get	Determined by user defined
0x04	Map Memory Address	UINT	Get	Determined by user defined
0x05	Modbus Address	UINT	Get	Determined by user defined
0x06	I/O Channel Count	UINT	Get	Determined by user defined
0x07	Command Error Response	UINT	Get	Refer to Appendix A
0x0A	Input Discrete Data	USINT […]	Get	(array of values)
0x0B	Coils Status Data	USINT […]	Get/Set	(array of values)
0x0C	Input Register Data	UINT […]	Get	(array of values)
0x0D	Registers Data	UINT […]	Get/Set	(array of values)

Instance Service

Service Code	Service name	Need
0x0E	Get_Attribute_Single	Required
0x10	Set_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.11 Modbus TCP Server 01 Object (Class ID: 0x67)

This object defines the command parameters of the Modbus TCP server 01 which is used to communicate to the Modbus slave via Ethernet port. There are five instances for five Modbus TCP commands. The DeviceNet master can read the configuration and IO data from this object. The definition of this object is shown as the following tables.

Class Attribute (ID = 0x00) (for Connection Status / Parameters)

Attribute ID	Description	Data Type	Method	Value
0x01	Application Objection Version	UINT	Get	-
0x02	Application Objection Max. Instance	UINT	Get	0x05
0x03	Connection Valid Flag	USINT	Get	0 : No command 1 : Command valid
0x04	Server IP	UDINT	Get	Determined by user defined
0x05	Connection Timeout	UDINT	Get	Determined by user defined
0x06	Retry Connection Timeout	UDINT	Get	Determined by user defined
0x07	Command Timeout	UDINT	Get	Determined by user defined
0x08	Connection Status	UINT	Get	Refer to Appendix A

Class Service

Service Code	Service Name	Need
0x0E	Get_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Instance Attribute (ID = 0x01 ~ 0x05) (Modbus TCP Server Commands)

Attribute ID	Description	Data Type	Method	Value
0x01	Command Valid Flag	USINT	Get	0 : No command 1 : Command valid
0x02	Net ID	USINT	Get	Determined by user defined
0x03	Function Code	USINT	Get	Determined by user defined
0x04	Map Memory Address	UINT	Get	Determined by user defined
0x05	Modbus Address	UINT	Get	Determined by user defined
0x06	I/O Channel Count	UINT	Get	Determined by user defined
0x07	Command Error Response	UINT	Get	Refer to Appendix A
0x0A	Input Discrete Data	USINT […]	Get	(array of values)
0x0B	Coils Status Data	USINT […]	Get/Set	(array of values)
0x0C	Input Register Data	UINT […]	Get	(array of values)
0x0D	Registers Data	USINT […]	Get/Set	(array of values)

Instance Service

Service Code	Service name	Need
0x0E	Get_Attribute_Single	Required
0x10	Set_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.12 Modbus TCP Server 02 Object (Class ID: 0x68)

This object defines the command parameters of the Modbus TCP server 02 which is used to communicate to the Modbus slave via Ethernet port. There are five instances for five Modbus TCP commands. The DeviceNet master can read the configuration and IO data from this object. The definition of this object is shown as the following tables.

Class Attribute (ID = 0x00) (for Connection Status / Parameters)

Attribute ID	Description	Data Type	Method	Value
0x01	Application Objection Version	UINT	Get	-
0x02	Application Objection Max. Instance Count	UINT	Get	0x05
0x03	Valid Flag	USINT	Get	0 : No command 1 : Command valid
0x04	Server IP	UDINT	Get	Determined by user defined
0x05	Connection Timeout	UDINT	Get	Determined by user defined
0x06	Retry Connection Timeout	UDINT	Get	Determined by user defined
0x07	Command Timeout	UDINT	Get	Determined by user defined
0x08	Connection Status	UINT	Get	Refer to Appendix A

Class Service

Service Code	Service Name	Need
0x0E	Get_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Instance Attribute (ID = 0x01 ~ 0x05) (Modbus TCP Server Commands)

Attribute ID	Description	Data Type	Method	Value
0x01	Command Valid Flag	USINT	Get	0 : No command 1 : Command valid
0x02	Net ID	USINT	Get	Determined by user defined
0x03	Function Code	USINT	Get	Determined by user defined
0x04	Map Memory Address	UINT	Get	Determined by user defined
0x05	Modbus Address	UINT	Get	Determined by user defined
0x06	I/O Channel Count	UINT	Get	Determined by user defined
0x07	Command Error Response	UINT	Get	Refer to Appendix A
0x0A	Input Discrete Data	USINT […]	Get	(array of values)
0x0B	Coils Status Data	USINT […]	Get/Set	(array of values)
0x0C	Input Register Data	UINT […]	Get	(array of values)
0x0D	Registers Data	UINT […]	Get/Set	(array of values)

Instance Service

Service Code	Service name	Need
0x0E	Get_Attribute_Single	Required
0x10	Set_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.13 Modbus TCP Server 03 Object (Class ID: 0x69)

This object defines the command parameters of the Modbus TCP server 03 which is used to communicate to the Modbus slave via Ethernet port. There are five instances for five Modbus TCP commands. The DeviceNet master can read the configuration and IO data from this object. The definition of this object is shown as the following tables.

Class Attribute (ID = 0x00) (for Connection Status / Parameters)

Attribute ID	Description	Data Type	Method	Value
0x01	Application Objection Version	UINT	Get	-
0x02	Application Objection Max. Instance	UINT	Get	0x05
0x03	Connection Valid Flag	USINT	Get	0 : No command 1 : Command valid
0x04	Server IP	UDINT	Get	Determined by user defined
0x05	Connection Timeout	UDINT	Get	Determined by user defined
0x06	Retry Connection Timeout	UDINT	Get	Determined by user defined
0x07	Command Timeout	UDINT	Get	Determined by user defined
0x08	Connection Status	UINT	Get	Refer to Appendix A

Class Service

Service Code	Service Name	Need
0x0E	Get_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Instance Attribute (ID = 0x01 ~ 0x05) (Modbus TCP Server Commands)

Attribute ID	Description	Data Type	Method	Value
0x01	Command Valid Flag	USINT	Get	0 : No command 1 : Command valid
0x02	Net ID	USINT	Get	Determined by user defined
0x03	Function Code	USINT	Get	Determined by user defined
0x04	Map Memory Address	UINT	Get	Determined by user defined
0x05	Modbus Address	UINT	Get	Determined by user defined
0x06	I/O Channel Count	UINT	Get	Determined by user defined
0x07	Command Error Response	UINT	Get	Refer to Appendix A
0x0A	Input Discrete Data	USINT […]	Get	(array of values)
0x0B	Coils Status Data	USINT […]	Get/Set	(array of values)
0x0C	Input Register Data	UINT […]	Get	(array of values)
0x0D	Registers Data	UINT […]	Get/Set	(array of values)

Instance Service

Service Code	Service name	Need
0x0E	Get_Attribute_Single	Required
0x10	Set_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.14 Modbus TCP Server 04 Object (Class ID: 0x6A)

This object defines the command parameters of the Modbus TCP server 04 which is used to communicate to the Modbus slave via Ethernet port. There are five instances for five Modbus TCP commands. The DeviceNet master can read the configuration and IO data from this object. The definition of this object is shown as the following tables.

Class Attribute (ID = 0x00) (for Connection Status / Parameters)

Attribute ID	Description	Data Type	Method	Value
0x01	Application Objection Version	UINT	Get	1.0
0x02	Application Objection Max. Instance	UINT	Get	0x05
0x03	Connection Valid Flag	USINT	Get	0 : No command 1 : Command valid
0x04	Server IP	UDINT	Get	Determined by user defined
0x05	Connection Timeout	UDINT	Get	Determined by user defined
0x06	Retry Connection Timeout	UDINT	Get	Determined by user defined
0x07	Command Timeout	UDINT	Get	Determined by user defined
0x08	Connection Status	UINT	Get	Refer to Appendix A

Class Service

Service Code	Service Name	Need
0x0E	Get_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Instance Attribute (ID = 0x01 ~ 0x05) (Modbus TCP Server Commands)

Attribute ID	Description	Data Type	Method	Value
0x01	Command Valid Flag	USINT	Get	0 : No command 1 : Command valid
0x02	Net ID	USINT	Get	Determined by user defined
0x03	Function Code	USINT	Get	Determined by user defined
0x04	Map Memory Address	UINT	Get	Determined by user defined
0x05	Modbus Address	UINT	Get	Determined by user defined
0x06	I/O Channel Count	UINT	Get	Determined by user defined
0x07	Command Error Response	UINT	Get	Refer to Appendix A
0x0A	Input Discrete Data	USINT […]	Get	(array of values)
0x0B	Coils Status Data	USINT […]	Get/Set	(array of values)
0x0C	Input Register Data	UINT […]	Get	(array of values)
0x0D	Registers Data	UINT […]	Get/Set	(array of values)

Instance Service

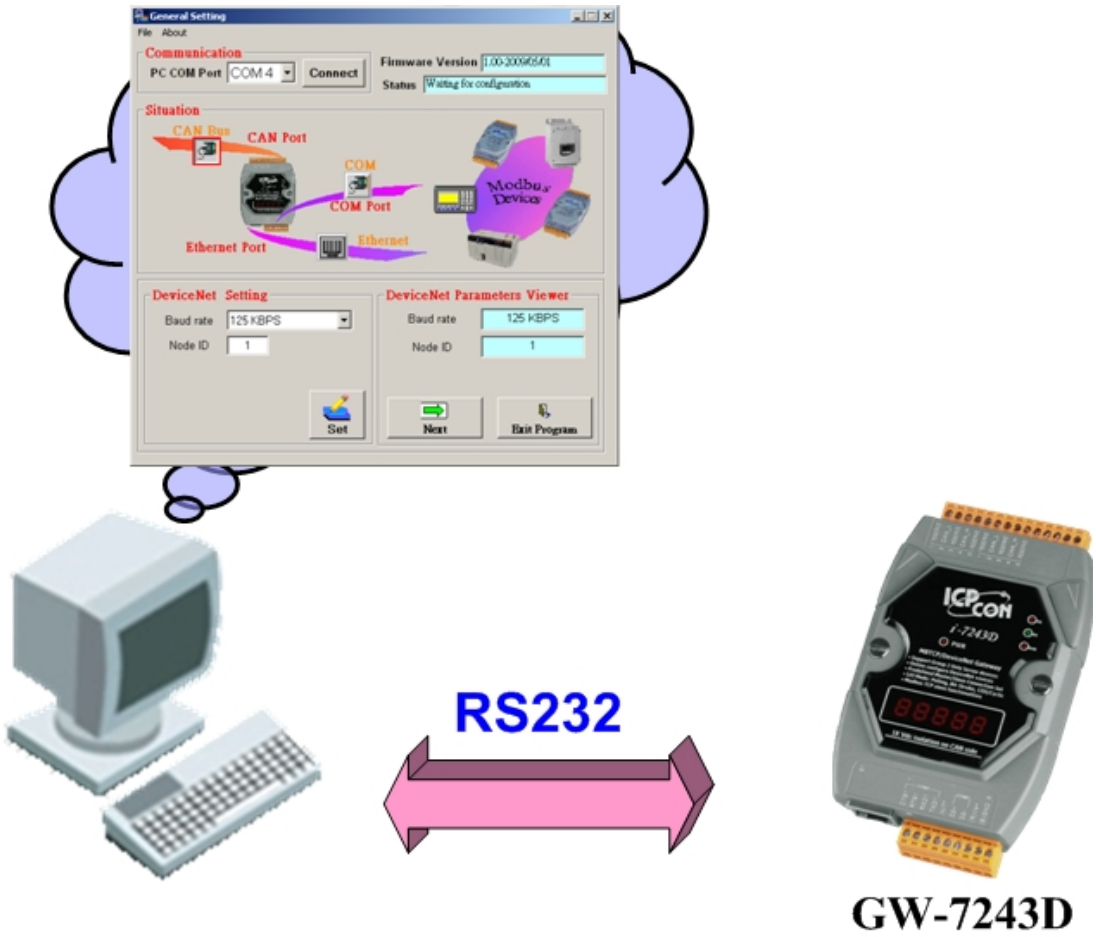
Service Code	Service name	Need
0x0E	Get_Attribute_Single	Required
0x10	Set_Attribute_Single	Required

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

5. Configuration & Getting Started

5.1 Utility Overview

Before users apply the GW-7243D into the DeviceNet application, they must understand the relationship between the DeviceNet application object and Modbus commands in the GW-7243D. ICP DAS provides the GW-7243D Utility for user to configure the communication parameters, I/O connection path and the EDS file of the GW-7243D device. The software also provides the information of application objects and communication parameters that they set.



