i-7241D
DeviceNet slave/DCON 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 2004 by ICP DAS Co., LTD. All rights reserved worldwide.

Trademark

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

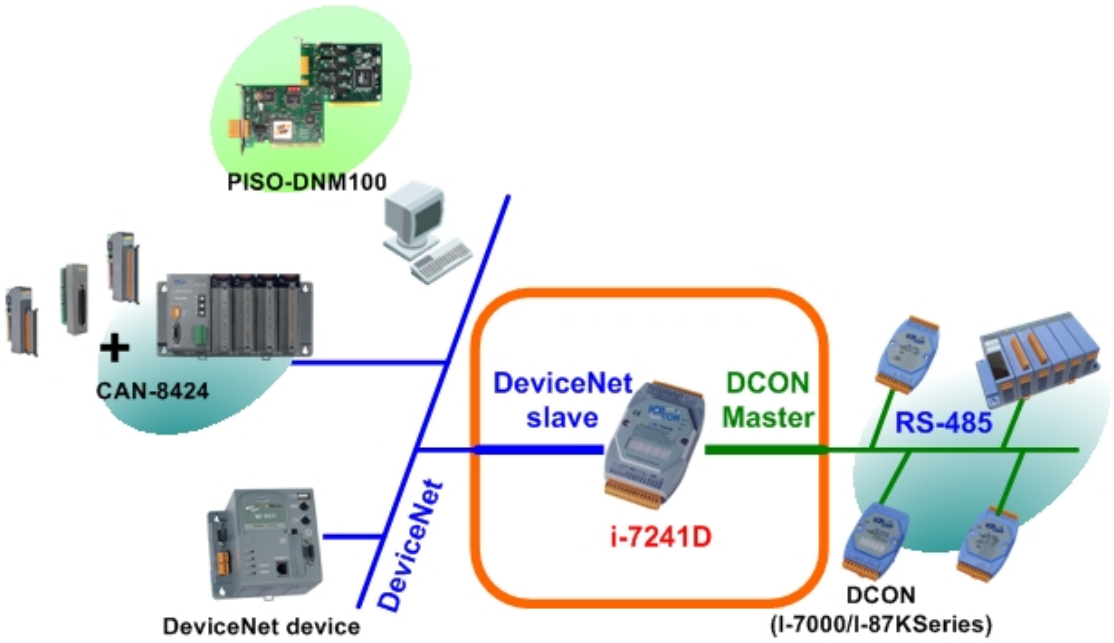
Table of Contents

Chapter 1 Introduction

1.1 Overview

The i-7241D is one of the CAN series products produced by ICP DAS. The product provides a communication protocol transfer the DeviceNet and the DCON protocol. DCON protocol is the communication protocol for the i-7K and i-87K series IO modules developed by ICP DAS. The i-7241D can be a DeviceNet slave device in the CAN bus on the DeviceNet network. It is a Group 2 Only Slave device, and supports the “Predefined Master/slave Connection Set”. In addition, we also provide utility software for users to configure their device parameters and build EDS file for the i-7241D. Users can easily apply i-7K and i-87K IO modules into DeviceNet applications through the i-7241D.

The application architecture is depicted below. Users can connect the DCON network to the DeviceNet network via the i-7241D.



Application architecture

1.2 Hardware Features

- CPU: 80188, 40MHz
- Philip SJA1000 CAN controller
- Philip 82C250 CAN transceiver
- SRAM: 512K bytes
- Flash Memory: 512K bytes
- EEPROM:2k bytes
- Real Time Clock
- Built-in Dual-Watchdog
- 16-bit Timer
- 2500 Vrms isolation on CAN side
- Power Consumption: 2.8 W
- Unregulated +10VDC to +30VDC
- Operating Temperature: -25°C to +75°C
- Storage Temperature: -30°C to +85°C
- Humidity: 5%~95%
- NS, MS and IO Led directors

COM1

- RS-232
- RS-232: TXD, RXD, RTS, CTS, GND
- Communication speed: 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 bps.
- Configure tool connection

COM2

- RS-485: D2+, D2-
- Communication speed: 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 bps.
- Connect to DCON IO modules

Display

- 7-segmemt LED to show operation mode, Node ID, CAN baud and RS-485 baud

1.3 i-7241D Features

- Group 2 Only Slave
- Support of Predefined Master/slave Connection Set
- Dynamic Assembly Objects Mapping
- Support of Offline Connection Set, Device Heartbeat message and Device Shutdown message
- Dynamic Product EDS File
- I/O operating modes: Polling, Bit-Strobe, Change of State/Cyclic
- Support max 15 i-7K/i-87K IO modules
- Auto scan the input channel situations from the DCON modules
- Support the watchdog function of i-7K/87K I/O series modules
- Baud Rate: 125Kbaud, 250Kbaud and 500Kbaud
- Provide friendly Utility to configure
- On-line change baud rate and MAC ID of CAN
- NS, MS and IO LED indicators
- 7-segmemt LED to show operation mode, MAC ID, baud rate and error code

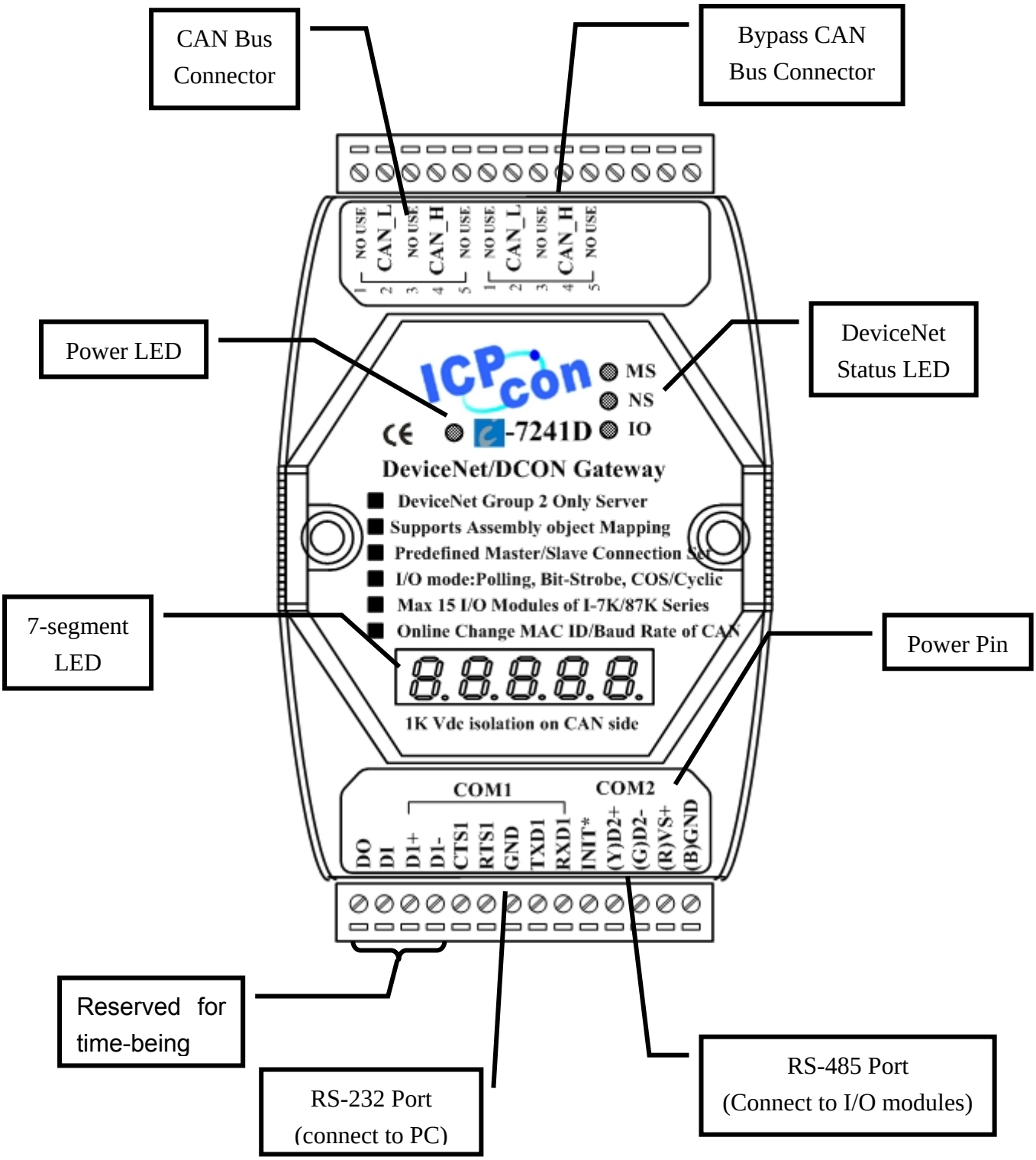
1.4 Utility Features

- Support DeviceNet node ID, baud rate setting, and com port parameters setting
- Support auto scan i-7K/i-87K modules
- Show i-7K/i-87K modules configuration
- Show Application and assembly objects configuration
- Support IO connection path setting
- Support EDS file creating

Please refer to Appendix A to know how to mount i-7241D

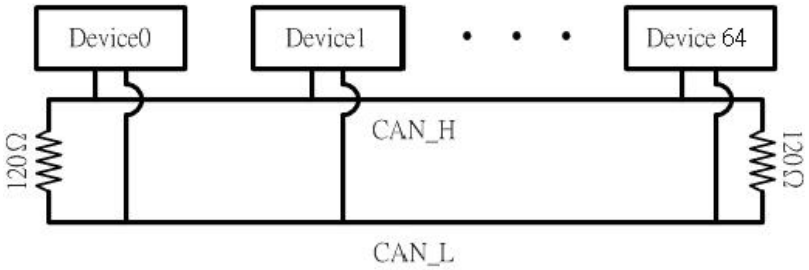
Chapter 2 Hardware Specification

2.1 Hardware Structure



2.2 Wire Connection

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



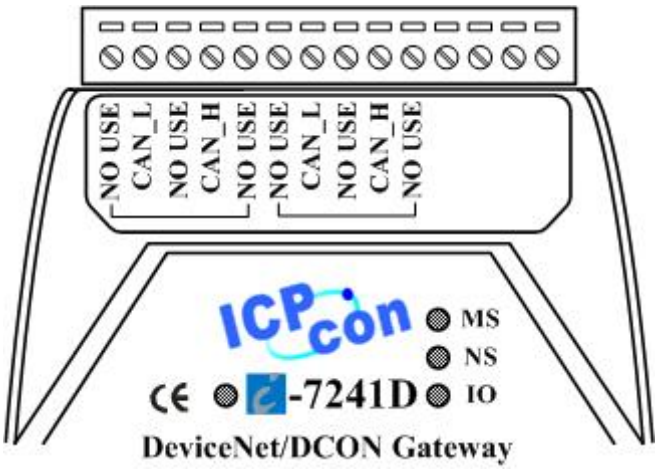
Moreover, to minimize the voltage drop on long distance, the terminal resistance should be higher than the value defined in the ISO 11898-2. The following table could be a reference.

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.mm^2 (18AWG)	150~300

The CAN bus bard rate has the high relationship with the bus length. The following table indicates the corresponding bus length on every kind of baud rate in DeviceNet protocol.

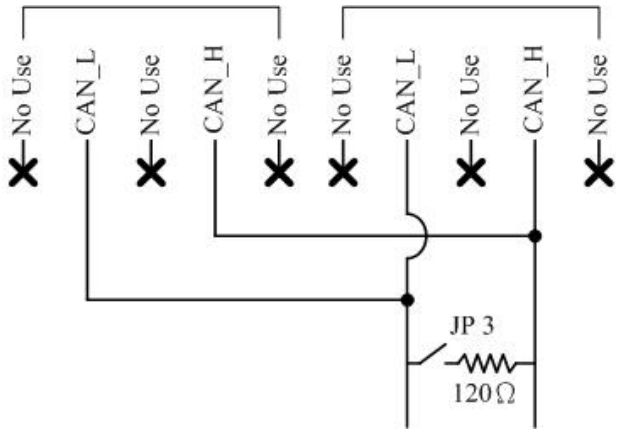
Baud rate (bit/s)	Max. Bus length (m)
500 K	100
250 K	250
125 K	500

In order to wiring conveniently, the i-7241D supplies two CAN bus connector. Each connecter built on the i-7241D looks like as the following figure.



Pin No.	Signal	Description
2	CAN_L	CAN_L bus line (dominant low)
3	CAN_SHLD	Optional CAN Shield
4	CAN_H	CAN_H bus line (dominant high)

Be Careful that the bypass CAN bus connector is not another CAN channel. It is designed for connecting to another DeviceNet device conveniently. The structure of the inside electronic circuit is displayed as follows.



2.3 Power LED

The i-7241D needs 10~30 VDC power input and consumes 2.8W. The Power LED will be turn on after applying power.

2.4 DeviceNet Status LED

i-7241D provides three DeviceNet LED indicators. 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.

2.4.1 MS LED

This LED provides device status. It indicates whether or not the device is operating properly.

condition	status	indicates
Solid 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

2.4.2 NS LED

This LED indicates the status of the communication link.

condition	status	indicates
Off	Off line	DeviceNet is off line
Flashing green	On line	DeviceNet is on line, but not communicating
Init solid green	Link failed	(critical) 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

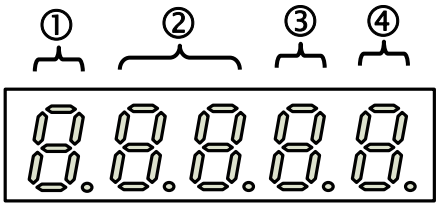
2.4.3 IO LED

This LED provides information of inputs and/or outputs. When Master get/set input/output data of DCON modules via the i-7241D, the LED would flash.

condition	status	indicates
off	No data	No data is being transmitted or received
flashing red	Communicating	Data is being transmitted or received

2.5 7-segment LED

The i-7241D provides five 7-segment LED displays to show the information of i-7241D as follows.



①: Show the operation state of i-7241D. If it works normally, the LED displays the character 'n'.Otherwise, the LED displays the error character.

7-segment LED Number	Description
'n'	Normal operation
'1'	DCON modules init Error
'2'	i-7241D Hardware error
'd'	i-7241D Use default setting: Node ID=1 CAN baud 125K All IO connection path= default : assembly object instance

②: These two LED indicate the DeviceNet node ID of i-7241D by using hex format. For example, if the DeviceNet node ID of i-7241D is 31, these two LED will show the characters "1F".

③: This LED displays the CAN bus baud rate of i-7241D by number 0~2. The meanings of these numbers are described in the table below.

7-segment LED Number	Baud rate (K BPS)
0	125
1	250
2	500

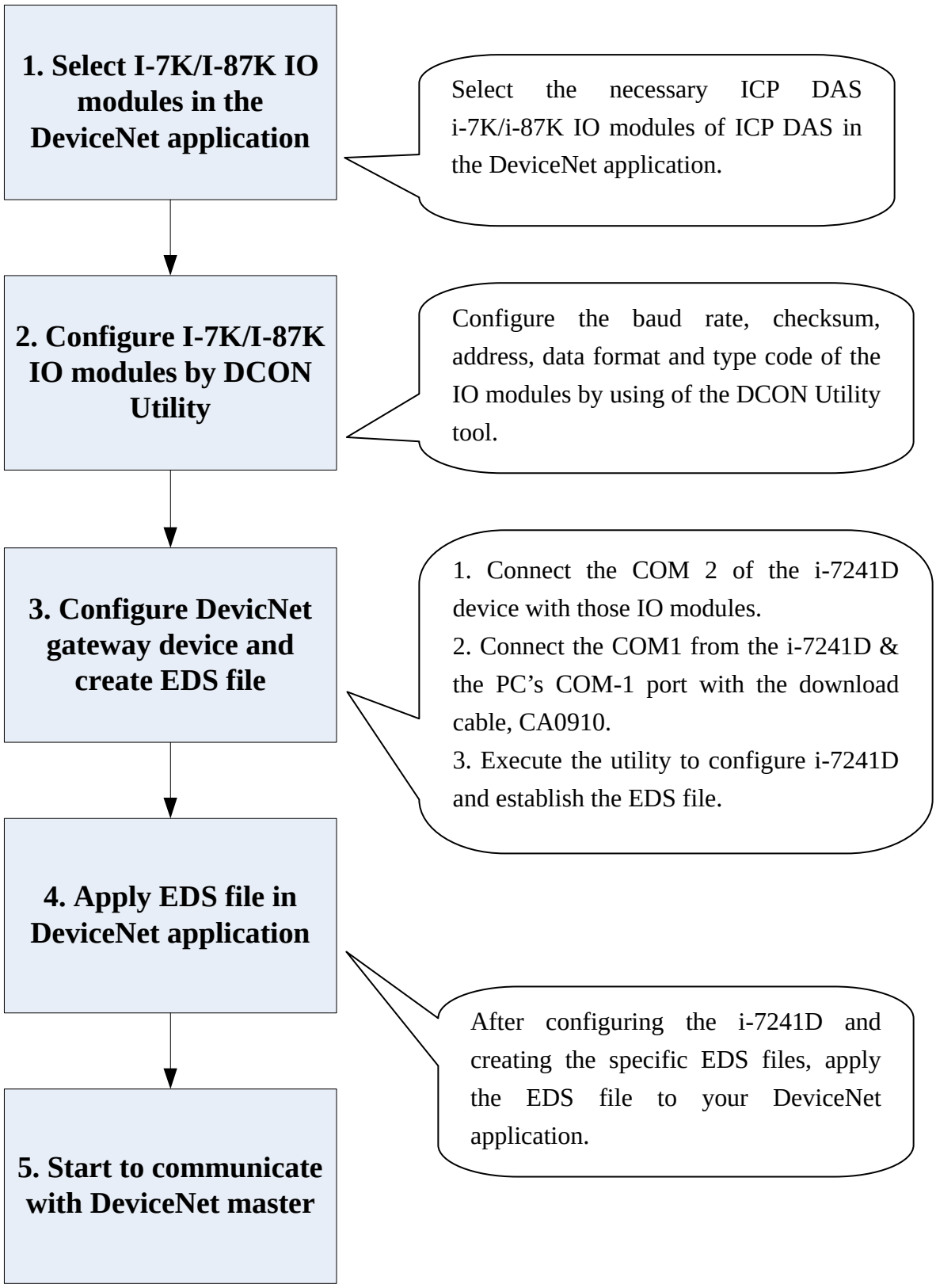
④: The RS-485 baud rate of i-7241D is indicated on this LED. The mapping table between LED number and RS-485 baud rate is displayed on the following table.