Architecture of the GW-7243D Utility

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

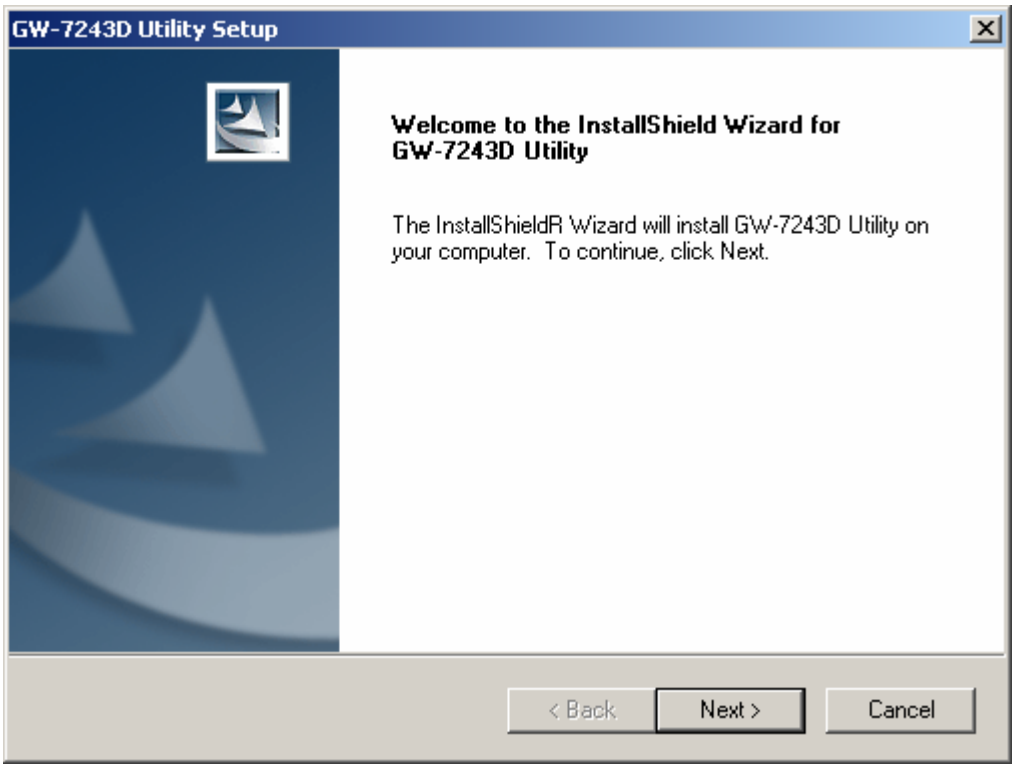
5.2 Install & Uninstall the GW-7243D Utility

5.2.1 Install GW-7243D Utility

Step1: Download the GW-7243D Utility setup file from the web site ftp://ftp.icpdas.com.tw/pub/cd/can_cd/devicenet/gateway/GW-7243D/utility/ or from the CD-ROM disk in following the path of "/ CAN-CD / DeviceNet / Gateway / GW-7243D / Utility /

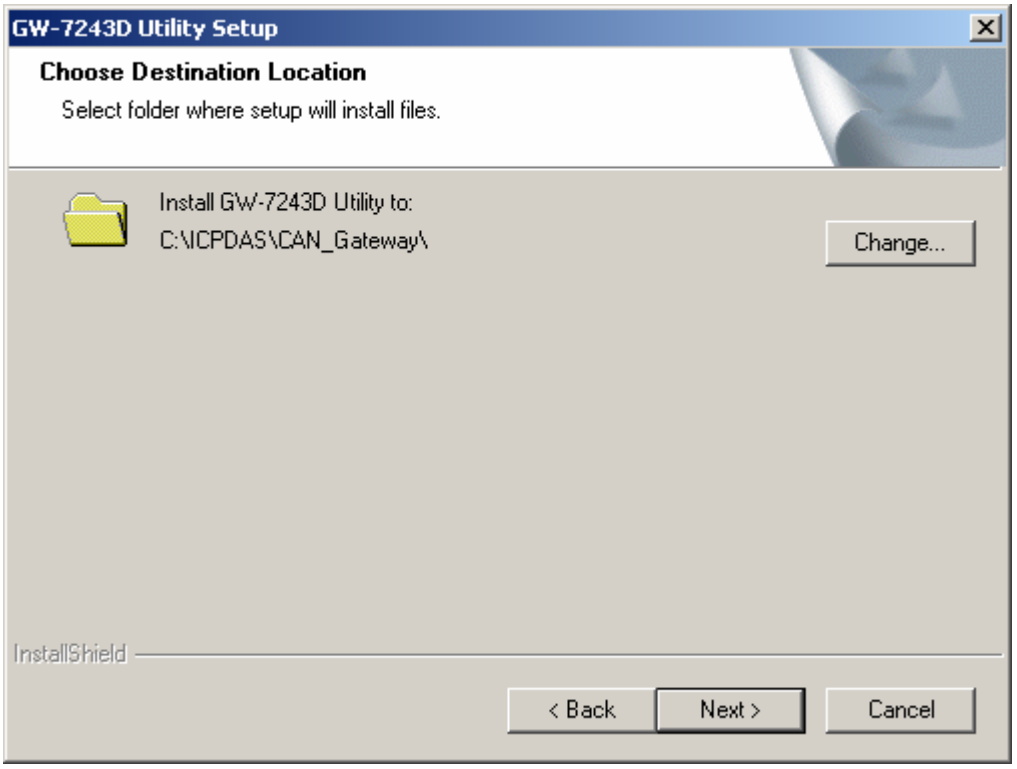
Step 2: Execute the setup.exe file to install GW-7243D Utility.

Step 3: A "Welcome" window pops up to prompt user to begin the installation. Click the "Next" button to go on the next step.

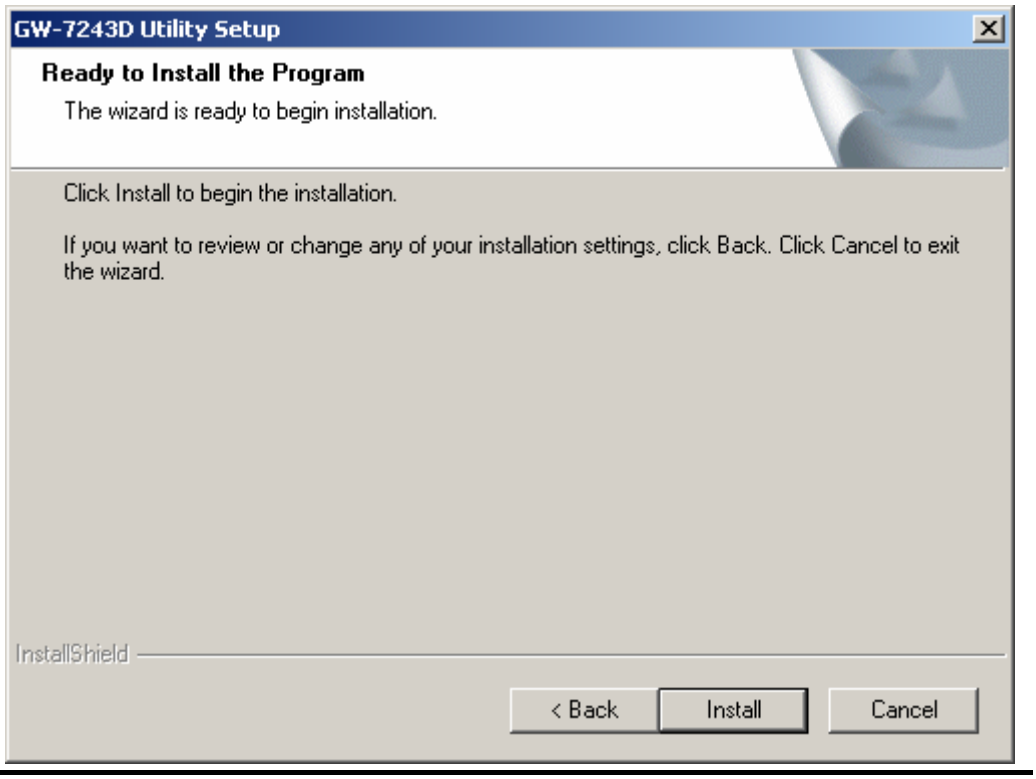


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Step4: A “Choose Destination Location” window will pop up for deciding the installation path. Here, default path is used. Click “Next” button to go on the next step.