7-segment LED Number	Baud rate (BPS)
0	1200
1	2400
2	4800
3	9600
4	19200
5	38400
6	57600
7	115200

2.6 Module support

The i-7241D supports many kinds of DI, DO, AI and AO modules of i-7K/i-87K series. When users want to use these modules on the DeviceNet network, they must connect these modules with the COM2 of i-7241D. Then, the firmware built in the i-7241D will search them for organizing the corresponding DeviceNet entries automatically. Please refer to the "support module table.htm" to know which modules are supported by the i-7241D.
This file is located on the following location.
http://ftp.icpdas.com/pub/cd/fieldbus_cd/devicenet/gateway/i-7241d/manual/
or
fieldbus_cd:/devicenet/gateway/i-7241d/manual/

2.7 i-7241D Application Flowchart



Chapter 3 DeviceNet System

3.1 DeviceNet Introduction

The CAN (Controller Area Network) is a serial communication protocol, which efficiently supports distributed real-time control with a very high level of security. It is an especially suited for networking "intelligent" devices as well as sensors and actuators within a system or sub-system. In CAN networks, there is no addressing of subscribers or stations in the conventional sense, but instead, prioritized messages are transmitted.

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 3.1 shows the DeviceNet layer in the control and information layers.

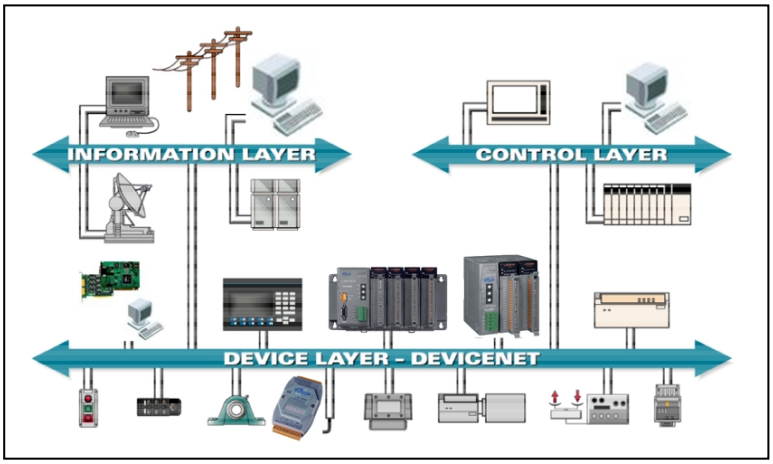


Figure 3.1 DeviceNet layer

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 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.

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

Table 3.1 DeviceNet's Use of the CAN Identifier Field

The i-7241D 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-to-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.

IDENTIFIER BITS											IDENTITY USAGE	HEX
10	9	8	7	6	5	4	3	2	1	0		RANGE
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	

Table 3.2 DeviceNet Identifiers

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 i-7241D (Group 2 Only). The next section would explain these objects. The detail information about Predefined Master/Slave Connection Set is described in the next section.

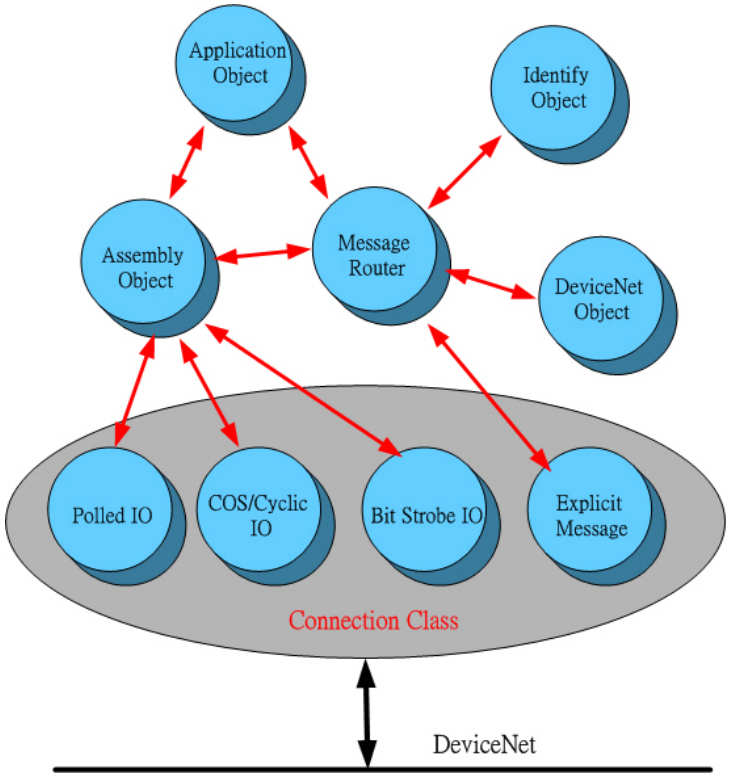


Figure 3.2 Object models of i-7241D

3.2 Predefined Master Slave Connection Set

The i-7241D provides “Predefined Master Slave Connection Set” device. Users must understand these connection set to know how to operate the device. The following section explains what the “Predefined Master Slave Connection Set” is.

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 establishing of a DeviceNet communication environment using less network and Device resources. The Predefined 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 types of messages are processed by a DeviceNet slave.

3.2.1 Explicit 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.

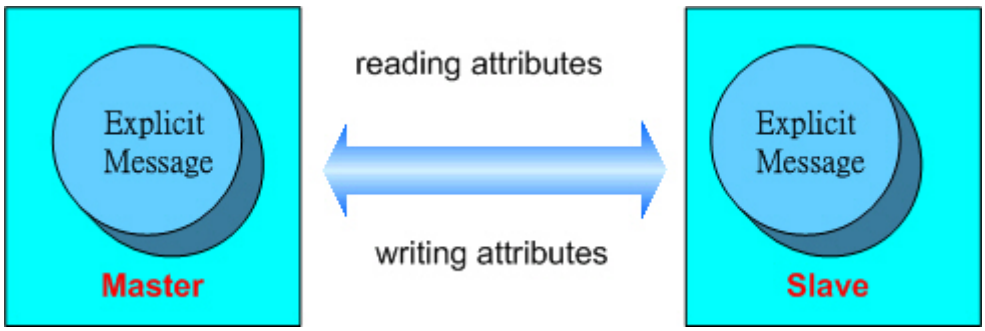


Figure 3.3 The architecture of Explicit message

3.2.2 I/O Bit Strobe Messages

The Bit-Strobe Command is an I/O message that is transmitted by the Master. A Bit-Strobe Command has multicast capabilities. Multiple Slaves can receive and react to the same Bit Strobe Command. The Bit-Strobe response is an I/O message that a Slave transmits back to the Master when the Bit-Strobe Command has been received. Within a Slave the two messages are received/ transmitted by a single Connection Object. The architecture is as figure 3.4.

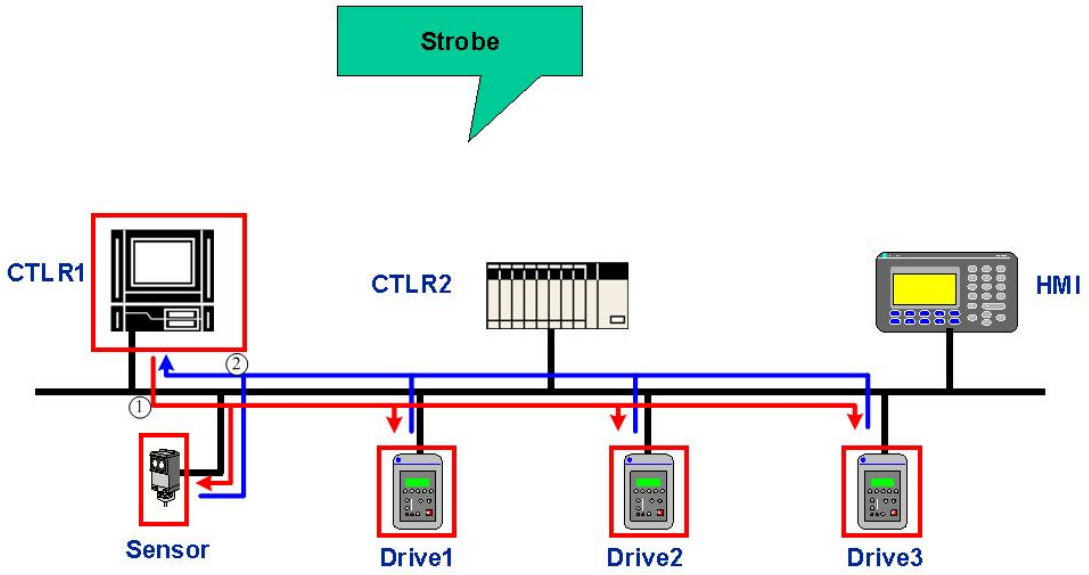


Figure 3.4 the architecture of IO bit strobe message

3.2.3 I/O Poll Messages

The Poll Command 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 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 3.5.

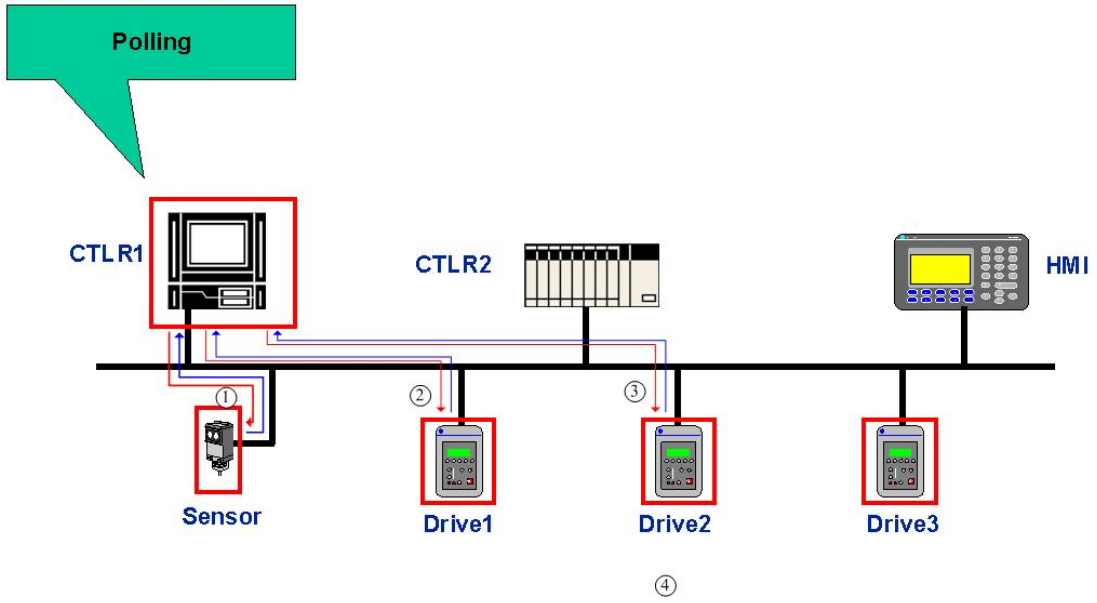


Figure 3.5 The architecture of IO bit strobe message

3.2.4 I/O Change of State/Cyclic Messages

The Change of State Message is transmitted by either the Master or the Slave. A Change of Sate/Cyclic is directed towards a single specific node (point-to-point). An Acknowledge Message may be returned in response to this message. Within either the Master or the Slave the producing Change of State Message and consuming Acknowledge Message are received or transmitted by one Connection Object. The consuming Change of State and producing Acknowledge Message are received or transmitted by a second Connection Object. The architecture is as figure 3.6 and figure 3.7.

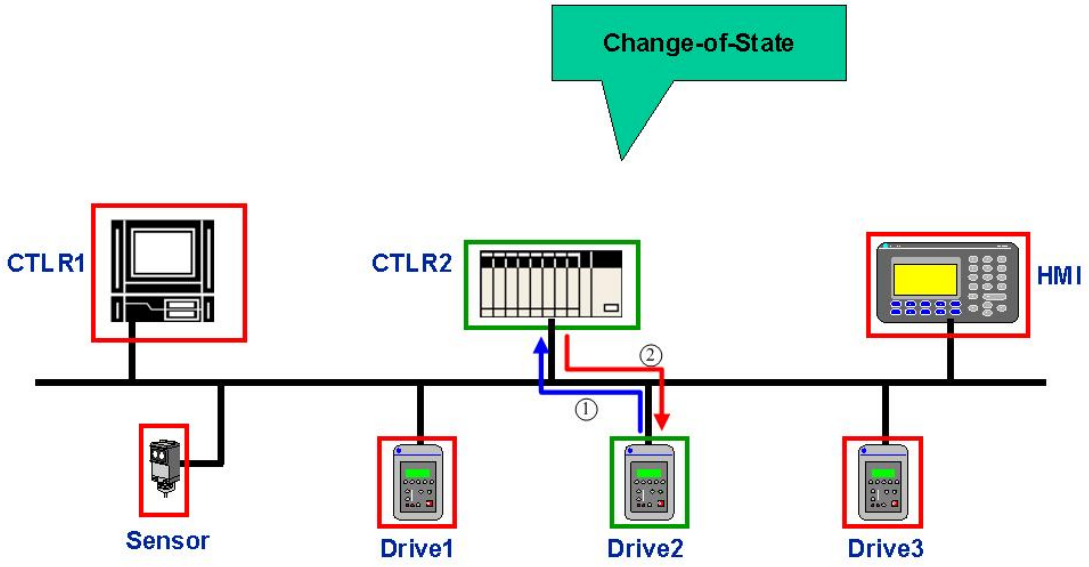


Figure 3.6 Architecture of IO COS message

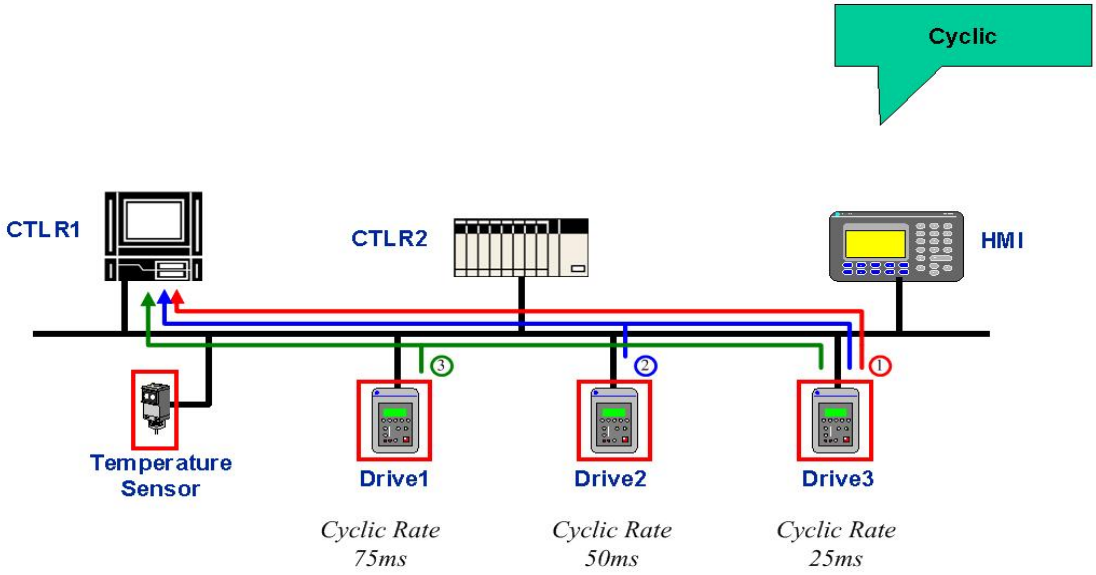


Figure 3.7 Architecture of IO Cyclic message

3.3 EDS file

An Electronic Data Sheet is an external disk that contains information about configurable attributes for a device, including the object addresses of each parameter. The following figure shows the configuration of a device through configuration tool that supports an EDS. The application objects in the device represent the destination addresses for the configuration data. These addresses are encoded in the EDS. ICP DAS provides users with CAN gateway utility software for users to create the suitable EDS file. The EDS file system architecture is as figure 3.8.