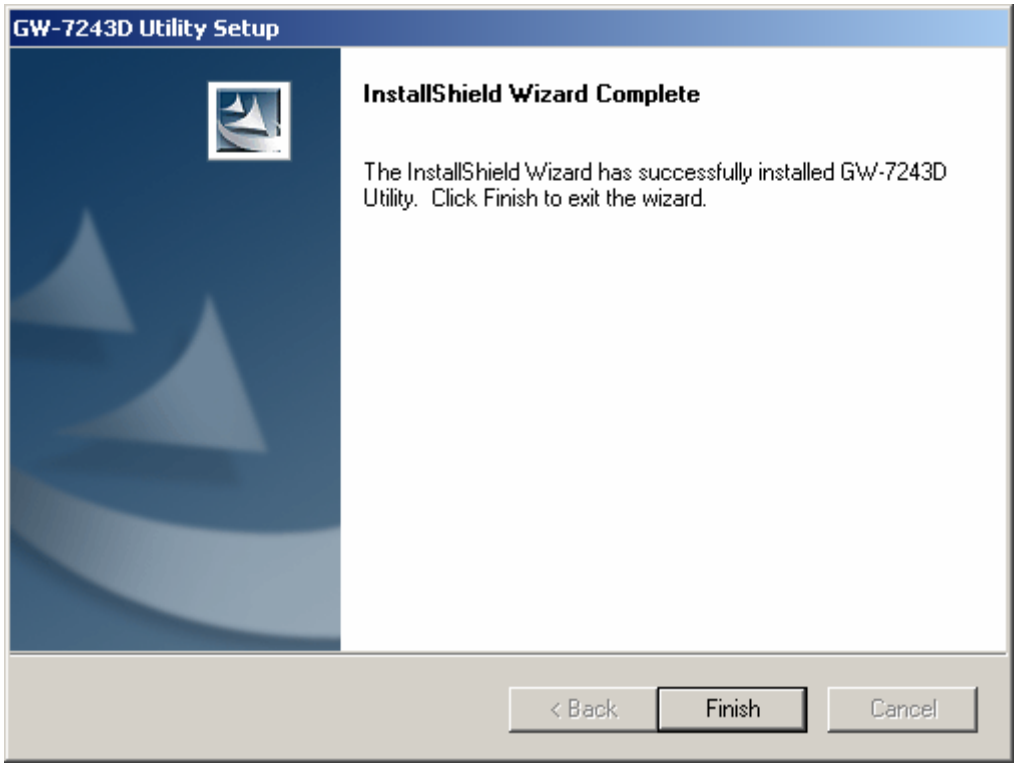


Step 5: A “Ready to Install the Program” is popped up. Click “Install” button.

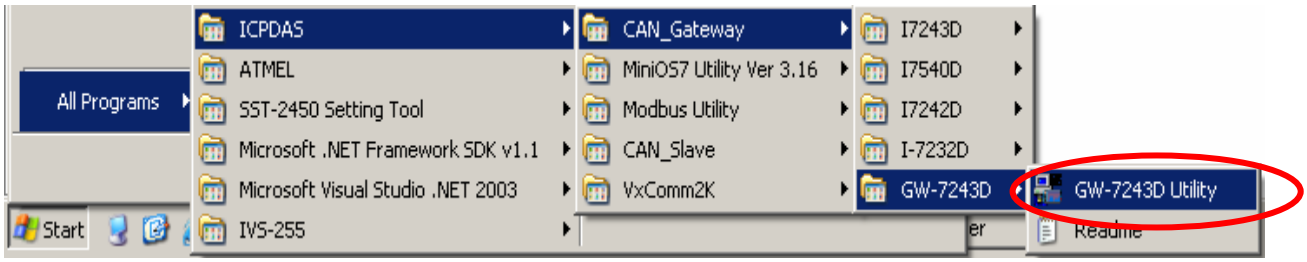


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Step 6: After finishing the process, a “Complete” window will pop up to prompt users that the successful completion of the installation. Then, click “Finish” button to exit.



Step 7: After finishing the installation of the GW-7243D Utility, users can find GW-7243D Utility in the “Start menu” as the following figure.



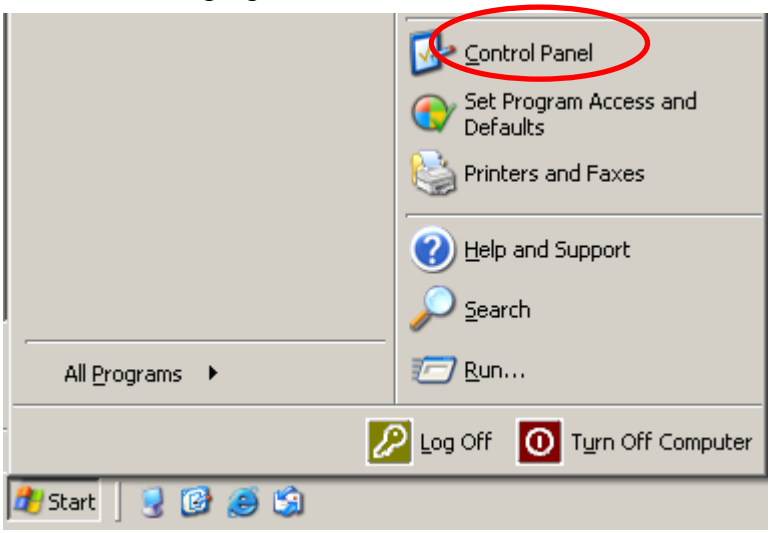
r

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

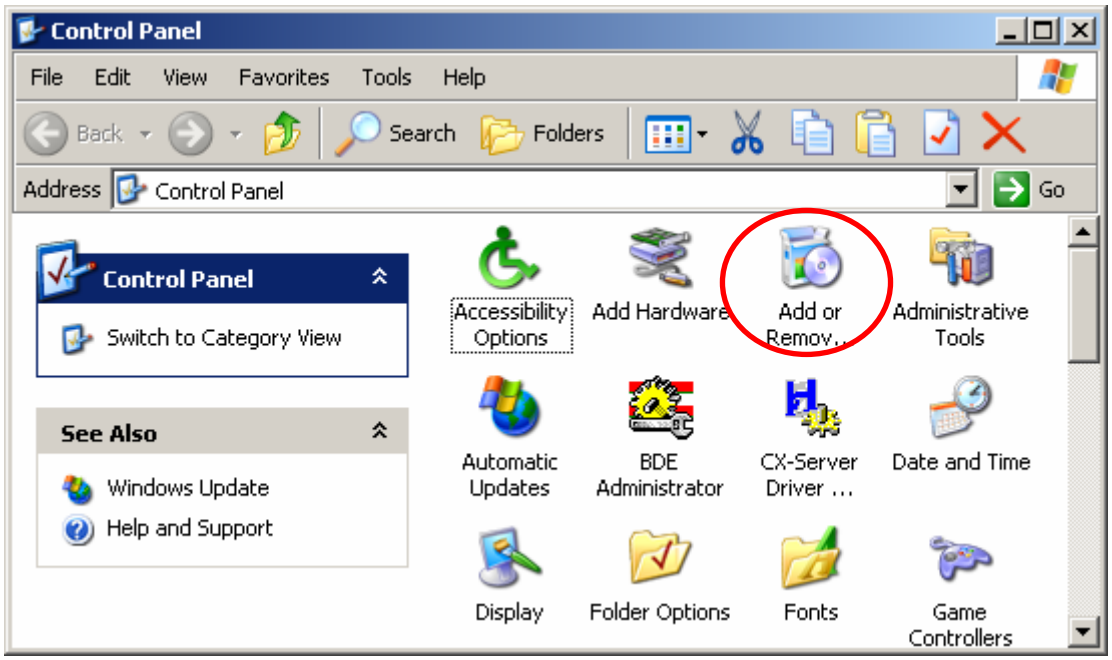
5.2.2 Uninstall GW-7243D Utility

You can uninstall the GW-7243D Utility software by the following steps:

Step 1: Click “Start” in the task bar, and select the “Settings/Control Panel” shown in following figure.

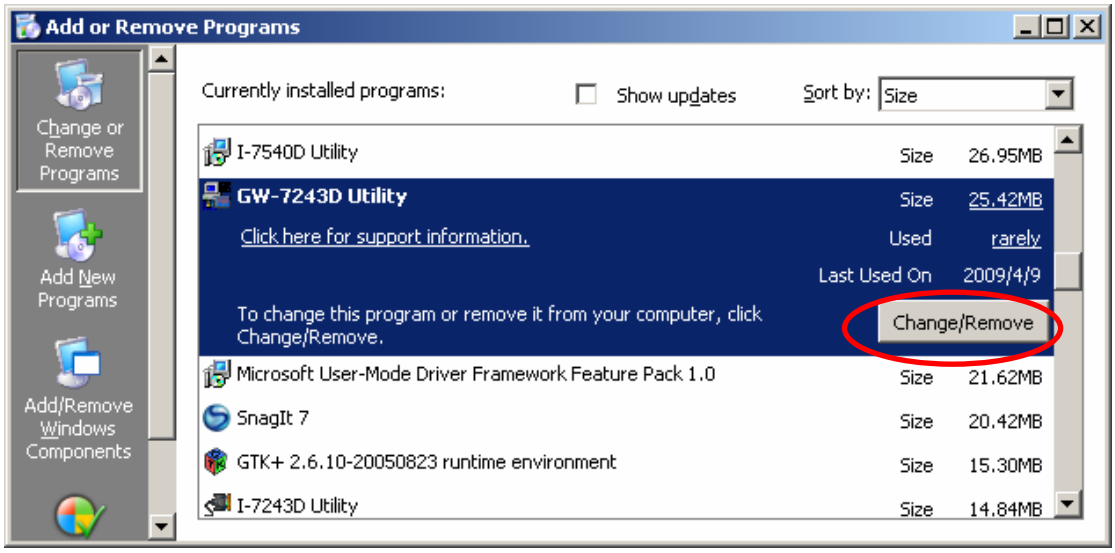


Step 2: Click the “Add/Remove Programs” button icon to open the dialog.

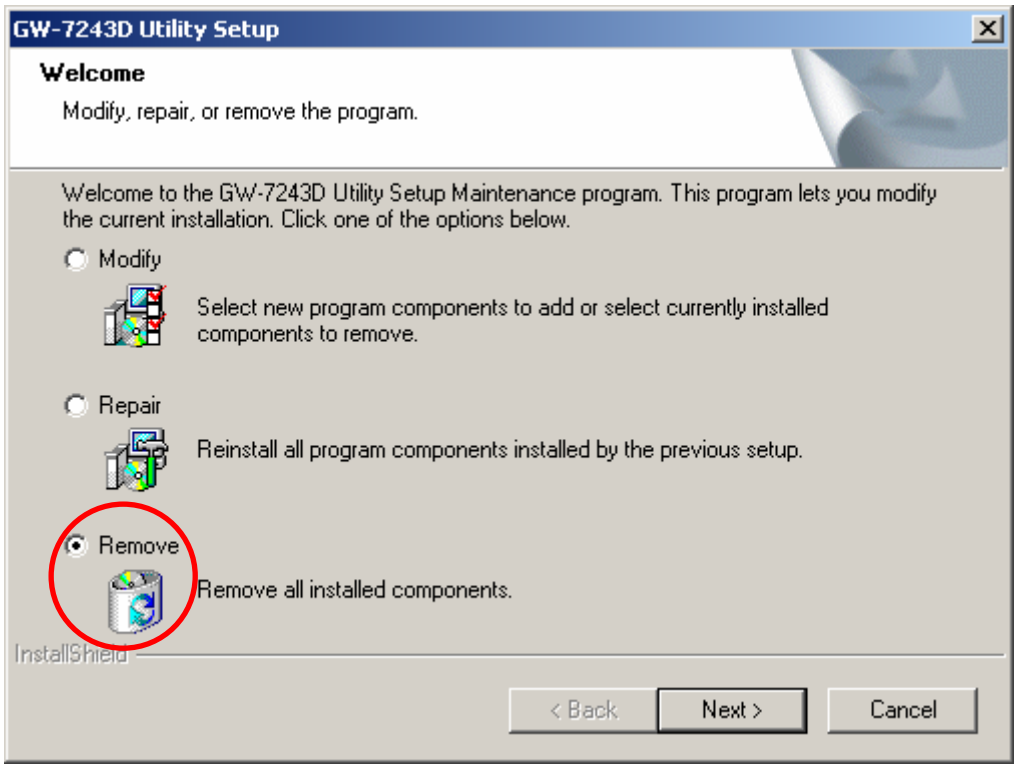


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Step 3: Find out the GW-7243D Utility, and click the Change/Remove button.