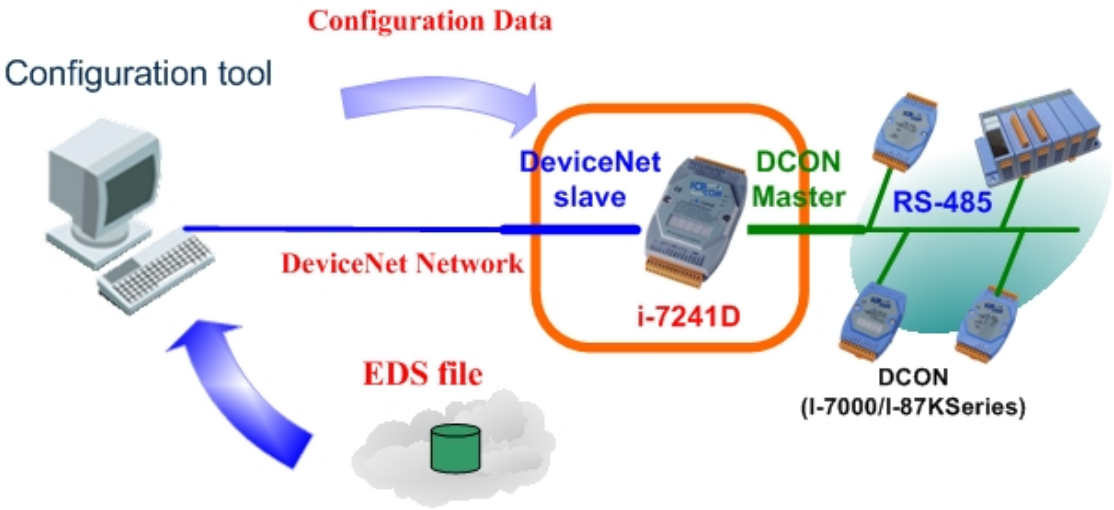


Figure 3.8 Architecture of the EDS file

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. The CAN utility is a very powerful tool for the DeviceNet network. Users can get more DeviceNet Slave information. The CAN utility can scan i-7K/i-87K IO modules connected with the i-7241D. It also provides the graph interface for users to make up the EDS file of their system. For more detail information on this topic, please refer to next section.

Chapter 4 DeviceNet Profile Area

This chapter is for users who want to understand more detailed information related to the i-7241D device when using the DeviceNet protocol. This section documents the detailed functions for each object class that is implemented in the DeviceNet gateway system.

4.1 DeviceNet Statement of Compliance

General Device Data

Conforms to DeviceNet Specification	Volume I-Release 1.1
Vendor Name	ICP DAS
Device Profile Name	ICPDAS-I7241D
Production Revision	2.1

DeviceNet Physical Conformance Data

Network Power Consumption(Max)	Open-Hardwired
Isolated Physical Layer	Yes
LEDs Supported	Yes
MAC ID Setting	Software
Default MAC ID	0x01
Communication Rate Setting	Software (Default is 125k bits/s)
Predefined Master/Slave Connection Set	Group 2 Only Server
Fragmented Explicit Message implemented	yes

4.2 Identity Object (Class ID: 0x01)

This object provides the identification of and general information about the device.

Class Attribute

Attribute ID	Attribute name	Data Type	Method	Value
0x01	Revision	UINT	Get	0001
0x02	Max Instance	UINT	Get	1

Class Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes

Instance attribute

Attribute ID	Description	Data Type	Method	Default Value
1	Vendor UINT	UINT	Get	803
2	Device type	UINT	Get	0x00
3	Product code	UINT	Get	1
4	Vendor Revision Major Revision Minor Revision	Struct. of USINT USINT	Get	2.1
5	Status	WORD	Get	0
6	Serial number	UDINT	Get	1
7	Product name	Short_String	Get	"ICPDAS-I7241D"
10	Heartbeat Interval	USINT	Get/Set	0(default)

Instance Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes
0x10	Set_Attribute_Single	Yes
0x05	Reset	Yes

4.3 DeviceNet object (Class ID:0x03)

The DeviceNet Object is used to provide the configuration and status of a physical attachment to the DeviceNet.

Class Attribute

Attribute ID	Attribute name	Data Type	Method	Default Value
0x01	revision	UINT	Get	2

Class Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes

Instance attribute

Attribute ID	Description	Data Type	Method	Default Value
1	MAC ID	USINT	Get/Set	0~63
2	Baud rate	USINT	Get/Set	0~2
3	BOI	USINT	Get/Set	0
4	Bus-off counter	USINT	Get/Set	0
5	Allocation information	BYTE	Get/Set	0

Instance Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes
0x10	Set_Attribute_Single	Yes

4.4 Assembly object (Class ID: 0x04)

The Assembly Object binds the attributes of multiple objects together, which allows data transfer to or from each object to be sent or received over a single connection. Assembly objects can be used to bind input data or output data. The terms, ”input” and ”output”, are defined from the network’s point of view. An input will produce data on the network and an output will consume data from the network.

Class Attribute

Attribute ID	Attribute name	Data Type	Method	Default Value
0x01	revision	UINT	Get	2
0x02	Max Instance	UINT	Get	15

Class Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes

Instance attribute

Attribute ID	Attribute name	Data Type	Method	Value
0x03	Data	Defined by users	Get/Set	0

Instance Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes
0x10	Set_Attribute_Single	Yes

4.5 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.

Connection Instance ID	Description
1	References the Explicit Messaging Connection into the Server
2	References the Poll I/O Connection
3	References the Bit–Strobe I/O Connection
4	References the Slave's Change of State or Cyclic I/O Connection

Class Attribute

Attribute ID	Attribute name	Data Type	Method	Value
0x01	Revision	UINT	Get	1

Class Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes

Instance (0x01) attribute – Explicit message connection

Attribute ID	Description	DeviceNet Data Type	Method	Default Value
0x01	state	USINT	Get	3
0x02	instance_type	USINT	Get	0
0x03	transportClass_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	0x20
0x08	consumed_connection_size	UINT	Get	0x20
0x09	expected_packet_rate	UINT	Get	0x09c4
0x0C	watchdog_timeout_action	USINT	Get	1
0x0D	produced_connection_path_ length	UINT	Get	0
0x0E	produced_connection_path	EPATH	Get	Empty
0x0F	consumed_connection_path_length	UINT	Get	0
0x10	consumed_connection_path	EPATH	Get	Empty
0x11	production_inhibit_time	UINT	Get	0

Instance Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes
0x10	Set_Attribute_Single	Yes

Instance (0x02) attribute – Polling IO connection

Attribute ID	Description	DeviceNet Data Type	Method	Default Value
0x01	state	USINT	Get	0x01
0x02	instance_type	USINT	Get	0x01
0x03	transportClass_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	No specified default
0x08	consumed_connection_size	UINT	Get	No specified default
0x09	expected_packet_rate	UINT	Get	0
0x0C	watchdog_timeout_action	USINT	Get	0
0x0D	produced_connection_path_ length	UINT	Get	No specified default
0x0E	produced_connection_path	EPATH	Get	No specified default
0x0F	consumed_connection_path_length	UINT	Get	No specified default
0x10	consumed_connection_path	EPATH	Get	No specified default
0x11	production_inhibit_time	UINT	Get	0

Instance Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes
0x10	Set_Attribute_Single	Yes

Instance (0x03) attribute – Bit-Strobe IO connection

Attribute ID	Description	DeviceNet Data Type	Method	Default Value
0x01	state	USINT	Get	0x01
0x02	instance_type	USINT	Get	0x01
0x03	transportClass_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	0x02
0x07	produced_connection_size	UINT	Get	No specified default
0x08	consumed_connection_size	UINT	Get	0x08
0x09	expected_packet_rate	UINT	Get	0
0x0C	watchdog_timeout_action	USINT	Get	0
0x0D	produced_connection_path_ length	UINT	Get	No specified default
0x0E	produced_connection_path	EPATH	Get	No specified default
0x0F	consumed_connection_path_length	UINT	Get	No specified default
0x10	consumed_connection_path	EPATH	Get	No specified default
0x11	production_inhibit_time	UINT	Get	0

Instance Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes
0x10	Set_Attribute_Single	Yes

Instance (0x04) attribute (Acknowledge) – COS/Cyclic IO connection

Attribute ID	Description	DeviceNet Data Type	Method	Default Value
0x01	state	USINT	Get	0x01
0x02	instance_type	USINT	Get	0x01
0x03	transportClass_trigger	BYTE	Get	0x02
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	No specified default
0x08	consumed_connection_size	UINT	Get	No specified default
0x09	expected_packet_rate	UINT	Get	0
0x0C	watchdog_timeout_action	USINT	Get	0
0x0D	produced_connection_path_ length	UINT	Get	No specified default
0x0E	produced_connection_path	EPATH	Get	No specified default
0x0F	consumed_connection_path_length	UINT	Get	No specified default
0x10	consumed_connection_path	EPATH	Get	20h 2Bh 24h 01h
0x11	production_inhibit_time	UINT	Get	0

Instance Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes
0x10	Set_Attribute_Single	Yes

Instance (0x04) attribute (Unacknowledge) – COS/Cyclic IO connection

Attribute ID	Description	DeviceNet Data Type	Method	Default Value
0x01	state	USINT	Get	0x01
0x02	instance_type	USINT	Get	0x01
0x03	transportClass_trigger	BYTE	Get	0x00
0x04	produced_connection_id	UINT	Get	Table 3.2
0x05	consumed_connection_id	UINT	Get	0xFFFF
0x06	initial_comm_characteristics	BYTE	Get	0x0F
0x07	produced_connection_size	UINT	Get	No specified default
0x08	consumed_connection_size	UINT	Get	No specified default
0x09	expected_packet_rate	UINT	Get	0
0x0C	watchdog_timeout_action	USINT	Get	0
0x0D	produced_connection_path_ length	UINT	Get	No specified default
0x0E	produced_connection_path	EPATH	Get	No specified default
0x0F	consumed_connection_path_length	UINT	Get	0
0x10	consumed_connection_path	EPATH	Get	Empty
0x11	production_inhibit_time	UINT	Get	0

Instance Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes
0x10	Set_Attribute_Single	Yes

4.6 Application object (Class ID:0x64)

The application objects are the interfaces between an application and the DeviceNet Layer. Application Objects attributes contain the application data, which is to be accessed and exchanged via the DeviceNet. The DeviceNet accesses the application data by invoking read and write functions. These functions need to be provided by an Application Object. The i-7241D provides Get_Attribute_Single and Set_Attribute_Single to read and write data to i-7K/i-87K IO modules.

Class Attribute

Attribute ID	Attribute name	Data Type	Method	Default Value
0x01	Revision	UINT	Get	2
0x02	Max Instance	UINT	Get	Defined by DCON modules
0x03	DCON Com port Timeout Value	UINT	Get/Set	defined
0x04	Enable/Disable watchdog message	BYTE	Get/Set	0 - Disable 1 - Enable

Class Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes
0x10	Set_Attribute_Single	Yes

Instance attribute

Attribute ID	Description	Data Type	Method	Default Value
0x01	Module ID	DWORD	Get	0
0x02	Module address	CHAR	Get	0
0x03	Checksum	CHAR	Get	0
0x04	Baud rate	DWORD	Get	0
0x05	Type Code	CHAR	Get	0
0x06	Data Format	CHAR	Get	0
0x07	DCON module data	DWORD	Get	0

	lose counter			
0x08	Enable/Disable watch dog	CHAR	Get/Set	0
0x09	Watch dog time period	CHAR	Get/Set	50
0x0A	Watch dog status	CHAR	Get	0
0x0B	Clear Watch Dog	NULL	Set	NULL
0x0C	DO Length	CHAR	Get	0
0x0D	AO Length	CHAR	Get	0
0x0E	DI Length	CHAR	Get	0
0x0F	AI Length	CHAR	Get	0
0x10	DO channel num	CHAR	Get	0
0x11	AO channel num	CHAR	Get	0
0x12	DI channel num	CHAR	Get	0
0x13	AI channel num	CHAR	Get	0
0x14	DO data	Defined by module channel num	Set/Get	0
0x15	AO data	Defined by module channel num	Set/Get	0
0x16	DI data	Defined by module channel num	Get	0
0x17	AI data	Defined by module channel num	Get	0
0x18	Reset Counter value	Defined by module channel num	Set	0

Instance Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes
0x10	Set_Attribute_Single	Yes

Chapter 5 CAN Gateway Utility

5.1 CAN Gateway Utility Overview

The i-7241D can scan the i-7K modules and assigned these modules to their application and assembly objects automatically. Users can apply the i-7241D into the DeviceNet application. They must understand the relationship between these DeviceNet application and assembly objects in the i-7241D. ICP DAS provides the CAN Gateway Utility to configure the communication parameters, IO connection path and the EDS file for the i-7241D device. The software also provides the information of assembly and application objects. The Utility can also scan the i-7K modules on the 485 bus in the i-7241D.

Before users start to use the i-7241D, they must set the i-7K/i-87K IO modules to have a different ID, 0x01~0x0F, and same baud-rate. If the modules have the same ID or different baud-rate, it would cause errors in the system. ICP DAS provides DCON Utility to configure the DCON modules. The DCON Utility can free download in the following web site.

http://www.icpdas.com/download/7000/7000.htm

For more information about how to configure the i-7K/i-87K modules, please refer to the user manual of the i-7K/87K modules.

After configuring the i-7K/i-87K IO modules by using the DCON utility tool, users can use the CAN Gateway Utility to configure the DeviceNet connection path and create a relative EDS file.

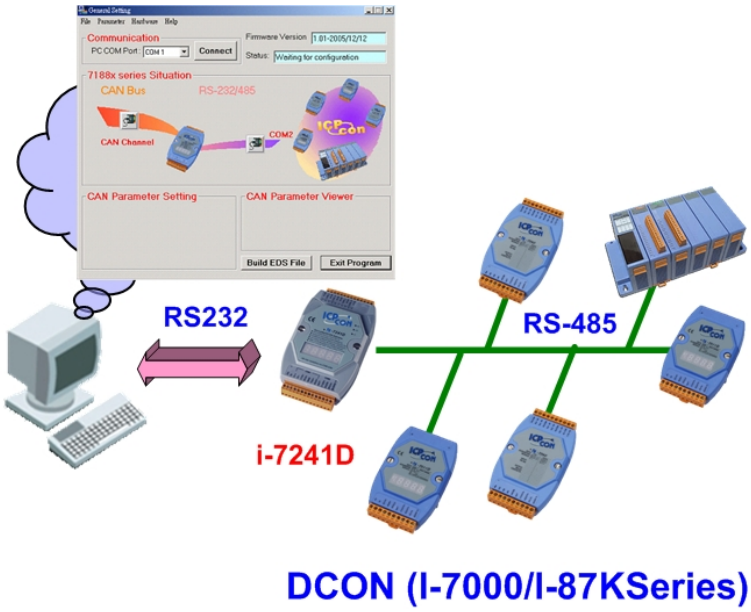


Figure 5-1 CAN_GW Utility architecture

5.2 Configuration with the CAN gateway Utility

Step 1: Download the CAN Gateway Utility setup files from http://www.icpdas.com/download/can/Gateway_Converter.htm or CD-ROM disk drive in the path of “fieldbus_cd/devicenet/gateway/i-7241D/utility/.

Step 2: Execute the setup.exe file to install CAN Gateway Utility.

Step 3: A “Welcome” window pops up to prompt user as Fig 5-2 shown.

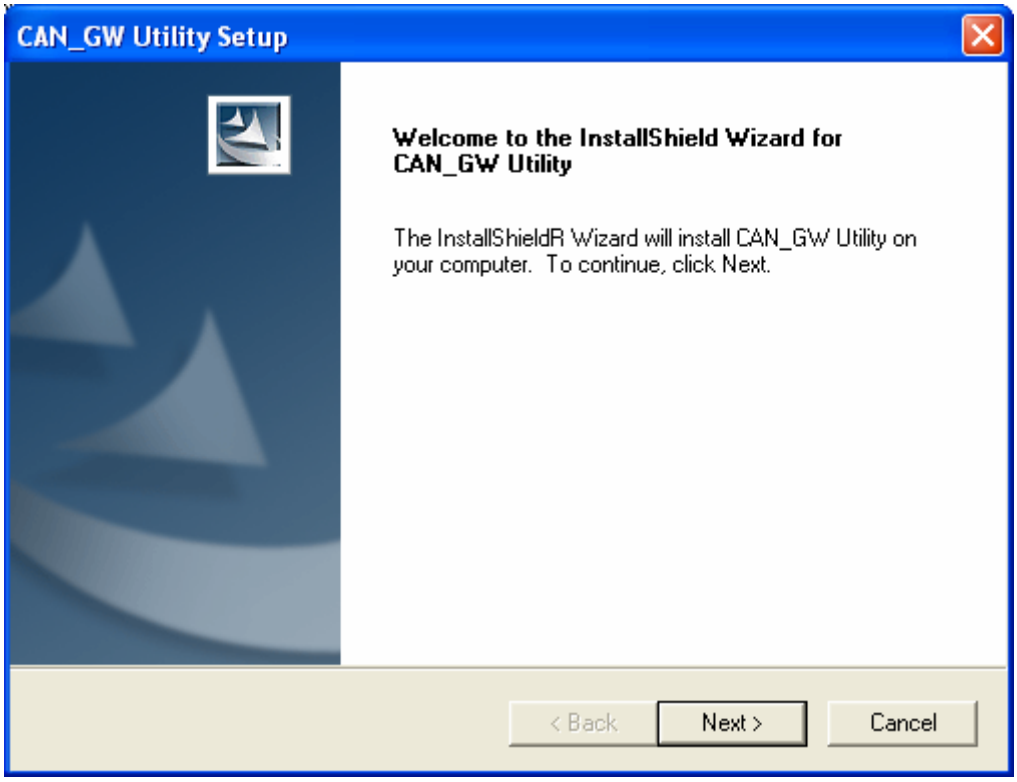


Fig 5-2. Welcome dialog

Step 4: Click “Next” button and A “Choose Destination Location” window pops up to prompt user as Fig 5-3 shown.

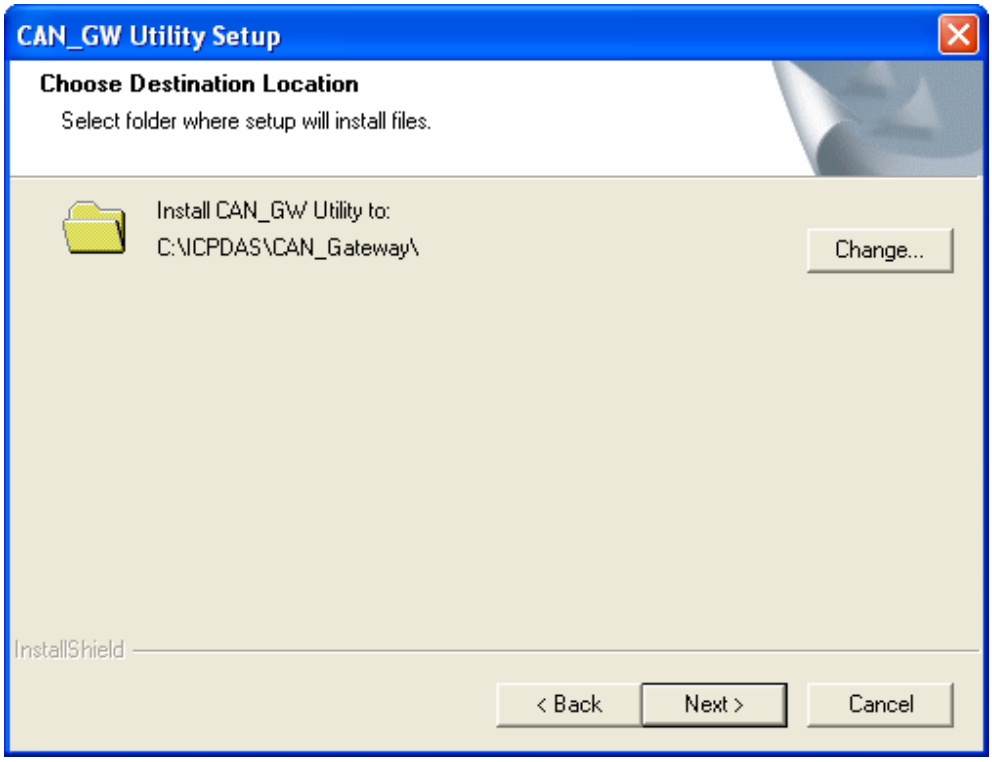


Fig 5-3. “Choose Destination Location” dialog

Step 5: Click “Next” button and A “Ready to Install” window pops up to prompt user as Fig 5-4 shown.

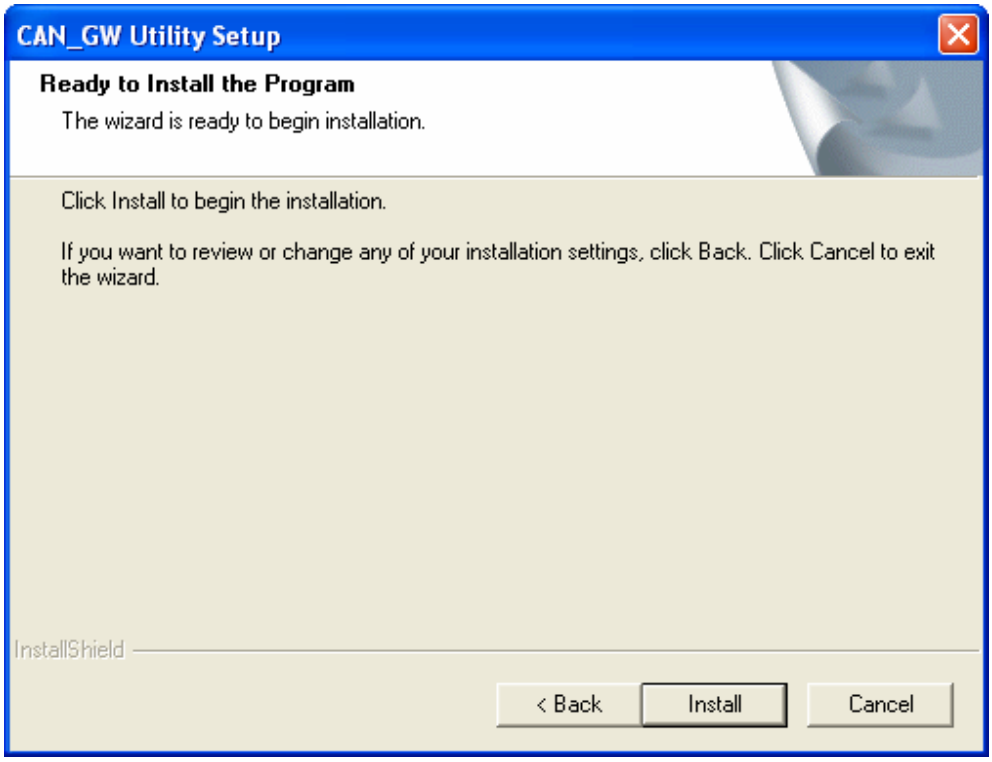


Fig 5-4. “Ready to Install the Program” dialog

Step 6: Click the “Next” button and start to install the CAN Gateway Utility to the system. After finishing the process, A “Complete” window pops up to prompt user as Fig 5-5 shown.

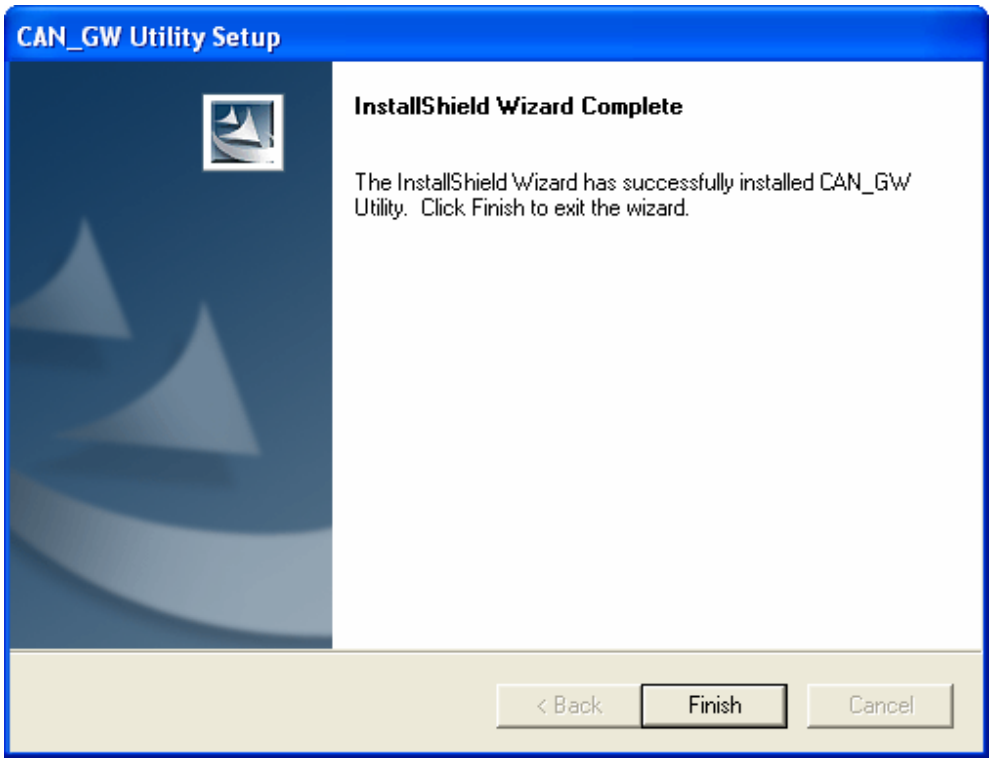


Fig 5-5. “Complete” dialog

Step 7: After finishing installing the CAN Gateway Utility, users can find CAN_GW Utility as following figure 5-6.

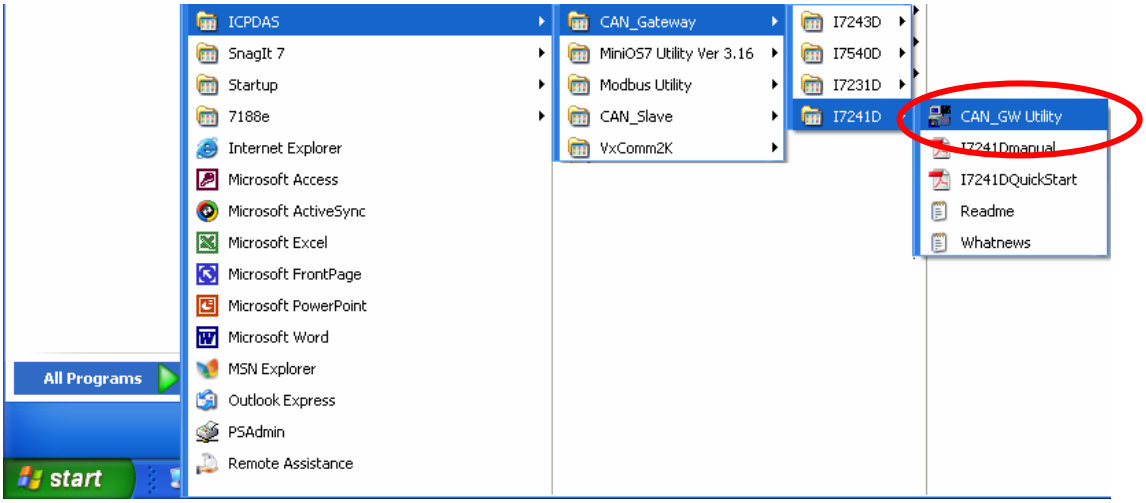


Fig 5-6. You can find “CAN_GW Utility“ at Start in the task bar

5.3 Uninstall CAN Gateway Utility

You can uninstall CAN gateway Utility software by following describing steps:

Step 1: Clicking Start in the task bar, then clicking Setting/Control Panel as shown in Fig 5-10.

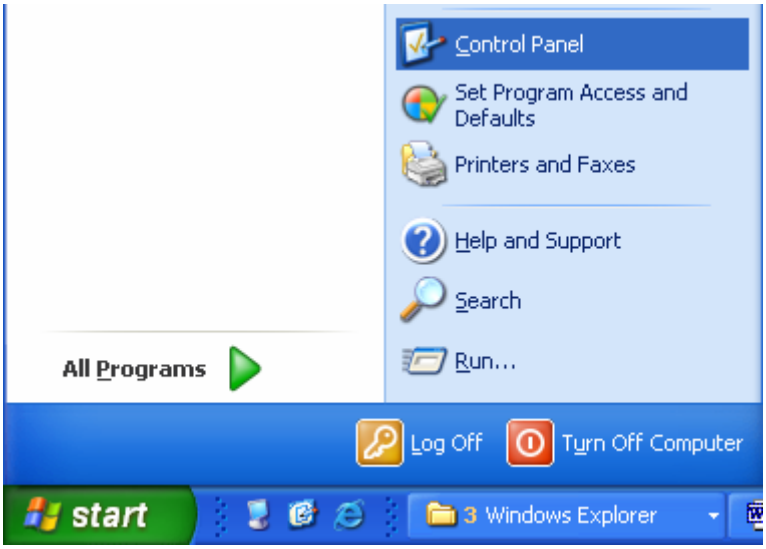


Fig 5-10. Select settings

Step 2: Click Add/Remove Programs icon as shown in Fig 5-11.

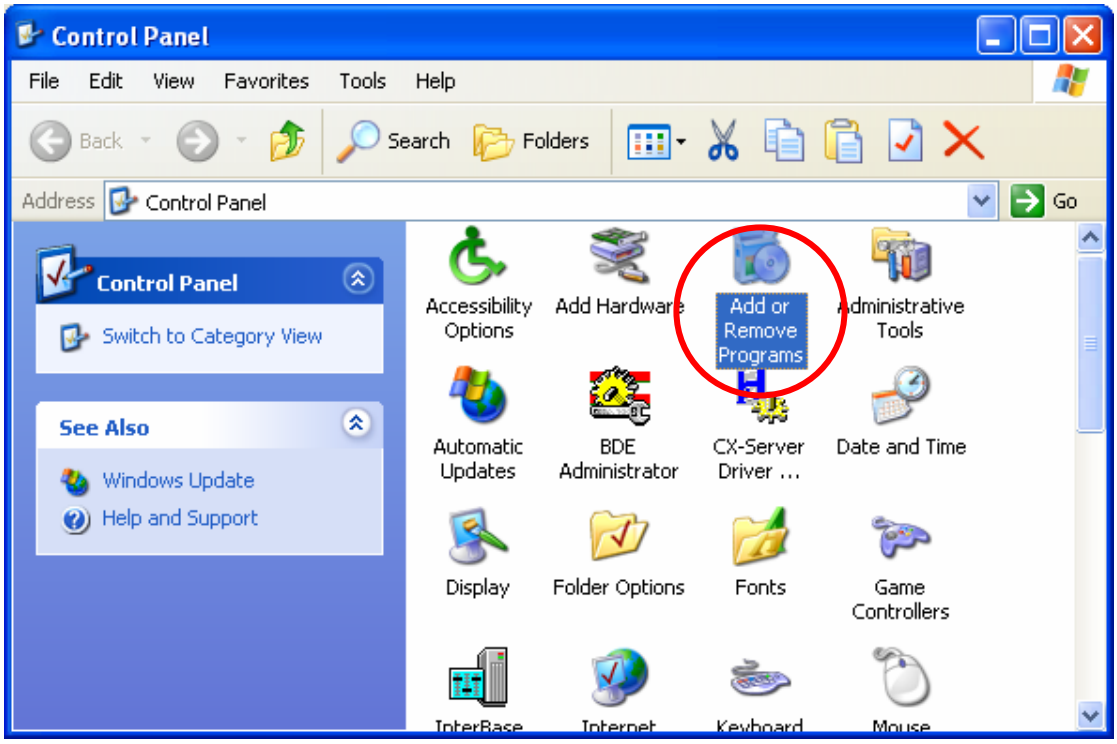
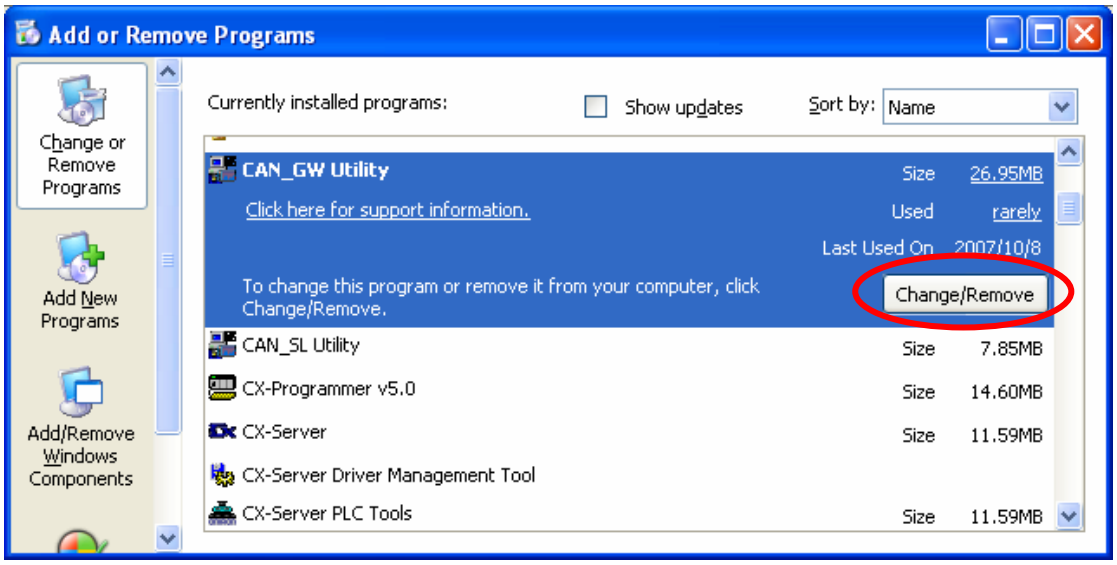
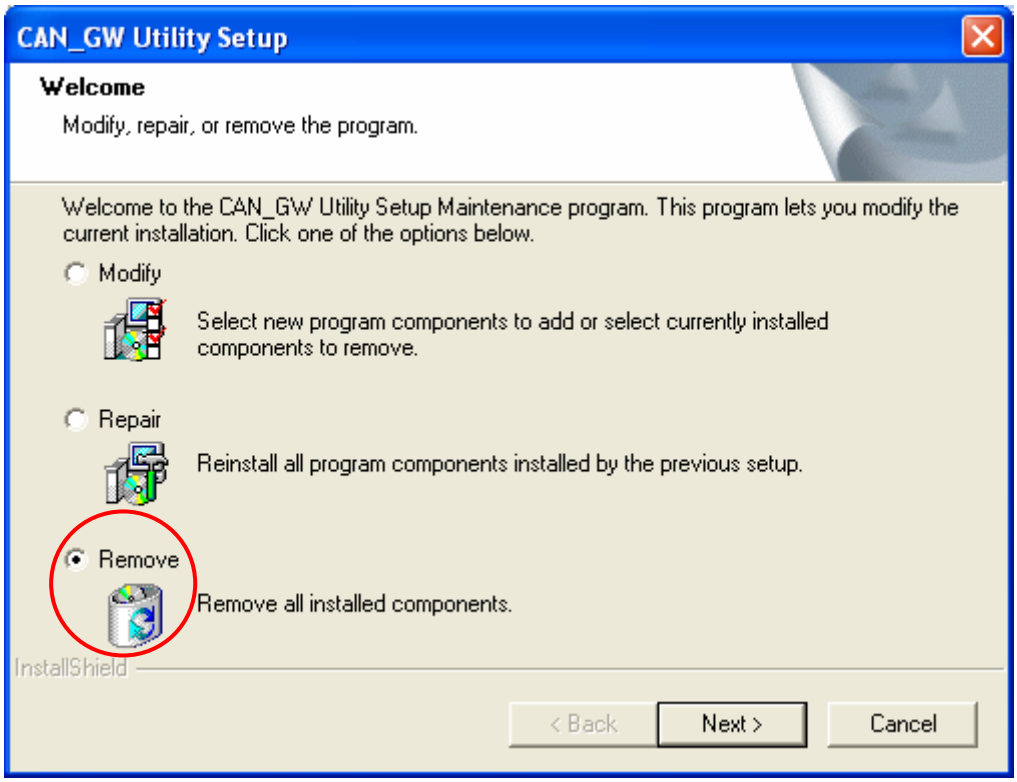


Fig 5-11. Click “Add/Remove Programs”

Step 3: Select the “CAN GW Utility” then click button “Change/Remove” to start the uninstall process. Then select the “Remove” option and click the “Next” button to remove the software.