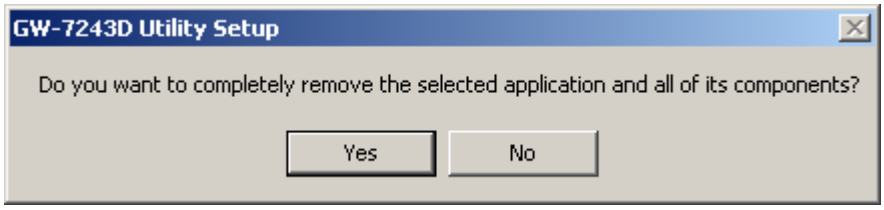


Step 4: Select the “Remove” option button, and press the “Next” button to remove GW-7243D Utility.

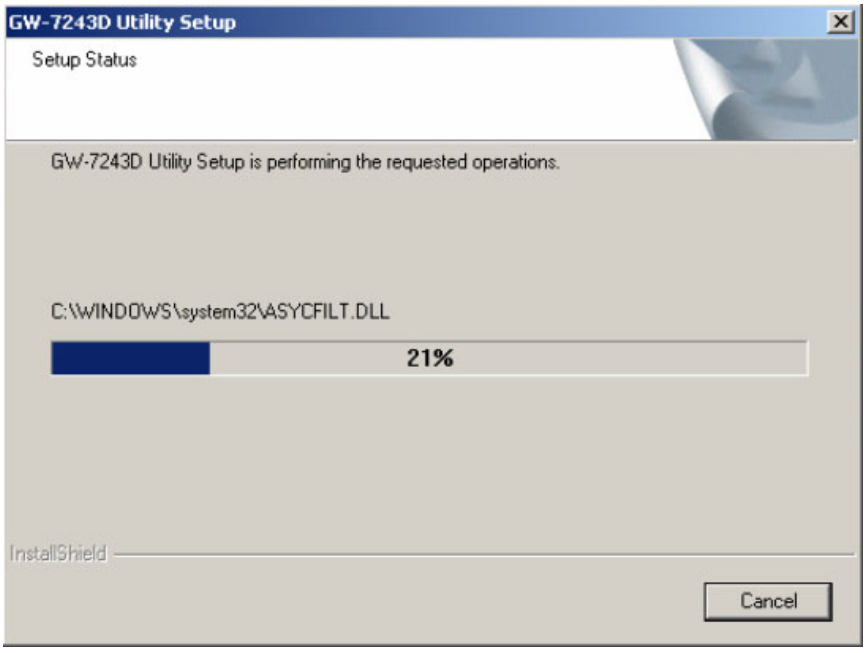


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

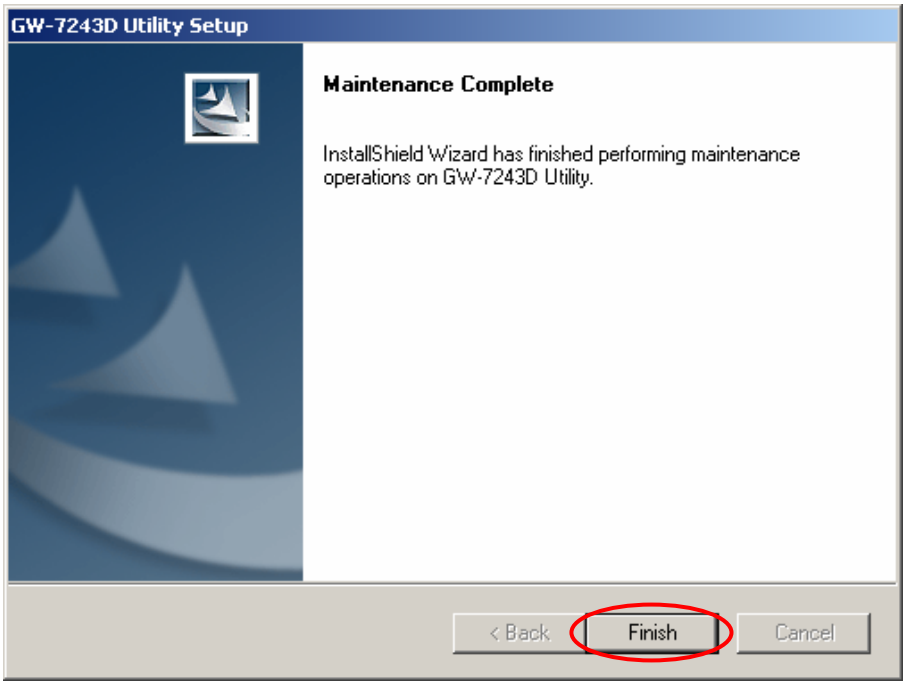
Step 5: Click the button “Yes” to remove the Utility tool.



Step 6: Removing GW-7243D Utility.



Step 7: Finally, click the “Finish” button to finish the uninstall process.

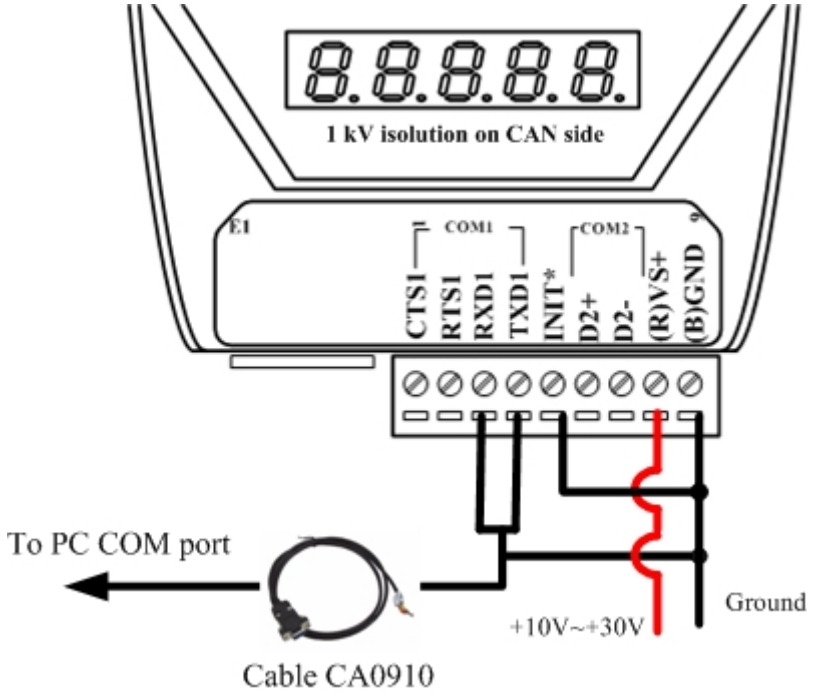


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

5.3 Configure GW-7243D by using Utility tool

5.3.1 Steps to configure the GW-7243D

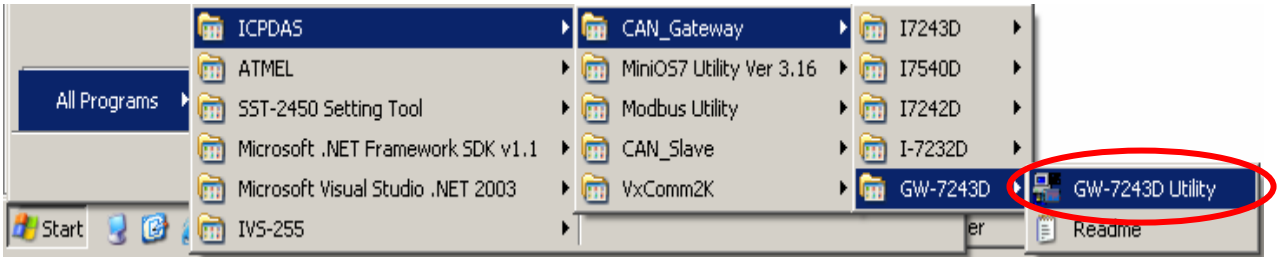
Step 1: Before configuring the GW-7243D, turn it off. Then connect the INIT* pin with the GND pin of the GW-7243D as following picture. Connect an available COM port of PC with the COM1 of GW-7243D. Then turn on the GW-7243D.



Step 2: Download the GW-7243D Utility setup file from the web site ftp://ftp.icpdas.com.tw/pub/cd/can_cd/devicenet/gateway/GW-7243D/utility/ or the CD-ROM disk following the path of "/ Fieldbus-CD / DeviceNet / Gateway / GW-7243D / Utility /

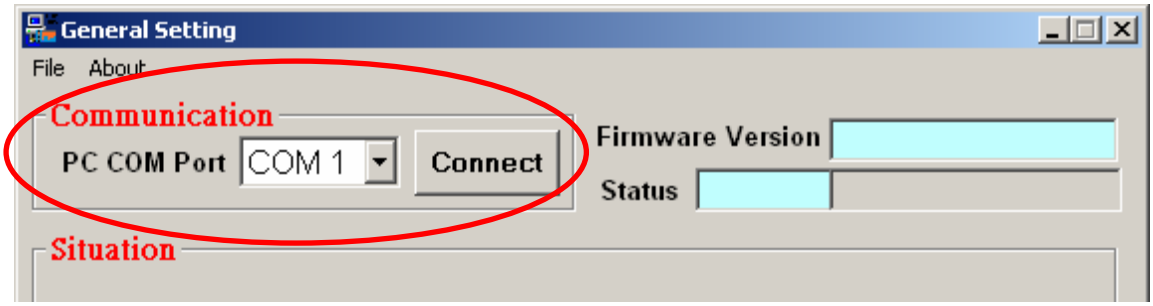
Step 3: Execute the setup.exe file to install GW-7243D Utility.