Step 4: Click the “Yes” button to remove the software as shown in Fig 5-12.

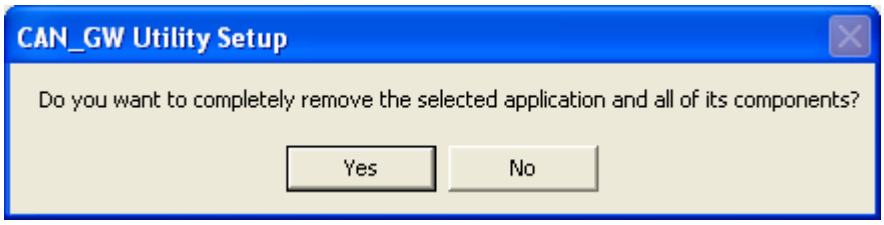


Fig 5-12. Click the button “Yes” to remove the software

Step 5: Click the “Finish” button to finish the uninstalling process. Refer to Fig 5-13.

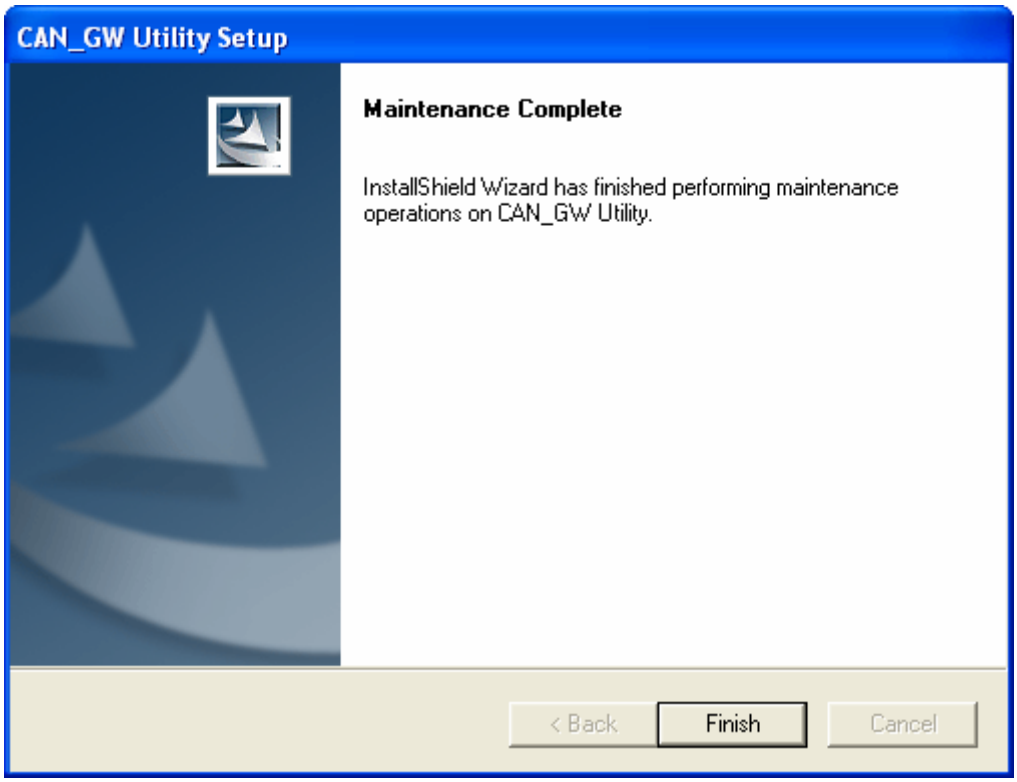


Fig 5-13. Click button “Finish”.

5.4 CAN Gateway Utility Steps

Before using this utility software, please make sure that you have connected the i-7241D and your PC. And set the i-7K/87K modules with the DCON Utility tool. The architecture is depicted in the following figure 5-14.

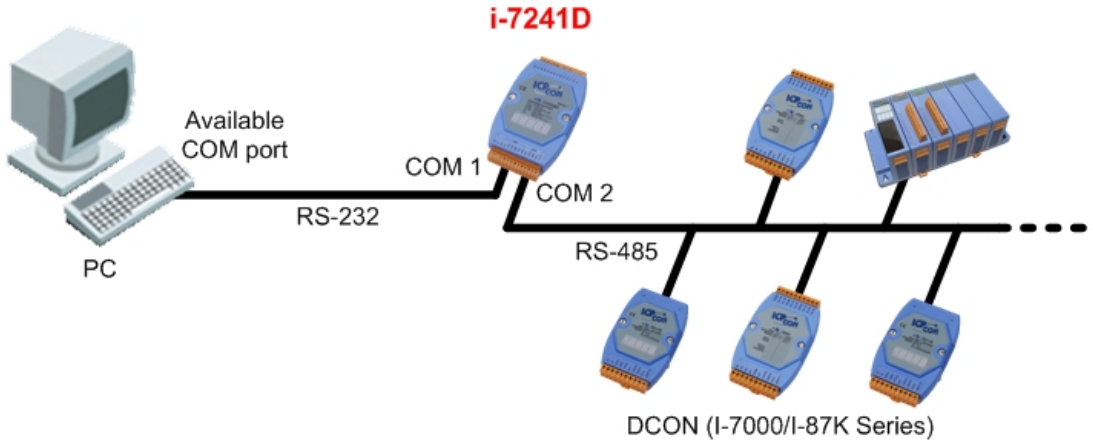


Fig 5-14. The configuration architecture of the i-7241D

Step 1. Turn off the i-7241D. Connect the INIT* pin with the GND pin of the i-7241D as figure 5-15.

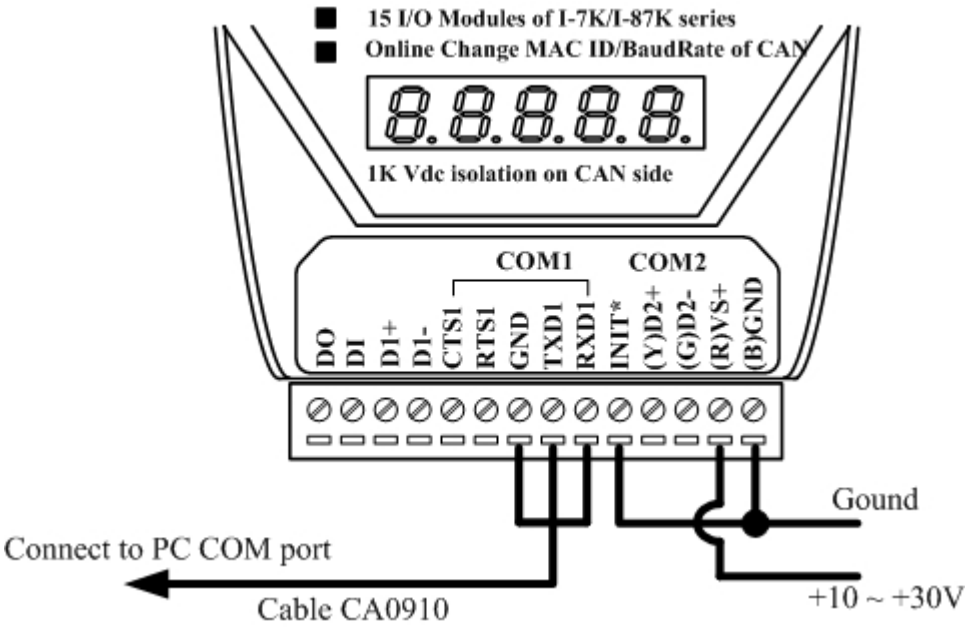


Fig 5-15. The wire connection of the i-7241D

Step 2: Turn on the i-7241D. Then execute the CAN_GW.exe file. The startup figure would be displayed as figure 5-16.

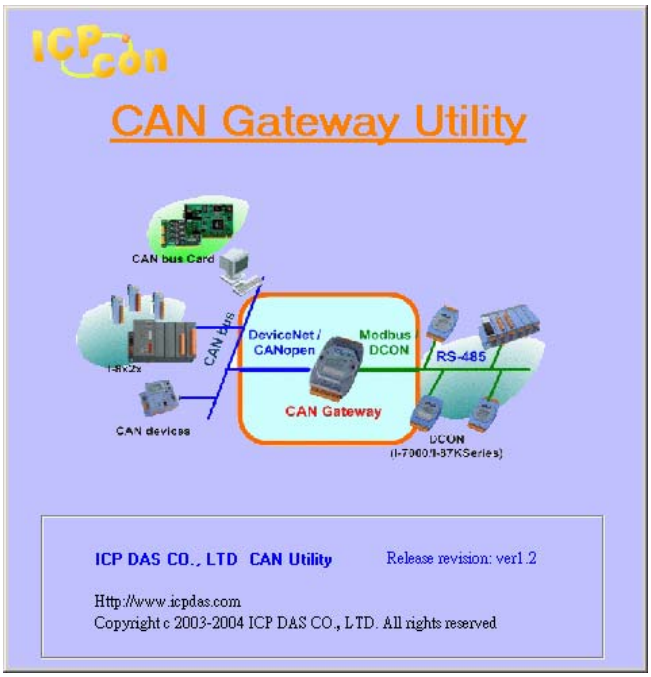


Fig 5-16. Start-up figure

After shows the startup figure, the software figure would be displayed as figure 5-17.

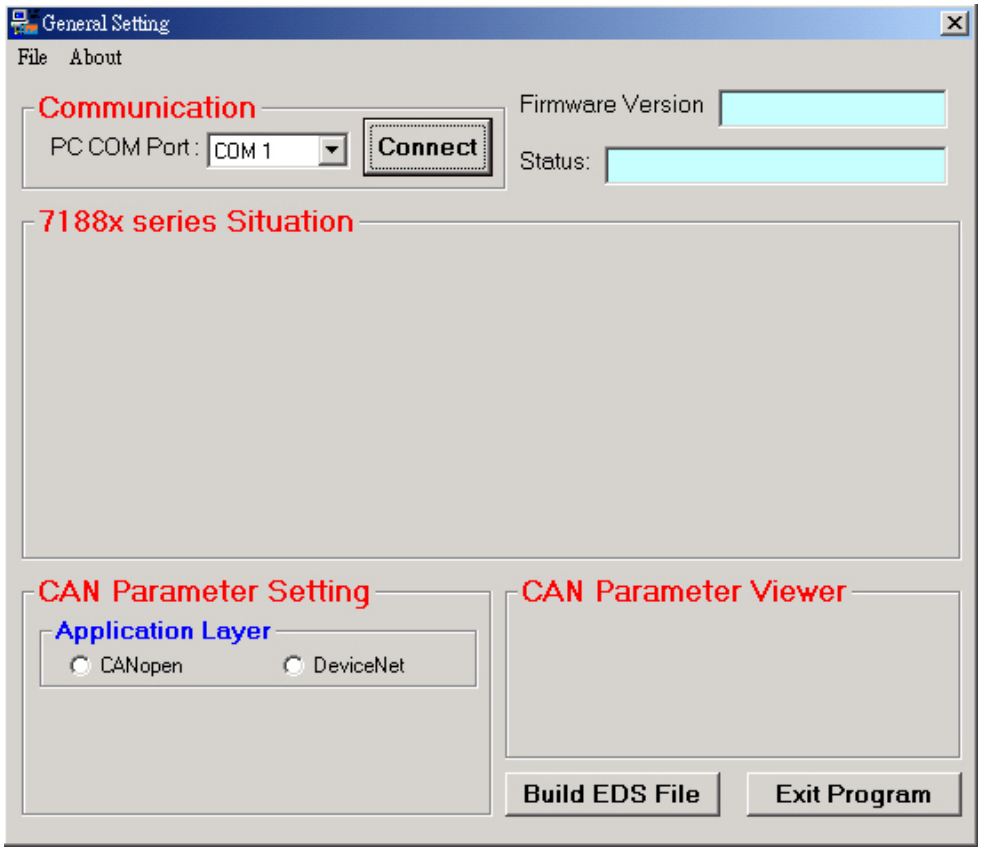


Fig 5-17. “General-Setting” window

Step 3: Press the “Connect” button to connect the i-7241D. Then the “Com Port Scan Parameter Setting” dialog will be pop-up as follows. Please set the correct com port communication parameters to scan the i-7K modules. Then press the “OK” button to begin the modules scans as figure 5-18.

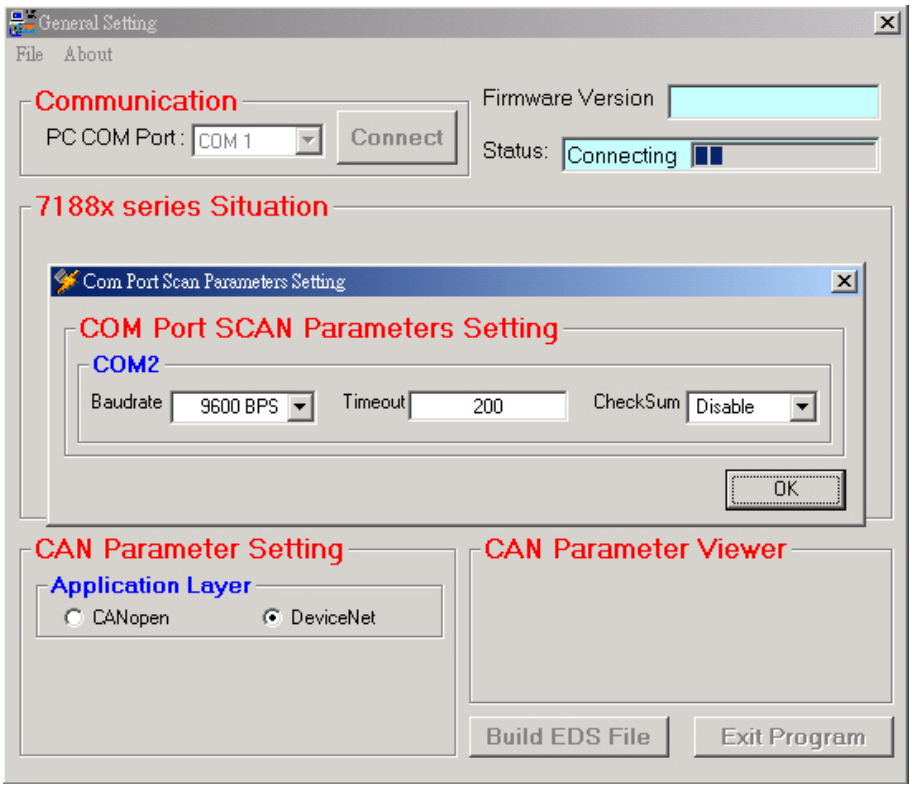


Fig 5-18 “Com Port Scan Parameters Setting”

Step 4: When the i-7K modules scan is finished; the result of scan will be compared with the parameters stored in the EEPROM of the i-7241D. If any difference has been detected, the warning message will pop up as figure 5-19.

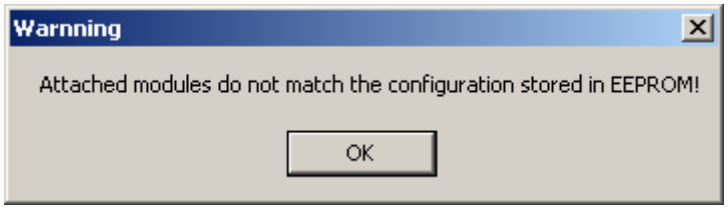


Fig 5-19 Warning message dialog

Because of the default connected module is I-7012, I-7021, I-7053 and I-7043. If users connect the i-7241D with any different I/O module described above, the “Attached modules do not match the configuration store in EEPROM!” warning message may pop up. In this case, the default value will be shown on each parameter setting field. If no error or warning message occurs, the last setting value will be displayed on each parameter setting field.

Step 5: When the i-7K modules scan is finished; a related information dialog box will be displayed as figure 5-20.