Step 4: After finishing the installation of the GW-7243D Utility, users can find GW-7243D Utility as shown in the following picture.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

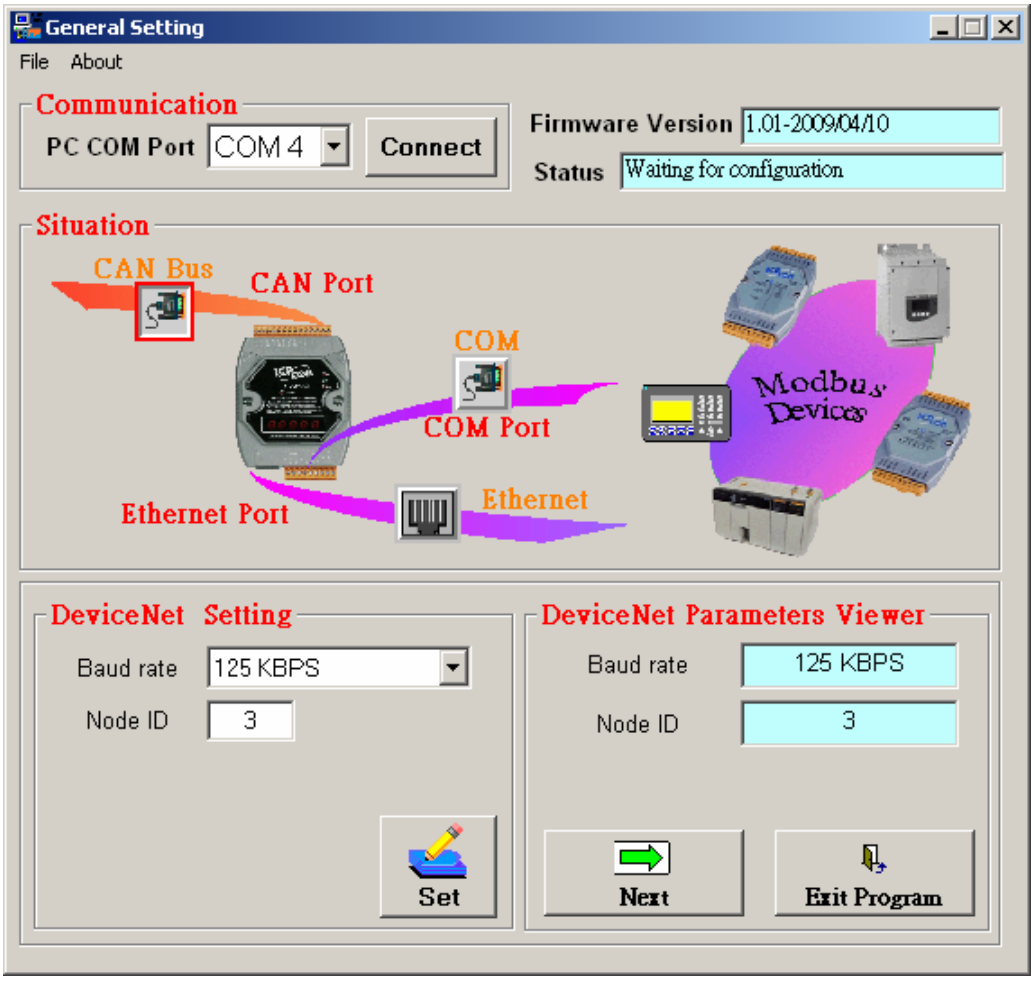
Step 5: Select the COM port of PC which is connected with the COM1 of GW-7243D, and then click the “Connect” button.



Step 6: If you get an error message during the connection, please check the wire connection described in Step 1. Then try the Step 5 again.

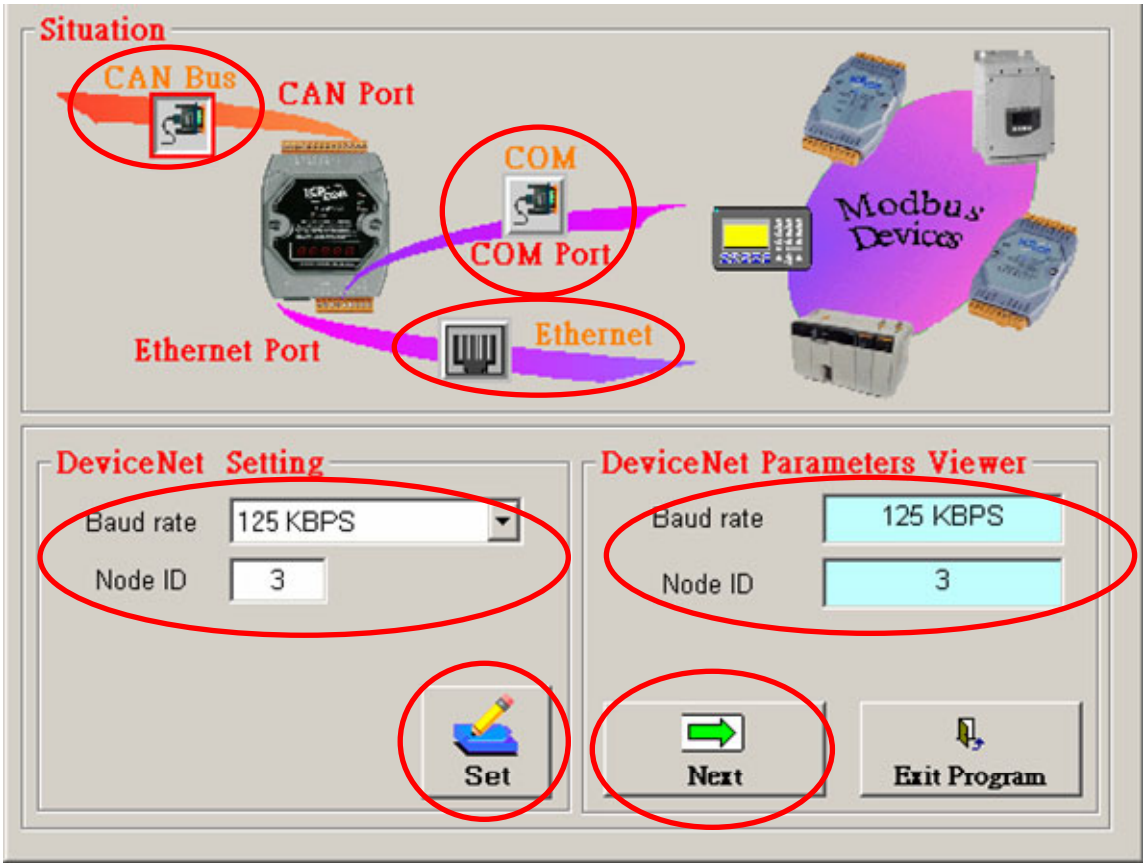


Step 7: If the connection is successful, the screen will be shown as follows.

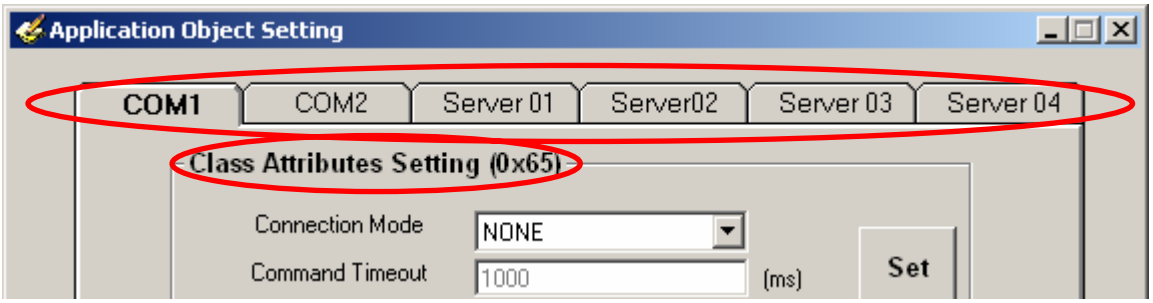


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Step 8: Click the icon and to configure the parameters of CAN bus (DeviceNet), COM port and Ethernet. For example, if you want to configure the CAN bus, click the icon of CAN bus. When you finish the configuration of DeviceNet on the frame of "DeviceNet Setting", click "Set" button. If configuration is successful, you can see the current configuration on the frame of "DeviceNet Parameters Viewer". After finishing all of the configuration, click "Next" button.



Step 9: There are 6 tabs for Modbus configuration. The COM1/COM2 tab is the configuration of Modbus RTU or Modbus ASCII. The Server01/ Server02/Server03/Server04 are the configuration of Modbus TCP. "Class Attributes Setting (0x65)" means that the configuration of COM1 will be saved in the class ID 0x65 of DeviceNet object. Select the tab which you want to configure and go on the next step.



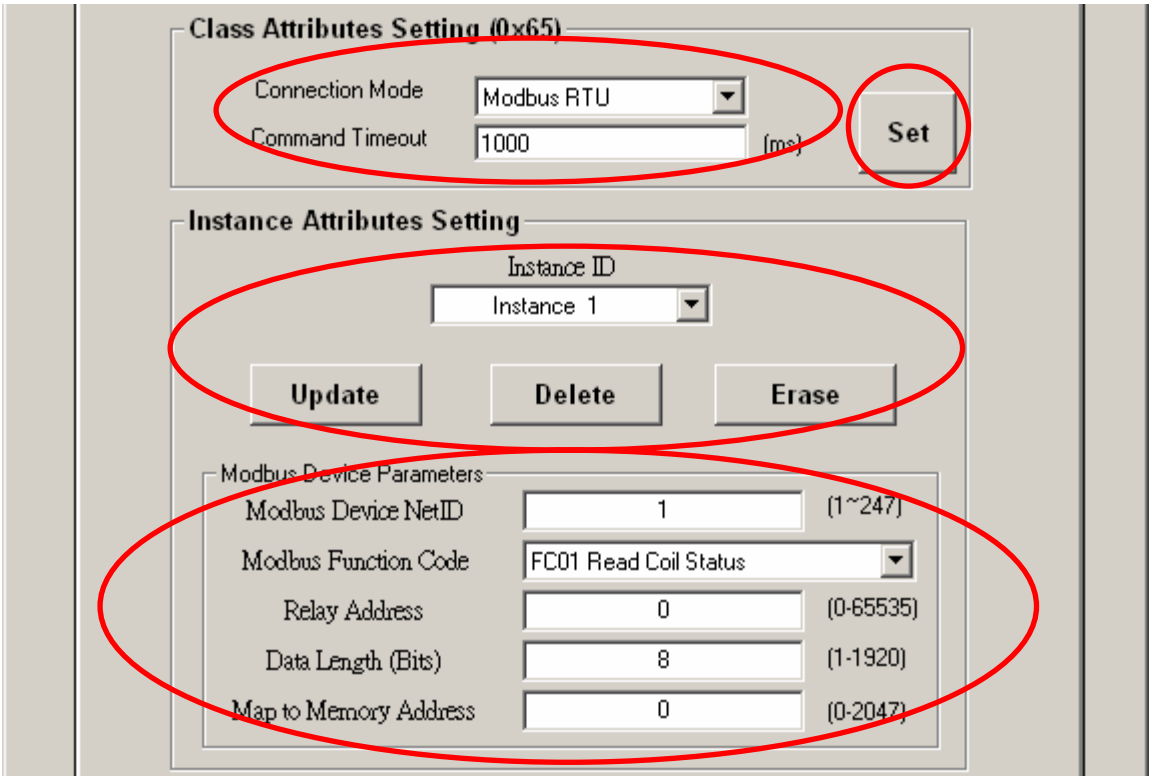
Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Step 10: Configure “connection mode” and “command timeout” for selected COM port. Select “None” of “connection mode” will disable this COM port. The “Command Timeout” is the response time of the Modbus slave when the GW-7243D queries this slave. When finishing the “Class Attribute Setting”, click “Set” button. In “Instance Attributes Setting” configuration, select “Instance ID” firstly. If the instance exists, users can use “Update”, “Delete” and “Erase” button. If not, only “Add” button can be used. Each instance has it own “Modbus Device Parameters”. The “Modbus Device NetID” is the station No. of the target Modbus slave. The “Modbus Function Code” is the command type which GW-7243D will execute. “Relay Address” and “Data Length (Bits)” are the start address and data length of reading or writing memory of Modbus slave. “Map to Memory Address” is the start channel of GW-7243D storage zone. GW-7243D assigns 6 independent memory blocks for 6 connections (COM1, COM2, Server01, Server02, Server03 and Server04). Each memory block can save 2048 DI channels, 2048 DO channels, 1024 AI channels, and 1024 AO channels. Users need to arrange the proper storage section for each channel in the same IO type. If the storage section of different “Modbus Device Parameters” is overlap, the data will be covered. For example, assume that users set two instances, and the configurations of “Modbus Device Parameters” are as follows:

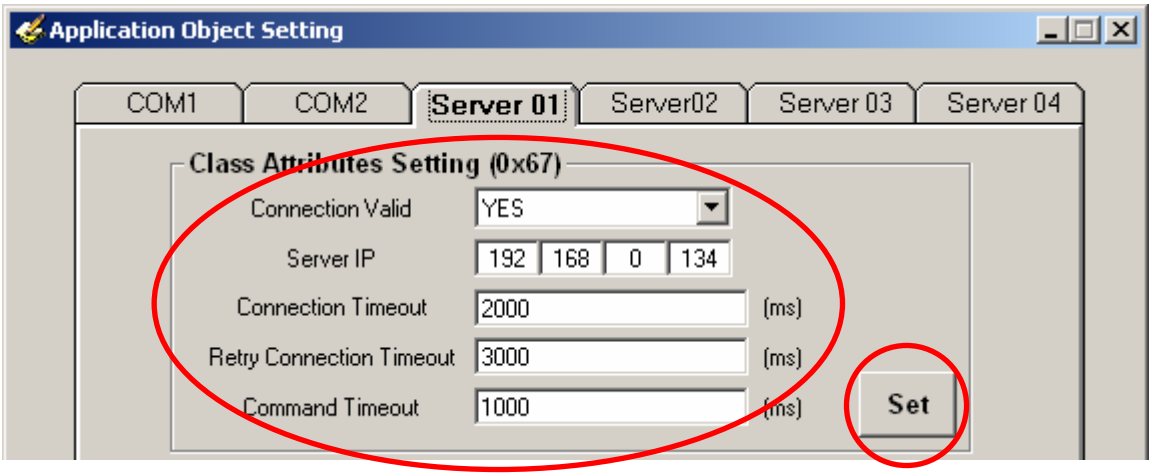
Instance ID	Modbus Device NetID	Modbus Function Code	Relay Address	Data Length (Bits)	Map to Memory Address
1	1	01	0	16	0
2	2	01	1	32	16

Instance 1 is set to get 16 DO data from the address 0 of Modbus slave with station No. 1. Instance 2 gets 32 DO data from the address 1 of Modbus slave 2. The “Map to Memory Address” of instance 1 is 0. It means that the data of instant 1 will be stored from channel 0 to channel 15 of GW-7243D memory. So, the start channel of memory for instance 2 must be set to 16. If it is set to 15, the last bit of data of instance 1 will be covered by the first bit of data of instance 2. Take a note that the data from Modbus Function code 0x03, 0x06, and 0x10 will be regard as the data of AO channels. The data from Modbus Function code 0x01, 0x05, and 0x0F will be regarded as the data of DO channels.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000



Step 11: The configuration step of the Modbus TCP is similar with Step10. "Connection Valid" can decide if this connection is active or not. "Server IP" is the IP address of the target Modbus slave. "Connection Timeout" is the timeout when GW-7243D built the TCP/IP connection with Modbus slave. "Retry Connection Timeout" is the time period which the GW-7243D reconnects to the Modbus slave if the connection is fail. "Command Timeout" is the response time of the Modbus slave when the GW-7243D queries this slave. After finishing the "Class Attributes Setting", click "Set" button. The configuration method of "Instance Attributes Setting" configuration is the same as COM port. Please refer to Step 10 for details.

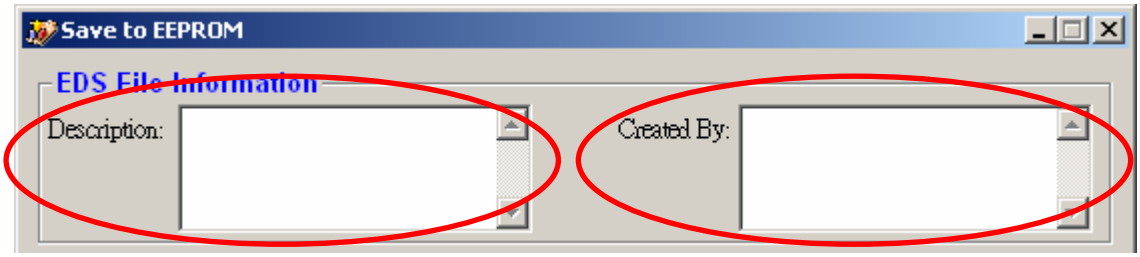


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

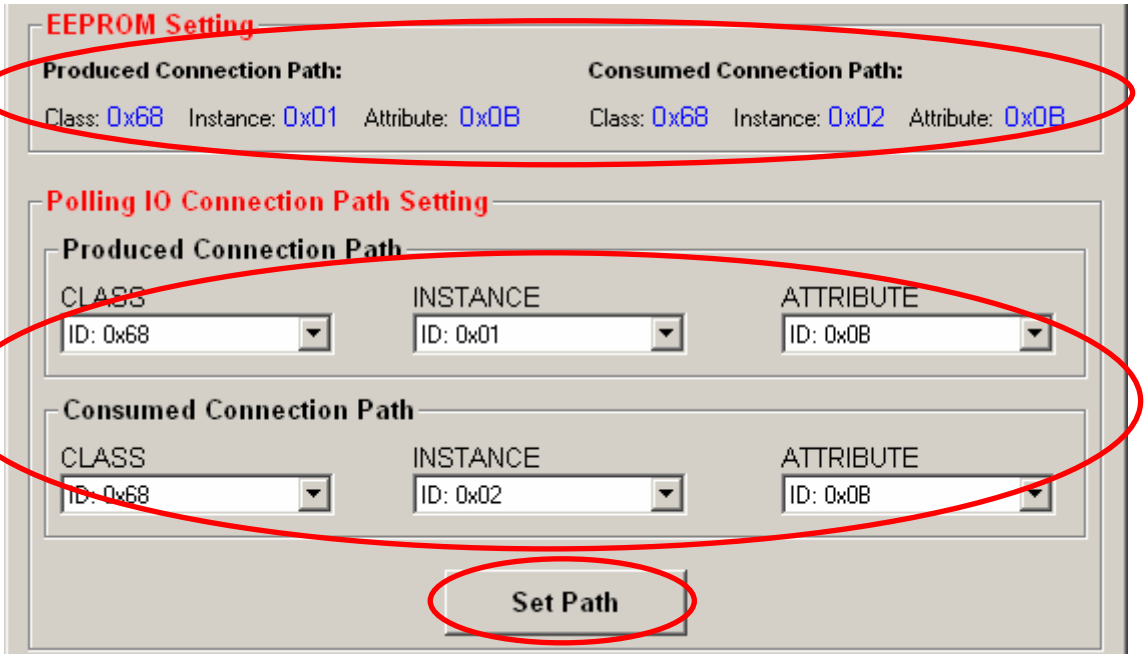
Step12: After finishing all configurations, users can check the configuration result by clicking the “Device View” button and “Object View” button. If everything is OK, click “Next” button to continue.



Step13: Fill the “Description” and “Create By” information of EDS file.



Step14: GW-7243D support Poll IO Connection. “EEPROM Setting” shows the current settings in the EEPROM of the GW-7243D. If you want to use Poll IO Connection, set the connection path in the frame of “Polling IO Connection Path Setting”. Or, just ignore this step. When you want to configure these parameters, select the “CLASS”, “INSTANCE”, and “ATTRIBUTE” from the corresponding combo boxes according to the following table. Only the instance which has been added in the Step 10 and Step 11 can be displayed in the combo box. The Class ID 0x64 is the default connection path. After finishing the settings, click “Set Path” button.



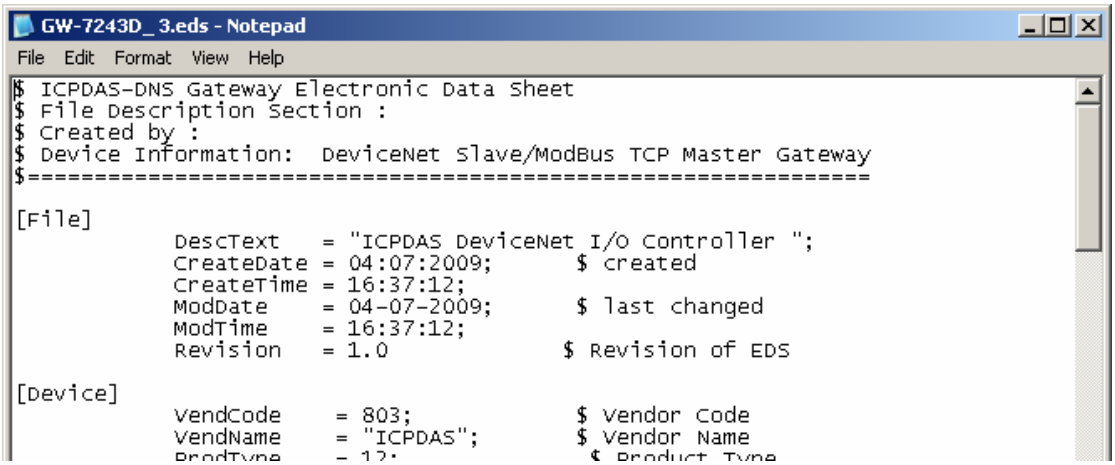
Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Class ID	Instance ID	Attrib. ID	Description	Method
0x64	0x01	0x01	GW-7243D IP	Get/Set
0x64	0x01	0x02	GW-7243D Gateway	Get/Set
0x64	0x01	0x03	GW-7243D Mask	Get/Set
0x65~0x66	0x01~0x0A	0x0A	Input Discrete Data	Get
0x67~0x6A	0x01~0x05	0x0A	Input Discrete Data	Get
0x65~0x66	0x01~0x0A	0x0B	Coils Status Data	Get/Set
0x67~0x6A	0x01~0x05	0x0B	Coils Status Data	Get/Set
0x65~0x66	0x01~0x0A	0x0C	Input Register Data	Get
0x67~0x6A	0x01~0x05	0x0C	Input Register Data	Get
0x65~0x66	0x01~0x0A	0x0D	Registers Data	Get/Set
0x67~0x6A	0x01~0x05	0x0D	Registers Data	Get/Set

Step14: Afterwards, you can click “Object View” to check all of the configurations set before. If you want to clear the all configurations of the GW-7243D, click “Factory Setting” button. Click “Save” button to save all of the configurations into the EEPROM of the GW-7243D and finish the configurations.