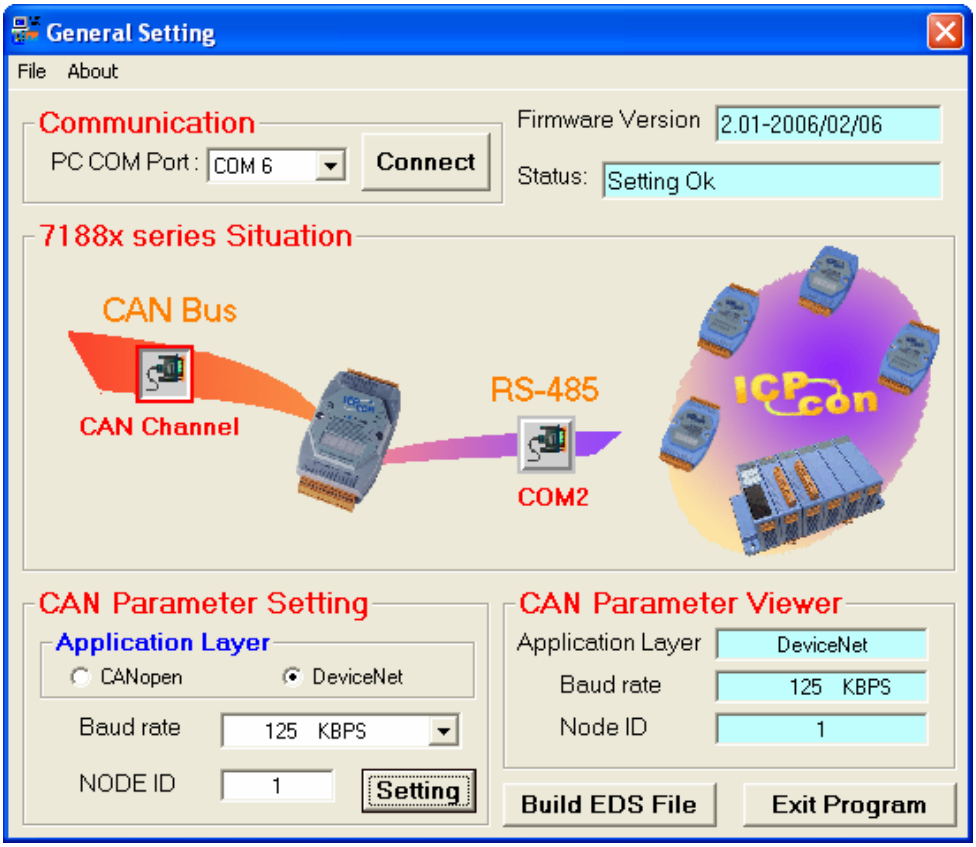


Fig 5-20

Step 6: Click the “CAN Channel” button,  so that the CAN bus configuration information will be given. Then, users can set the necessary CAN bus communication information. Afterwards, click the “Setting” button to finish the CAN parameter setting. The CAN Parameter Viewer frame on the right hand side indicates the parameter setting result. After clicking the “Setting” button, users can see that the each field value of the CAN Parameter Viewer frame is changed to the value configured in the CAN Parameter Setting frame on the left hand side as Fig 5-21.

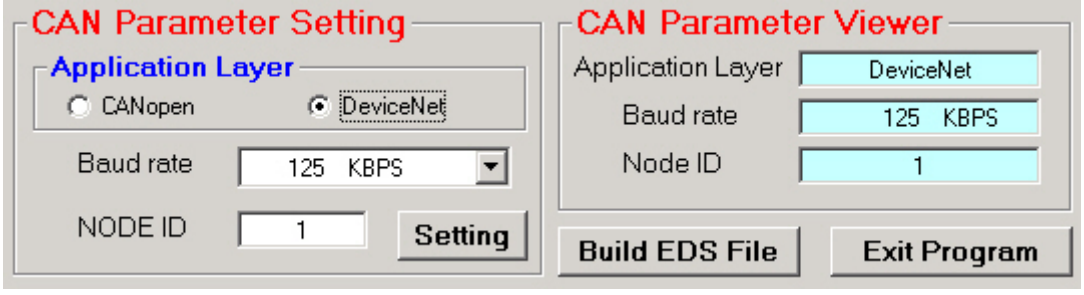


Fig 5-21

Step 7: Press the “COM 2”  button to display the com 2 bus configuration information on the i-7241D. Press the “Setting” button to set the needed CAN bus communication information in the dialog box as figure 5-22.

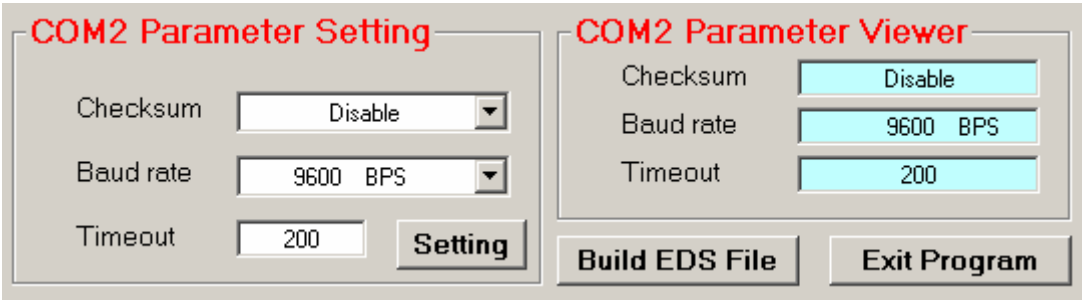


Fig 5-22

Step 8: Start to build the specific EDS file for the i-7241D. If the I/O connection path stored in EEPROM of i-7241D is not correct. The warring dialog would pop-up as figure 5-23.

Fig 5-23

Step 9: Start to build the specific EDS file for your DeviceNet gateway. The DeviceNet EDS file information is set according to the following dialog box. Users can configure the relative information for their EDS file using a dialog box like Fig 5-24.

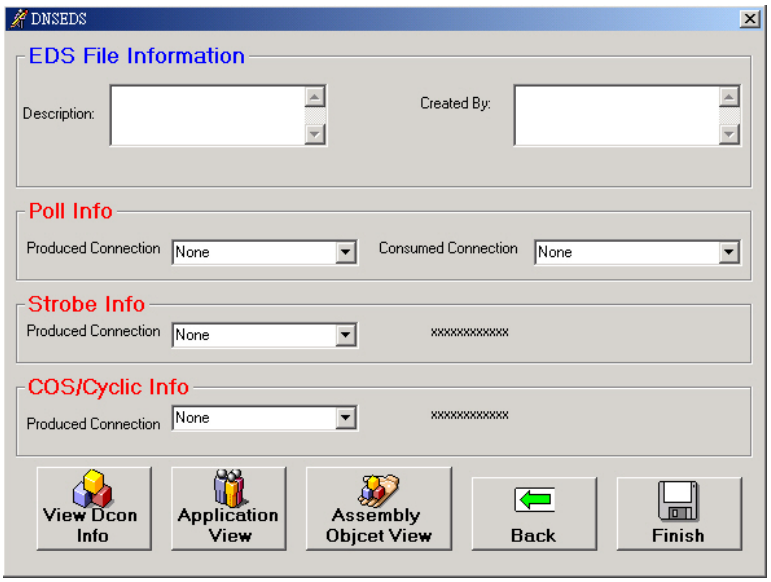


Fig 5-24

Step 10: Setting the EDS file information and give it a description in the description box provided as figure 5-25.

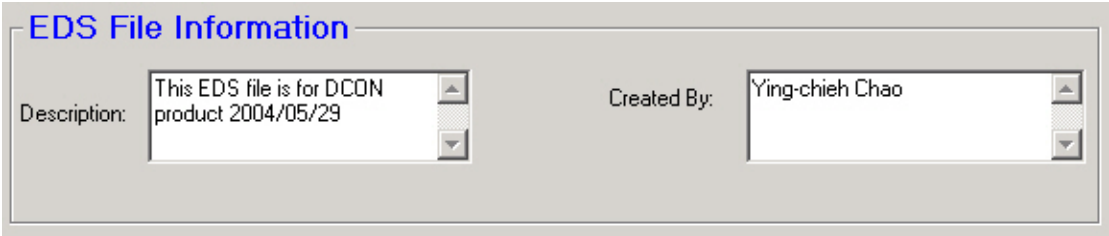


Fig 5-25

Step 11: Set the Polling/Bit Strobe/COS/Cyclic IO connection path for the i-7241D as figure 5-26.

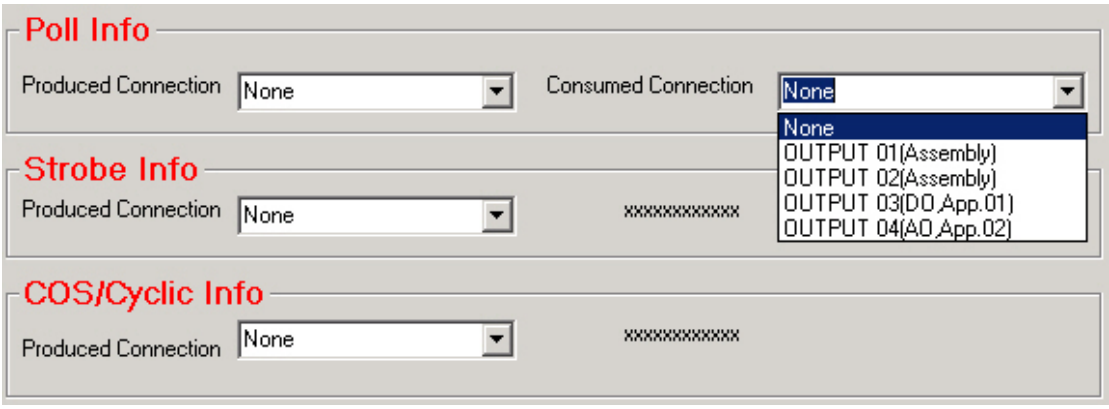


Fig 5-26

Step 12: The i-7241D also provides i-7K/i-87K IO modules, application objects and assembly objects information for users by clicking either the 

button, the 

button or the 

button. Then, these windows would pop-up as figure 5-27, figure 5-28 and figure 5-29