Step 15: Use the EDS file of the GW-7243D in the DeviceNet application. You can find it in the same folder of GW-7243D utility. The default path is “C:\ICPDAS\CAN_Gateway\GW-7243D\”. The file name may be GW-7243D_3.eds. “_3” indicates the DeviceNet Node ID of the GW-7243D that you set in the Step 8.



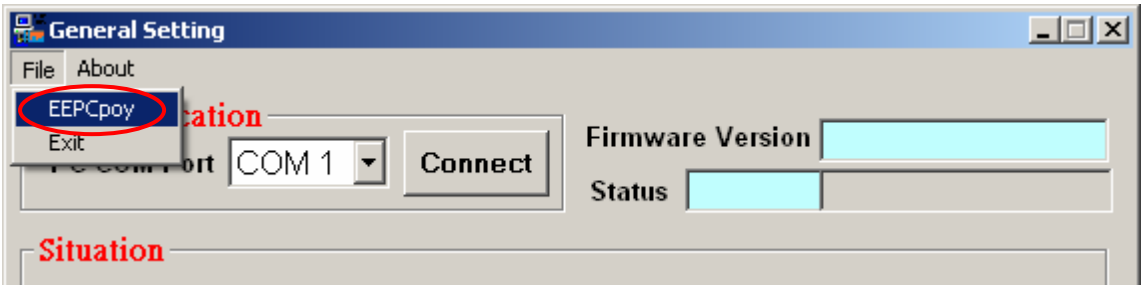
Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

5.3.2 Import/export all configuration to/from EEPROM

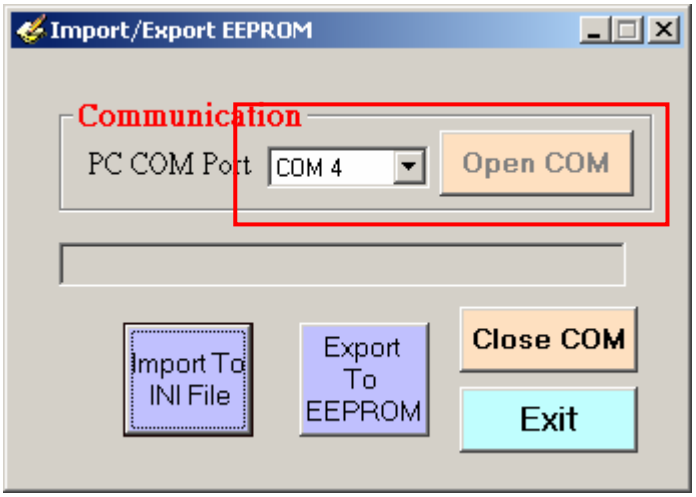
Users can import all the configuration of the GW-7243D into an .ini file or export the data from the .ini file to another GW-7243D via the GW -7243D Utility. The following paragraph shows how to do that step by step.

Step1: Turn off the power of the GW-7243D and switch the jump in the back plane of it into initial mode. Then turn on the power of the GW-7243D. Then, connect the COM1 of GW-7243D with PC available COM port.

Step2: Press the EEPCopy button.

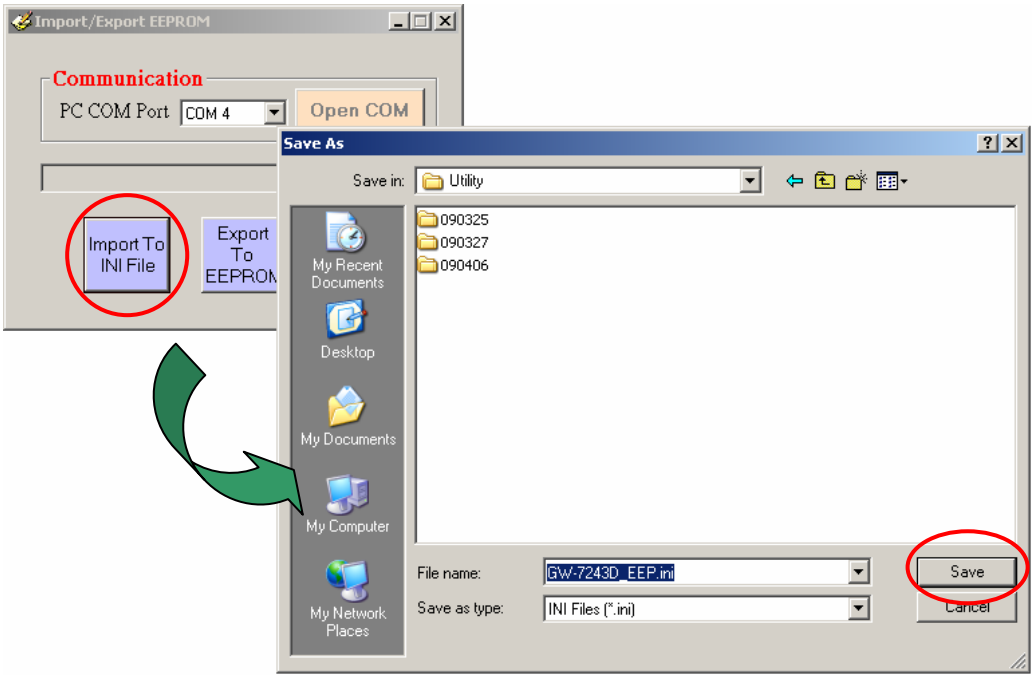


Step3: Select the COM port of the PC which is connected to the COM 1 of the GW-7243D in the step 1 and press the “Open COM” button to open the COM port.

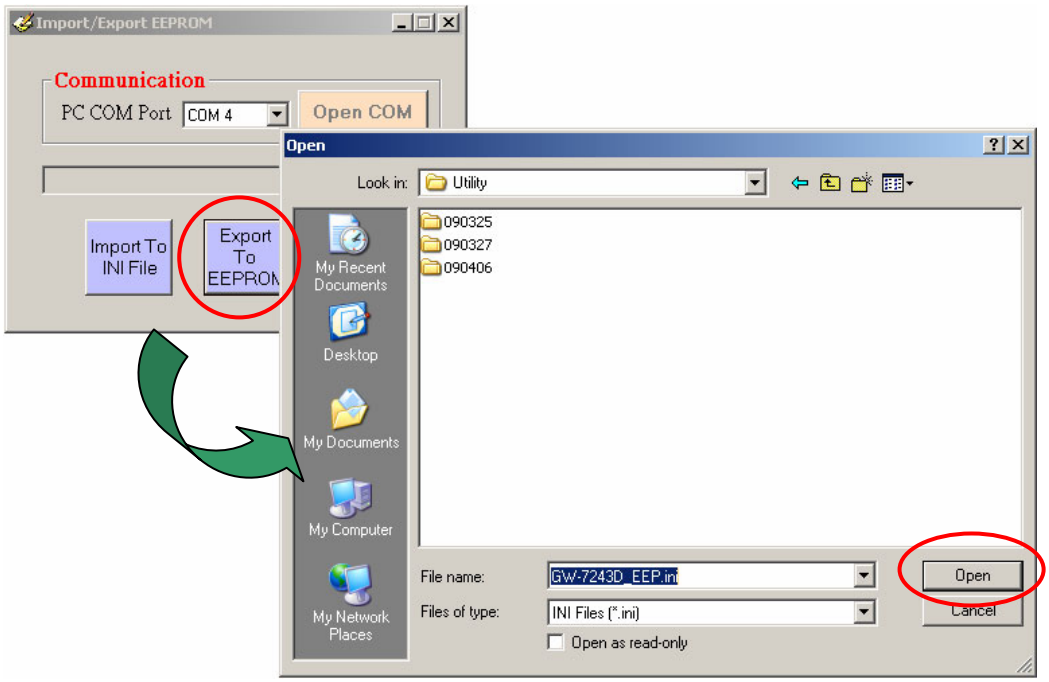


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Step4: Then users can press the “import To INI file” button to import all the EEPROM data of the GW-7243D into the selected .ini file.



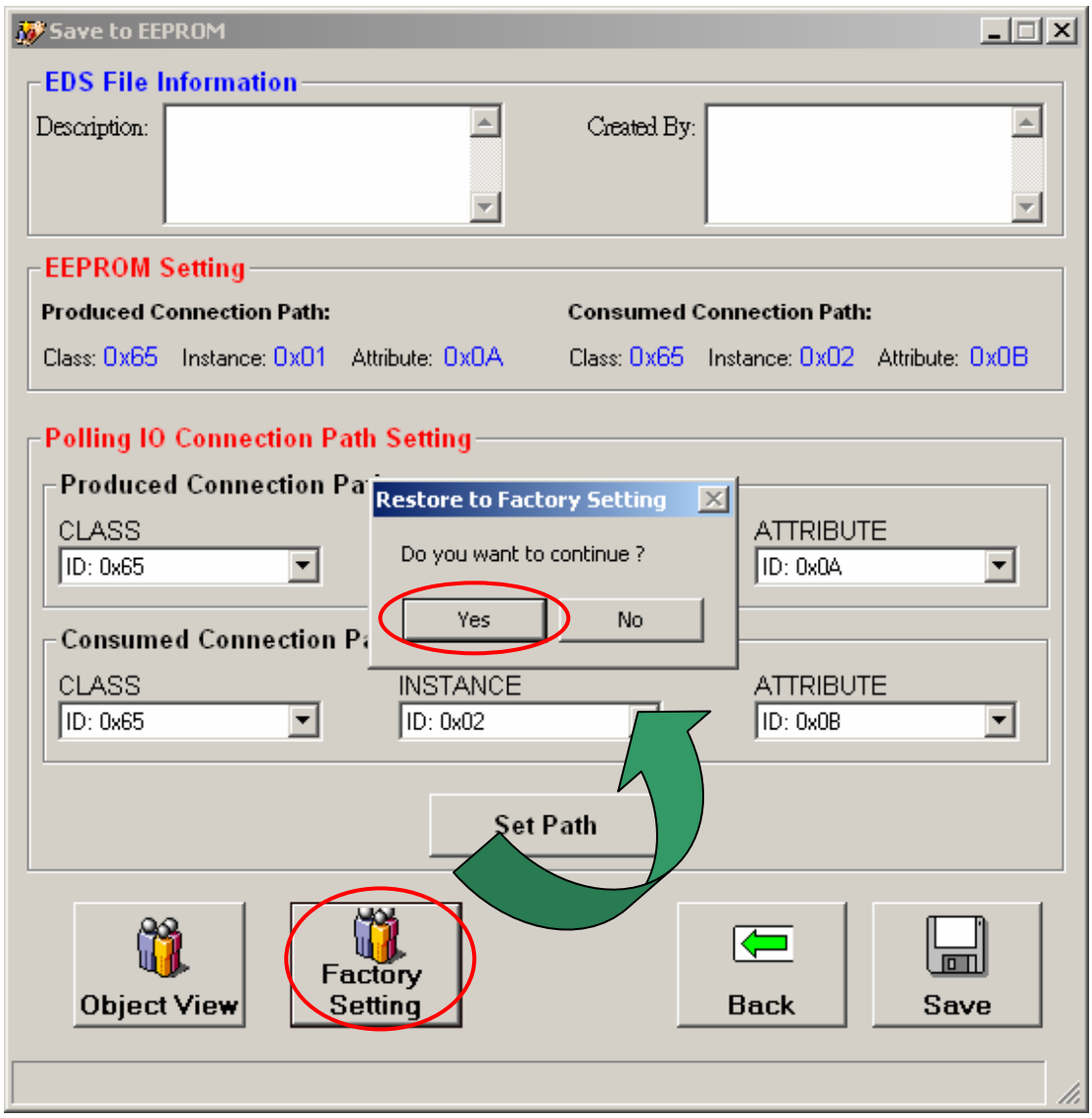
Step5: Or, users can press the “Export to EEPROM” button to export all the data from the selected .ini file into the EEPORM of the GW-7243D.



LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

5.3.3 Restore to Factory Setting

The GW-7243D support “Restore to Factory” function for user to recover the module to the factory default. After using this function and rebooting the GW-7243D, all of the configuration will restore to the factory default. Users can find this configuration page in the GW-7243D Utility as follows.



Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6. Modbus Commands

Only these Modbus function codes shown in the table 6-1 are supported by the GW-7243D. The structure of the query and response frames for each of these commands is then described in the following section.

Table 6-1 Support commands of GW-7243D

Function Code	Modbus Command
0x01	Read Coil Status
0x02	Read Input Status
0x03	Read Holding Registers
0x04	Read Input Registers
0x05	Write Single Coil
0x06	Write Single Register
0x0F	Force Multiple Coils
0x10	Preset Multiple Registers

In the table 6-2, each byte of the query and response frames of the Modbus command are described with the excepted of the field shown opposite. These are always present in the queries and responses of all Modbus commands.

Table 6-2 Frames of a Modbus command

Device address	Value cannot be changed (Valid Modbus device address: 1 to 247.)
Function code	Value can not be changed (code of the Modbus command)
… Other fields …	… Specific features of Modbus commands …
CRC (LSB)	Cyclic Redundancy Check, containing 16-bit binary value
CRC (MSB)	

It’s a better idea to get hold of a standard Modbus document, such as the guide entitled Modicon Modbus Protocol Reference Guide, so that you can see the correspondence between the elements displayed in Utility and the content of the corresponding Modbus frames

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Here is an example of correspondences for a full frame of Modbus RTU command, based on the “Read Holding Registers” Command (0x03).

	Element under Utility		**Modbus frame fields**	**Size (byte)**
Modbus query	Modbus Devices Address		Device no.	1
	Device I/O Type		Function no.	1
	Register Address		No. 1st word (MSB/LSB)	2
	Communication Words		No. of words (MSB/LSB)	2

	Element under Gateway		**Modbus frame fields**	**Size (byte)**
Modbus response	Modbus Devices Address		Device no.	1
	Device I/O Type		Function no.	1
	Byte count		No. of bytes read	1
	Data		Value of 1st word (MSB/LSB)	2
			…….	…
			Value of last word (MSB/LSB)	2

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6.1 “Read Coil Status” Command (0x01)

Read the On/Off status of discrete output in the slave. The query message specifies the starting coil and quantity of coils to be read. And the coil status in the response message is packed as one coil per bit of the data field.

Frame	Field	Description
Query	Starting Address (Hi)	Address of 1st output coil
	Starting Address (Lo)	
	No. of Points (Hi)	Number of output coils
	No. of Points (Lo)	
Response	Byte Count	Number of data bytes=(number of output coils + 7) / 8
	Data (Coils: H-L order)(L)	Byte swap=”Swap 1 byte” (or “No swapping”) Data length=Value of the “Byte Count” field Data location=Address in the gateway’s “DO” memory
	…..	
	Data (Coils: H-L order)(H)	

6.2 “Read Input Status” Command (0x02)

Read the On/Off status of discrete input in the slave. The query message specifies the starting input and quantity of inputs to be read. And the input status in the response message is packed as one input per bit of the data field.

Frame	Field	Description
Query	Starting Address (Hi)	Address of 1st input coil
	Starting Address (Lo)	
	No. of Points (Hi)	Number of input coils
	No. of Points (Lo)	
Response	Byte Count	Number of data bytes=(number of input coils + 7) / 8
	Data (Inputs: H-L order)(L)	Byte swap=”Swap 1 byte” (or “No swapping”) Data length=Value of the “Byte Count” field Data location=Address in the gateway’s “DI” memory
	…..	
	Data (Inputs: H-L order)(H)	

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6.3 “Read Holding Registers” Command (0x03)

Read the binary content of holding registers in the slave. The query message specifies the starting register and quantity of registers to be read. And the register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte.

Frame	Field	Description
Query	Starting Address (Hi)	Address of 1st output register
	Starting Address (Lo)	
	No. of Points (Hi)	Number of output registers
	No. of Points (Lo)	
Response	Byte Count	Number of data bytes=number of output registers x 2
	Data (first register/MSB)	Byte swap=”Swap 2 bytes” (or “No swapping”) Data length=Value of the “Byte Count” field Data location=Address in the gateway’s “AO” memory
	Data (first register/LSB)	
	…..	
	Data (last register/MSB)	
	Data (last register/LSB)	

6.4 “Read Input Registers” Command (0x04)

Read the binary content of input registers in the slave. The query message specifies the starting register and quantity of registers to be read. And the register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte.

Frame	Field	Description
Query	Starting Address (Hi)	Address of 1st input register
	Starting Address (Lo)	
	No. of Points (Hi)	Number of input registers
	No. of Points (Lo)	
Response	Byte Count	Number of data bytes=number of input registers x 2
	Data (first register/MSB)	Byte swap=”Swap 2 bytes” (or “No swapping”) Data length=Value of the “Byte Count” field Data location=Address in the gateway’s “AI” memory
	Data (first register/LSB)	
	…..	
	Data (last register/MSB)	
	Data (last register/LSB)	

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6.5 “Force Single Coil” Command (0x05)

Force a single coil to either ON or OFF. The query message specifies the coil reference to be forced.

Frame	Field	Description
Query	Coil Address (Hi)	Address of 1st output coil
	Coil Address (Lo)	
	Force Data (Coil)	=FF to trun ON coil, =00 to trun OFF coil
	Force Data	=00
Response	Coil Address (MSB)	Address of 1st output coil
	Coil Address (LSB)	
	Coil Data	=FF to turn ON coil, =00 to turn OFF coil
	Coil Data	=00

6.6 “Write Single Register” Command (0x06)

Preset a value into a single holding register. The query message specifies the register reference to be preset.

Frame	Field	Description
Query	Starting Address (Hi)	Address of output register
	Starting Address (Lo)	
	Register value (Hi)	Output register data
	Register value (Lo)	
Response	Starting Address (Hi)	Address of output register
	Starting Address (Lo)	
	Register value (Hi)	Output register data
	Register value (Lo)	

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6.7 “Force Multiple Coils” Command (0x0F)

Forces each coil in a sequence of coils to either ON or OFF. The query message specifies the coil references to be forced. The normal response returns the slave address, function code, starting address, and quantity of registers preset.

Frame	Field	Description
Query	Coil Address (Hi)	Address of 1st output coil
	Coil Address (Lo)	
	Quantity of Coils (Hi)	Number of output coils
	Quantity of Coils (Lo)	
	Byte Count	Number of data bytes=(number of output coils + 7) / 8
	Force Data (Coils: H-L)(L)	Byte swap=”Swap 1 byte”
	…..	Data length=Value of the “Byte Count” field
	Force Data (Coils: H-L)(H)	Data location=Address in the gateway’s “DO” memory
Response	Coil Address (MSB)	Address of 1st output coil
	Coil Address (LSB)	
	Quantity of Coils (MSB)	Number of output coils
	Quantity of Coils (LSB)	

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6.8 “Preset Multiple Registers” Command (0x10)

Preset values into sequence of holding registers. The query message specifies the register reference to be preset. The normal response returns the slave address, function code, starting address, and quantity of register preset.

Frame	Field	Description
Query	Starting Address (Hi)	Address of 1st output register
	Starting Address (Lo)	
	No. of Registers (Hi)	Number of output registers
	No. of Registers (Lo)	
	Byte Count	Number of data bytes=number of output registers x 2
	Data (first register/MSB)	Byte swap=”Swap 2 bytes” Data length=Value of the “Byte Count” field Data location=Address in the gateway’s “AO” memory
	Data (first register/LSB)	
	…..	
	Data (last register/MSB)	
	Data (last register/LSB)	
Response	Starting Address (Hi)	Address of 1st output register
	Starting Address (Lo)	
	No. of Registers (Hi)	Number of output registers
	No. of Registers (Lo)	

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

6.9 Exception Responses

When a slave receives the query without a communication error, but cannot handle it, the slave will return an exception response informing the master of the nature of the error. The structure of an exception response is independent of the Modbus command associated with the “Function” field of the query involved. The whole frame of an exception response is shown below, depended on Modbus devices.

Slave Address	Modbus address (1 to 247). The value of this field is identical to that of the “Slave Address” field of the query involved.
Function	The value of this field is set to 0x80 + the value of the “Function” field of the query involved.
Exception Code	Code indicating the nature of the error which has caused the exception response
Checksum (Lo)	Error check
Checksum (Hi)	

Code	Name	Meaning
0x01	Illegal Function	The function code received in the query is not an allowable action for the slave.
0x02	Illegal Data Address	The data address received in the query is not an allowable address for the slave.
0x03	Illegal Data Value	A value contained in the query data field is not an allowable value for the slave.
0x04	Slave Device Failure	An unrecoverable error occurred while the slave was attempting to perform the requested action.
0x05	Acknowledge	The slave has accepted the request and is processing it, but a long duration of time will be required to do so. The gateway should transmit subsequent queries in order to determine whether the request has finished or not.
0x06	Slave Device Busy	The slave is engaged in processing a long–duration program command. So the gateway should re-transmit the query subsequently.
0x07	Negative Acknowledge	The slave cannot perform the program function received in the query. This exception only affects commands 0x0D and 0x0E.
0x08	Memory Parity Error	The slave attempted to read extended memory, but detected a parity error in the memory. This exception only affects standard commands 0x14 and 0x15.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Appendix A: General Status Table

Modbus Type	Status value (Hex)	Description
Modbus TCP	0x00	No Error
	0x01 | 0x7F	Modbus Exception Code
	0x80	Connection of Modbus TCP server not initial
	0x81	Wait to re-connect with Modbus TCP server
	0x82	Building connection with Modbus tCP server
	0x83	Connect to Modbus TCP server Fail
	0x90	Send command to Modbus TCP server fail
	0x91	Send command OK, response command timeout
	0x92	Send command OK, response command error

Modbus RTU	0x00	No Error
	0x01 | 0x7F	Modbus Exception Code
	0x91	Send command OK, response command timeout
	0x93	Check response command CRC error
	0x95	Response command blank timeout
	0x96	Response command data length error

Modbus ASCII	0x00	No Error
	0x01 | 0x7F	Modbus Exception Code
	0x91	Send command OK, response command timeout

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

	0x94	Check response command CRC error
	0x95	Response command blank timeout
	0x96	Response command data length error

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000