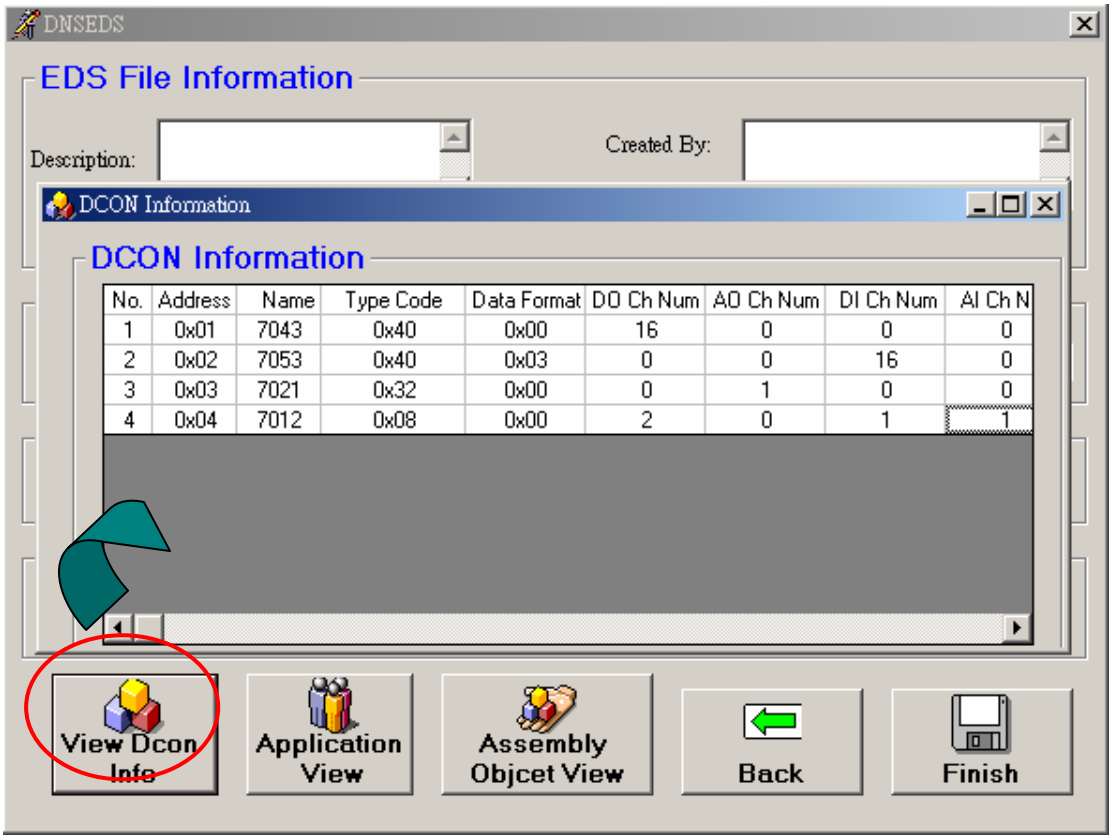


Fig 5-27 DCON information

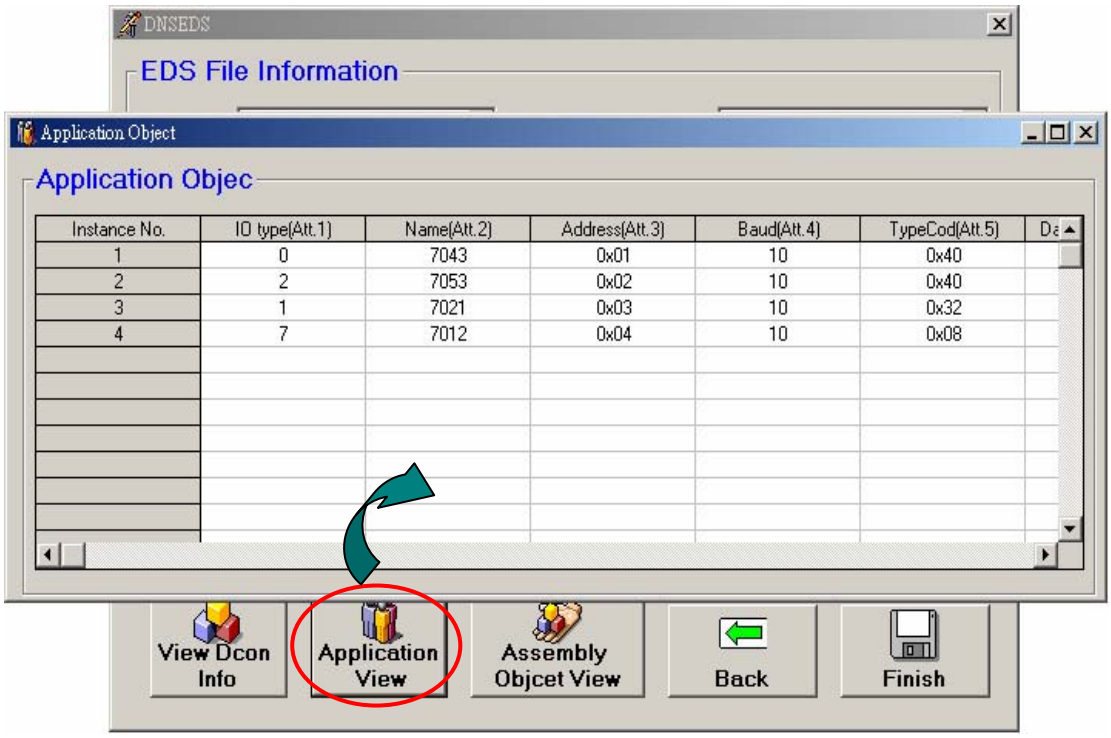


Fig 5-28 Application object information

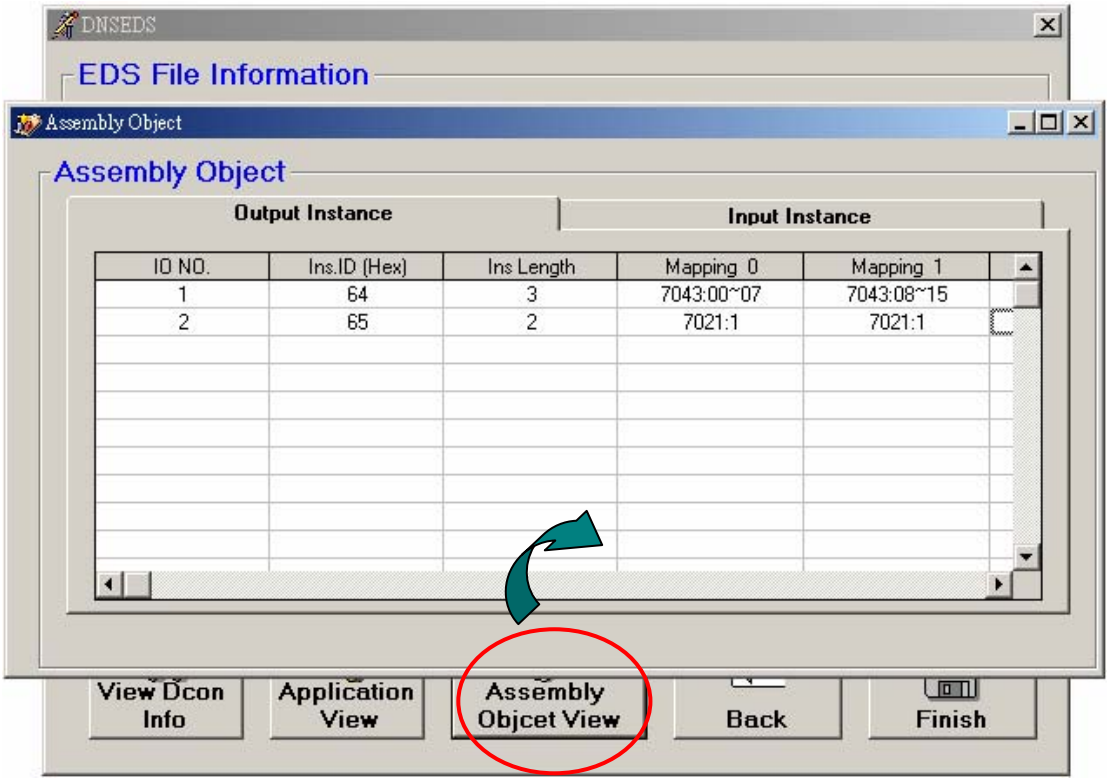


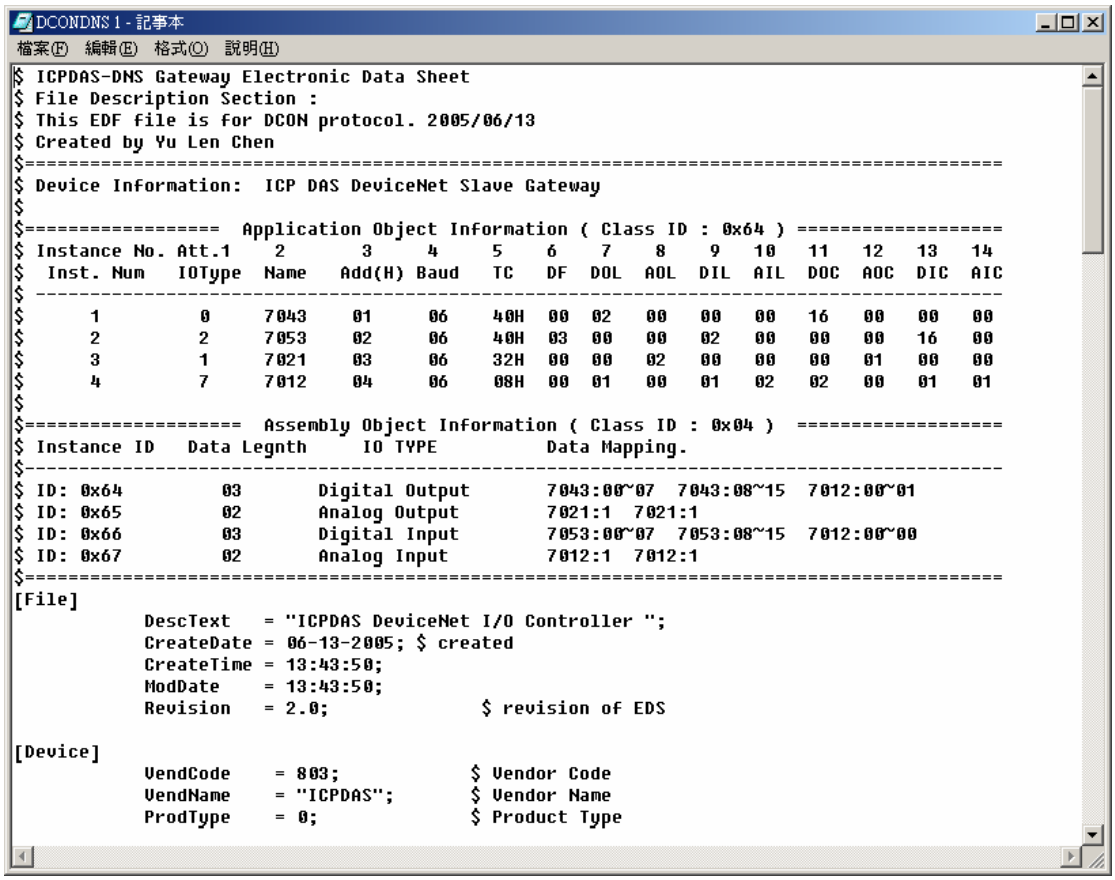
Fig 5-29 Assembly object information

Step 13: Click the “Finish” button to complete the DNS_DCON gateway configuration and the system will create the EDS file for users as Fig 5-30.

Fig 5-30 Press “Finish” button to finish and create an EDS file.

Step 14: When finish the process, the main window would pop-up. Then press the “Exit program” button to exit the program.

You can find the EDS file for the specific i-7241D. The file name is DCONDNS1.eds. The char “1” represent Node ID. Therefore, Users can apply the EDS file in the DeviceNet application as figure 5-30.



'Fig 5-30 Parts of EDS file

Note: There is some DCON information in the EDS file. Users can also see the DCON information form the EDS file.

Chapter 6 The components of Assembly Object

6.1 Components of the Assembly object

The Assembly Object binds the attributes of multiple objects, which allows data transfer to or from each object to be sent or received over a single connection. The i-7241D provides many assembly objects for users. The number of assembly objects is decided on by the i-7K/i-87K IO modules used. Every IO modules represents an application object instance. The i-7241D device would arrange the application instances in order by module address. Assembly object instances consist of these application object attributes. A sketch map is provided below.

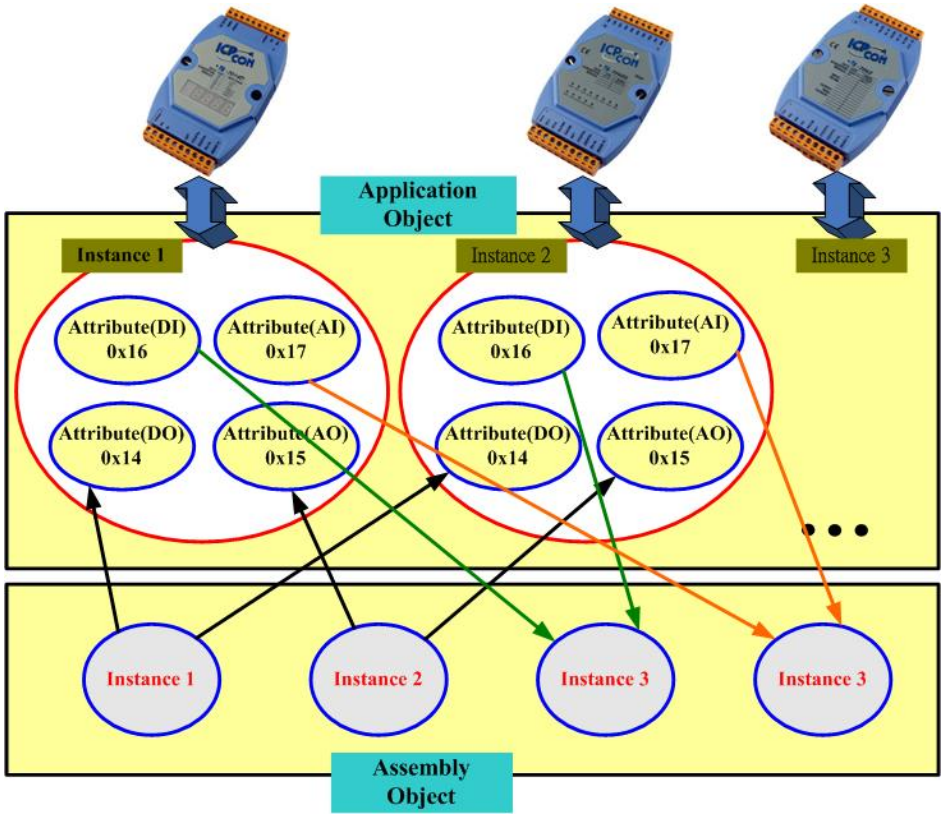


Figure 6-1 The assembly objects in i-7241D

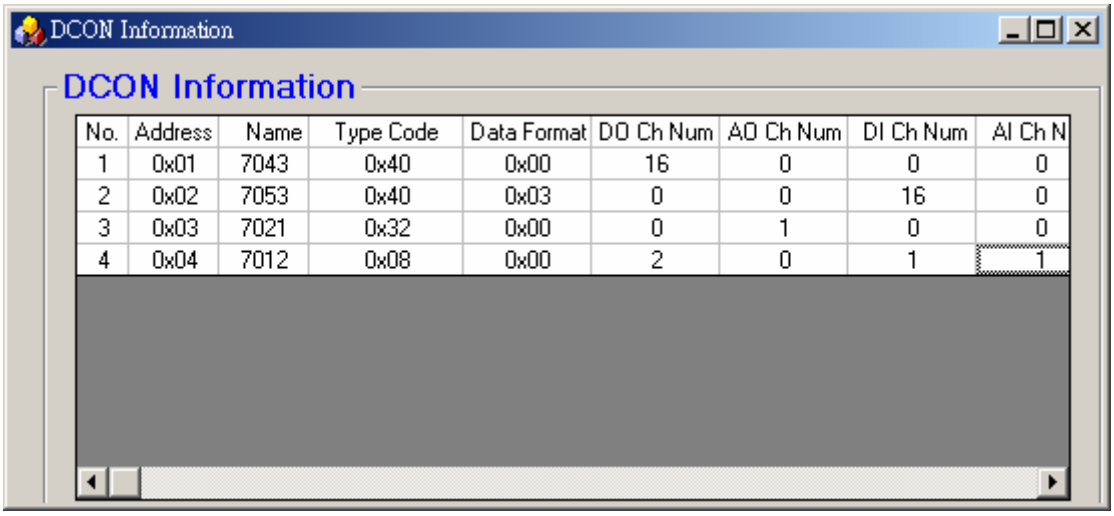
Note: There are some examples for users. Please refer to the next section 6.2.

6.2 Examples of Assembly objects in i-7241D

There are many IO examples related to the i-7241D in this section. These demos should help users to understand the i-7241D to a suitable degree.

Example 1:

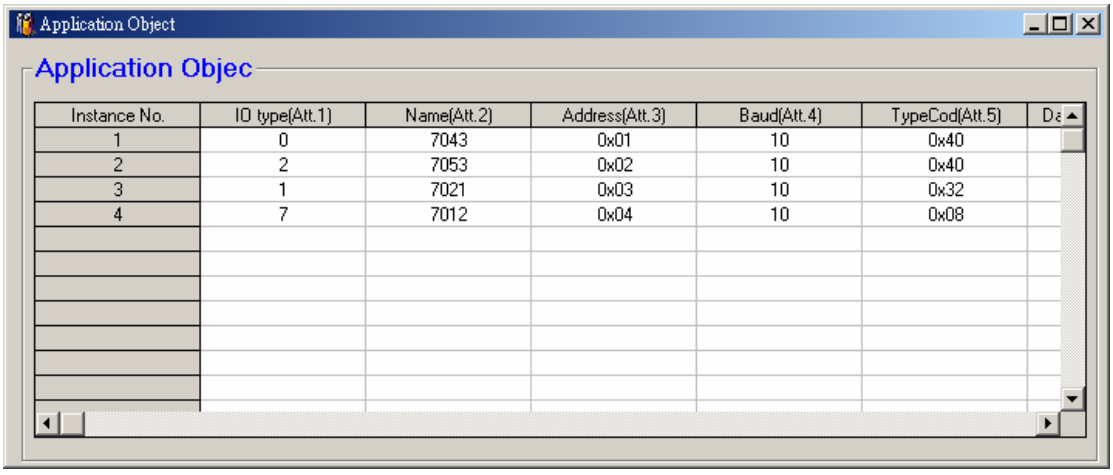
In the demo, apply I-7043 (address 0x01), I-7053 (address 0x02), I-7021 (address 0x03) and I-7012 (address 0x04) in the system. Users can scan the DCON modules from CAN Gateway Utility. In the demo, users can refer to the figure 6-2 to know the information of modules.



No.	Address	Name	Type Code	Data Format	DO Ch Num	AO Ch Num	DI Ch Num	AI Ch N
1	0x01	7043	0x40	0x00	16	0	0	0
2	0x02	7053	0x40	0x03	0	0	16	0
3	0x03	7021	0x32	0x00	0	1	0	0
4	0x04	7012	0x08	0x00	2	0	1	1

Figure 6-2 DCON modules information

And, the application object information is as figure 6-3.



Instance No.	IO type(Att.1)	Name(Att.2)	Address(Att.3)	Baud(Att.4)	TypeCod(Att.5)	Da
1	0	7043	0x01	10	0x40	
2	2	7053	0x02	10	0x40	
3	1	7021	0x03	10	0x32	
4	7	7012	0x04	10	0x08	

Figure 6-3 Information of application objects

Besides, the utility also shows the assembly objects information as the figure 6-4 below.

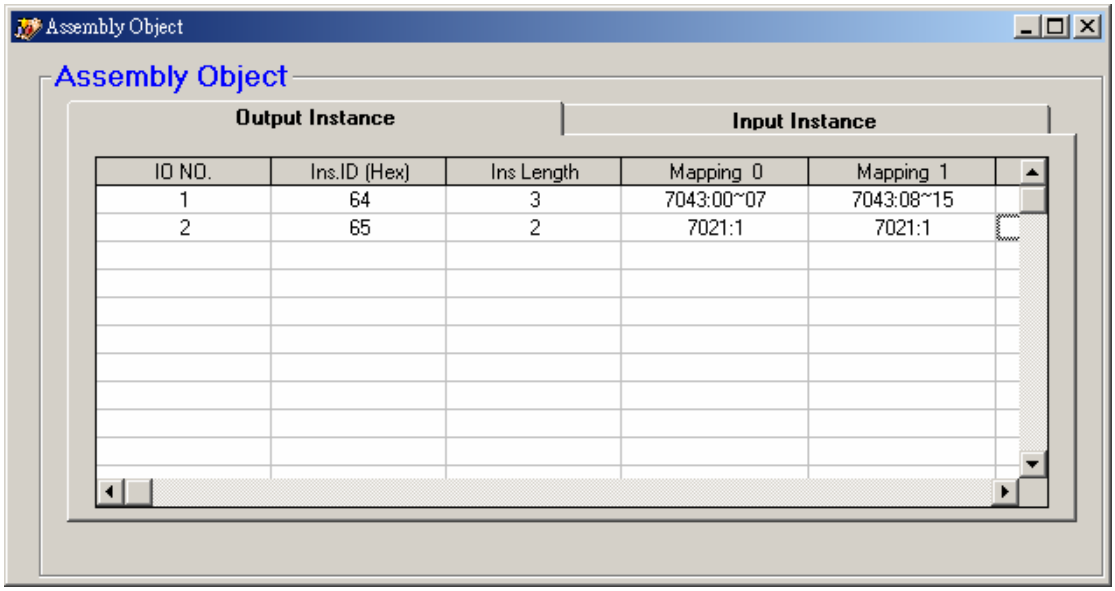


Figure 6-4 the information of Assembly objects

The i-7241D would arrange the application objects according to the table 6-1.

Application Instance ID	Module Address	Module name	DO Length(Byte)	AO Length(Byte)	DI Length(Byte)	AI Length(Byte)
0x01	0x01	7043	16	0	0	0
0x02	0x02	7053	0	0	16	0
0x03	0x03	7021	0	1	0	0
0x04	0x04	7012	2	0	1	1

Table 6-1 Application objects in the system

According the application objects, i-7241D would arranges the assembly objects as table 6-2.

Assembly Object Instance ID	Data Length(Byte)	Component modules
0x64	DO: 2+1=3	7043, 7012
0x65	AO: 2	7021
0x66	DI: 2+1	7053, 7021
0x67	AI: 2	7012

Table 6-2 Assembly objects in the system

Example2:

If users apply I-7066 (address 0x01), I-7033 (address 0x02), I-87024 (address 0x03) and I-87017 (address 0x04) in the system, the i-7241D would arrange these assembly and application objects according to the following table.

Application Instance ID	Module Address	Module name	DO Length(Byte)	AO Length(Byte)	DI Length(Byte)	AI Length(Byte)
0x01	0x01	7066	1	0	0	0
0x02	0x04	7033	0	0	0	6
0x03	0x05	87024	0	8	0	0
0x04	0x06	87017	0	0	0	16

Refer to the application object instances. The i-7241D would define the assembly object instances according to the following table.

Assembly Object Instance ID	Data Length(Byte)	Component modules
0x64	DO: 1	7066(ch0~ch6)
0x65	AO: 8	7016(ch0~ch3)
0x66	AI: 22	87017(ch0~ch7),7033(ch0~ch2)

This demo lacks of DI module. Therefore, the numbers of assembly object is 3. If the demo has DI modules, the number of assembly objects would be 4. And the assembly object instance ID of DI components is 0x65.

Example 3:

If users apply I-7017 (address 0x01), I-7043 (address 0x02), and I-7053 (address 0x04) in the system, the DeviceNet gateway would arrange these assembly and application objects according to the following table.

Application Instance	Module Address	Module name	DO Length(Byte)	AO Length(Byte)	DI Length(Byte)	AI Length(Byte)
0x01	0x01	7017	0	0	0	16
0x02	0x02	7043	2	0	0	0
0x03	0x04	7053	0	0	2	0

Refer to the application object instances. The DeviceNet gateway would define the assembly object instances according to the following table.

Assembly Object Instance ID	Data Length(Byte)	Component modules
0x64	DO: 2	7043
0x65	DI: 2	7053
0x66	AI: 16	7017

Note: In this example, the AO data length is 0. Therefore, we have three instances. One is assembly object instance (0x64) of DO. One is assembly object instance (0x65) of DI. The last is assembly object instance (0x66) of AI

Example 4:

If users apply I-7017 (address 0x01) and I-7053 (address 0x04) into the system, the i-7241D would arrange these assembly and application objects according to the following table.

Application Instance ID	Module Address	Module name	DO Length (Byte)	AO Length (Byte)	DI Length (Byte)	AI Length (Byte)
0x01	0x01	7017	0	0	0	16
0x02	0x04	7053	0	0	2	0

Refer to the application object instances. The DeviceNet gateway would define the assembly object instances according to the following table.

Assembly Object Instance ID	Data Length(Byte)	Component modules
0x64	DI: 2	7053
0x65	AI: 16	7017

Note: In this example, the total data lengths for DO and AO are 0. Therefore, the assembly objects instances are 2. The one is the assembly object instance (0x64) for DI. The other is the assembly object instance (0x65) for AI.

Chapter 7 DeviceNet Communication Set

7.1 DeviceNet Communication Set Introduction

Thei-7241D is a “Group 2 Only Slave” device, and supports the “Predefined Master/slave Connection Set”. To communicate with this device, the process for establishing a connection is important. In addition, we provide some examples on how to access IO modules.

The CAN Identifier Fields associated with the Predefined Master/Slave Connection Set for the i-7241D are given in the table below. The below table defines the Identifiers that are to be used with all the connection based messaging involved in the Predefined Master/Slave Connection Set for the i-7241D.

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	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	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	

1	1	1	1	1	Group 4 Message ID	Group 4 Messages	000 – 3ff
1	1	1	1	1	2C	Communication Faulted Response Message	
1	1	1	1	1	2D	Communication Faulted Request Message	

The following table lists the Error Codes that may be present in the General Error Code field of an Error Response message.

Error code

Error Condition	General Error code (Hex)	Additional Error Condition	Additional Error Code (Hex)
Resource unavailable	02	Invalid allocation choice	02
		Invalid Unconnected request	03
		Poll After COS_CYCLIC	04
Service not support	08	None	FF
Invalid attribute value	9	None	FF
Already in requested mode/state	0B	None	FF
Object state conflict	0C	Class specific error	01
Attribute not settable	0E	None	FF
Privilege violation	0F	None	FF
Device state conflict	10	None	FF
Reply data too large	11	None	FF
Not enough data	13	None	FF

Attribute not supported	14	None	FF
Too much data	15	None	FF
Object does not exist	16	None	FF
FRAGMENTATION EQ	17	None	FF
Invalid parameter	20	None	FF

The following steps may be useful to those users who would like to implement their DeviceNet applications using the command set.

1. **Requests the use of the Predefined Master/Slave Connection Set.**

2. **Use the Master's Explicit Request Messages to set expected_packet _rate attribute of IO connection and make I/O Connection Object State established.**

.

3. **There are two ways to access IO modules. The first method is by way of the IO connection object. The other is by using an explicit message to set/get the IO attribute of application object.**

4. **Release the use of the Predefined Master/Slave Connection Set.**

7.2 Examples on the DeviceNet communication set

7.2.1 Request the use of the Predefined Master/Slave Connection Set

An unconnected explicit messaging request sent by the Master node via a destination node's Group 2 Only Unconnected Explicit Request Message to requests the use of the Predefined Master/Slave Connection set. The example shows how to use these connection sets. In this demo, the Master establishes the Explicit Message, Poll IO and Bit-Strobe IO connections.

The figure below shows the Group 2 Only Unconnected Explicit connection Identifier Fields.

IDENTIFIER BITS											IDENTITY USAGE	HEX
10	9	8	7	6	5	4	3	2	1	0		RANGE
1	0	Source MAC ID						0	1	1	Slave's Explicit/ Unconnected Response Messages	
1	0	Destination MAC ID						1	1	0	Group 2 Only Unconnected Explicit Request Messages	

Note: i-7241D(Slave): Node ID=0x09, Master node ID=0x0A

Master(MAC ID =0x0A) Slave (MAC ID =0x09)

```
      _ Group 2 Message
      |      _ Destination MAC ID=0x09
      |      |      _ Message ID =6
      |      |      |            _ Frag=0. Transaction ID=0. Source MAC ID= 0x0A
      |      |      |            |     _ Service=Allocate_Master/Slave_Connection_Set Request
      |      |      |            |     |   _ Class ID=3
      |      |      |            |     |   |   _ Instance ID=1
      |      |      |            |     |   |   |   _ Allocation Choice=Explicit, Polled & Bit Strobe
      |      |      |            |     |   |   |   |     _ Allocator's MAC ID=0x0A
ID=10 001001 110. Data= 0A 4B 03 01 07 0A
```



```
      _ Group 2 Message
      |      _ Source MAC ID=0x09
      |      |      _ Message ID =3
      |      |      |            _ Frag=0. Transaction ID=0. Destination MAC ID= 0x0A
      |      |      |            |     _ Service=Allocate_Master/Slave_Connection_Set Response
      |      |      |            |     |   _Connection Message Body Format = DeviceNet (8/8)
      |      |      |            |     |   |
      |      |      |            |     |   |
      |      |      |            |     |   |
ID=10 001001 011. Data= 0A CB 00
```

7.2.2 How to apply the Poll IO connection

Poll Command and Response messages move any amount of I/O data between a Master and its Polled Slaves. The example explains how to apply the Poll IO connection in the DeviceNet application.

Note: i-7241D: Node ID=0x09, Master node ID=0x0A

The figure below shows Poll I/O connection Identifier Fields.

IDENTIFIER BITS											IDENTITY USAGE	HEX
10	9	8	7	6	5	4	3	2	1	0		RANGE
1	0	Destination MAC ID						1	0	1	Master's I/O Poll Command/Change of State/Cyclic Message	
1	0	Source MAC ID						0	1	1	Slave's Explicit/ Unconnected Response Messages	
1	0	Destination MAC ID						1	1	0	Group 2 Only Unconnected Explicit Request Messages	
1	0	Destination MAC ID						1	0	0	Master's Explicit Request Messages	
0	1	1	1	1	Source MAC ID						Slave's I/O Poll Response Message	

1. Request the use of the Predefined Master/Slave Connection set

Master (MAC ID =0x0A) Slave (MAC ID =0x09)

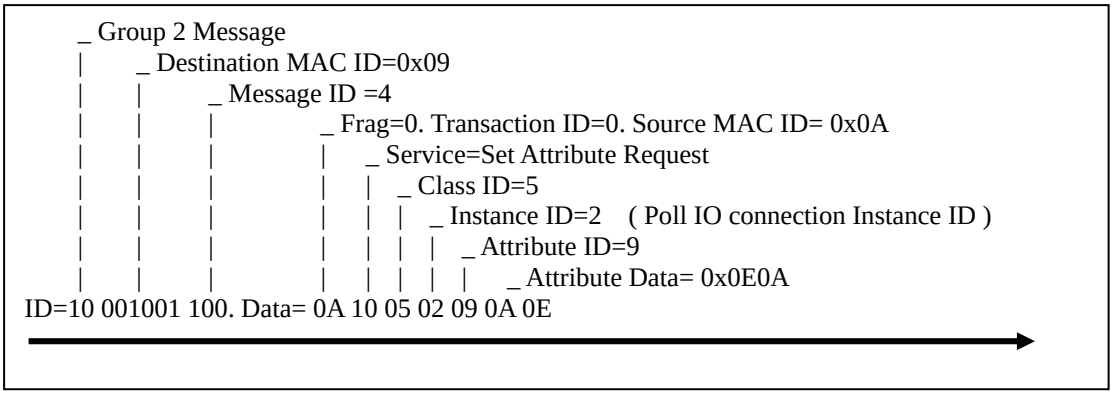
```
_ Group 2 Message
|   _ Destination MAC ID=0x09
|   |   _ Message ID =6
|   |   |   _ Frag=0. Transaction ID=0. Source MAC ID= 0x0A
|   |   |   |   _ Service=Allocate_Master/Slave_Connection_Set Request
|   |   |   |   |   _ Class ID=3
|   |   |   |   |   |   _ Instance ID=1
|   |   |   |   |   |   |   _ Allocation Choice=Explicit, Polled & Bit Strobe
|   |   |   |   |   |   |   |   _ Allocator's MAC ID=0x0A
ID=10 001001 110. Data= 0A 4B 03 01 07 0A
```

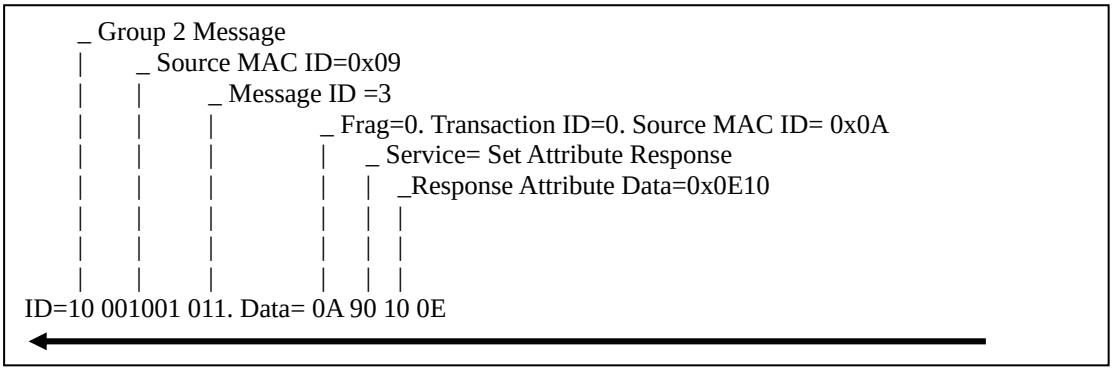


```
_ Group 2 Message
|   _ Source MAC ID=0x09
|   |   _ Message ID =3
|   |   |   _ Frag=0. Transaction ID=0. Destination MAC ID= 0x0A
|   |   |   |   _ Service=Allocate_Master/Slave_Connection_Set Response
|   |   |   |   |   _Connection Message Body Format = DeviceNet (8/8)
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
ID=10 001001 011. Data= 0A CB 00
```

2. Apply the Master's Explicit Request Messages to set the expected_packet _rate attribute for the IO connection and let the I/O Connection Object State established.

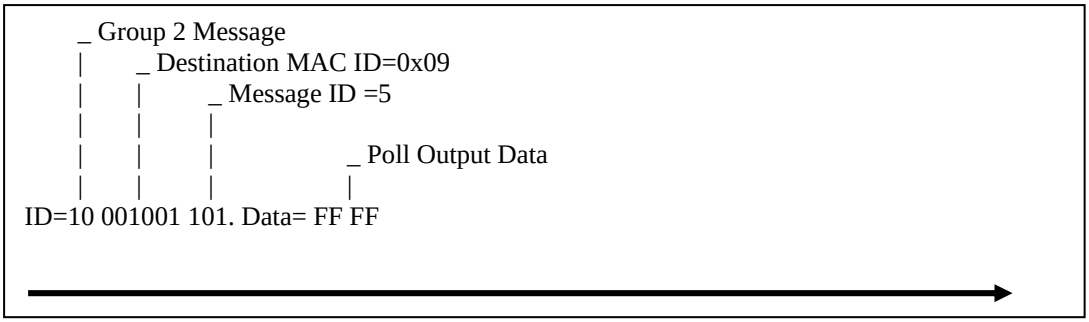
Master (MAC ID =0x0A) Slave (MAC ID =0x09)

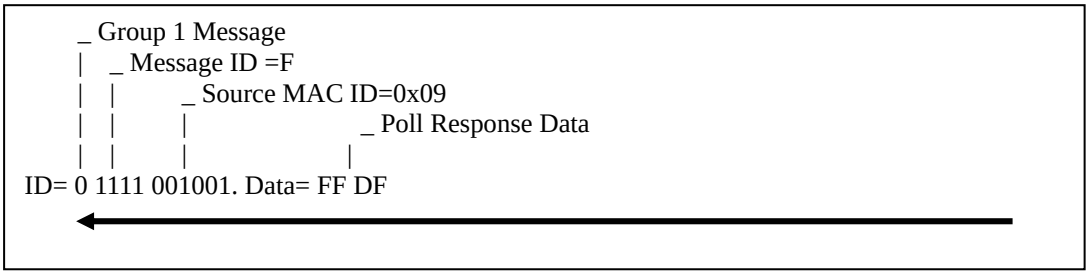




3. Apply the Poll IO connection to access the I/O modules

Master (MAC ID =0x0A) Slave (MAC ID =0x09)





7.2.3 The Bit-Strobe IO connection example

Bit-Strobe Command and Response messages rapidly move small amounts of I/O data between a master and its Bit-Strobe slaves.

IDENTIFIER BITS											IDENTITY USAGE	HEX RANGE
10	9	8	7	6	5	4	3	2	1	0		
0	1	1	1	0	Source MAC ID						Slave's I/O Bit–Strobe Response Message	
1	0	Source MAC ID						0	0	0	Master's I/O Bit–Strobe Command Message	
1	0	Source MAC ID						0	1	1	Slave's Explicit/ Unconnected Response Messages	
1	0	Destination MAC ID						1	1	0	Group 2 Only Unconnected Explicit Request Messages	
1	0	Destination MAC ID						1	0	0	Master's Explicit Request Messages	

Note: i-7241D: Node ID=0x09, Master node ID=0x0a

1. Requests the use of the Predefined Master/Slave Connection set

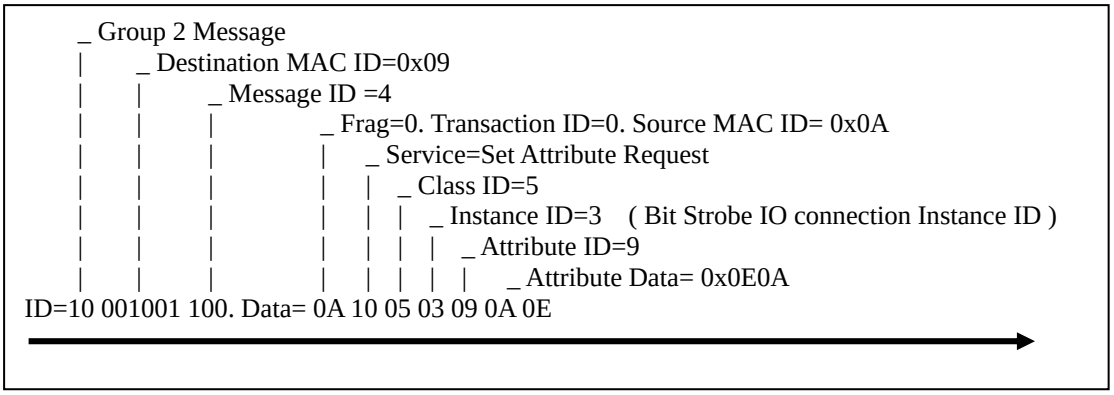
Master (MAC ID =0x0A) Slave (MAC ID =0x09)

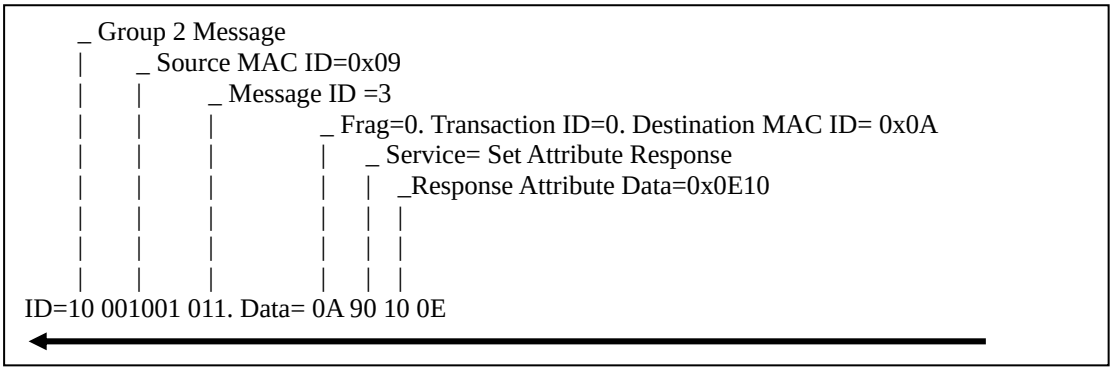
```
      _ Group 2 Message
      |      _ Destination MAC ID=0x09
      |      |       _ Message ID =6
      |      |       |           _ Frag=0. Transaction ID=0. Source MAC ID= 0x0A
      |      |       |           |     _ Service=Allocate_Master/Slave_Connection_Set Request
      |      |       |           |     |   _ Class ID=3
      |      |       |           |     |   |   _ Instance ID=1
      |      |       |           |     |   |   |   _ Allocation Choice=Explicit, Polled & Bit Strobe
      |      |       |           |     |   |   |   |     _ Allocator's MAC ID=0x0A
ID=10 001001 110. Data= 0A 4B 03 01 07 0A
```

```
      _ Group 2 Message
      |      _ Source MAC ID=0x09
      |      |       _ Message ID =3
      |      |       |           _ Frag=0. Transaction ID=0. Destination MAC ID= 0x0A
      |      |       |           |     _ Service=Allocate_Master/Slave_Connection_Set Response
      |      |       |           |     |   _Connection Message Body Format = DeviceNet (88)
      |      |       |           |     |   |
      |      |       |           |     |   |
      |      |       |           |     |   |
ID=10 001001 011. Data= 0A CB 00
```

2. Apply the Master's Explicit Request Messages to set the expected_packet_rate attribute for the IO connection and let the I/O Connection Object State established.

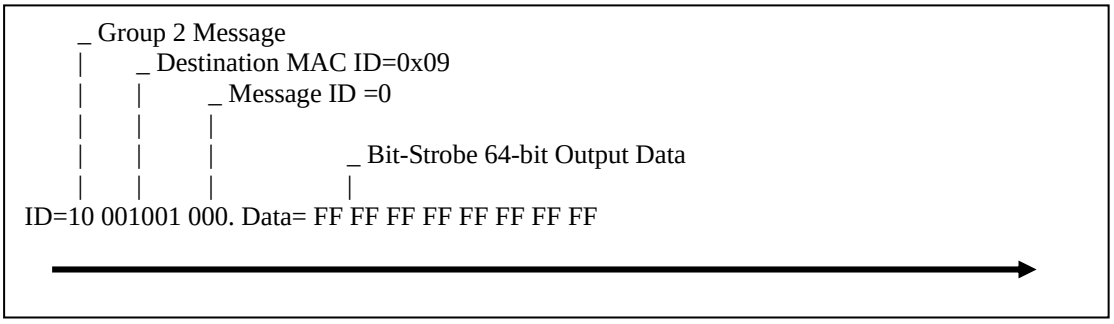
Master (MAC ID =0x0A) Slave (MAC ID =0x09)

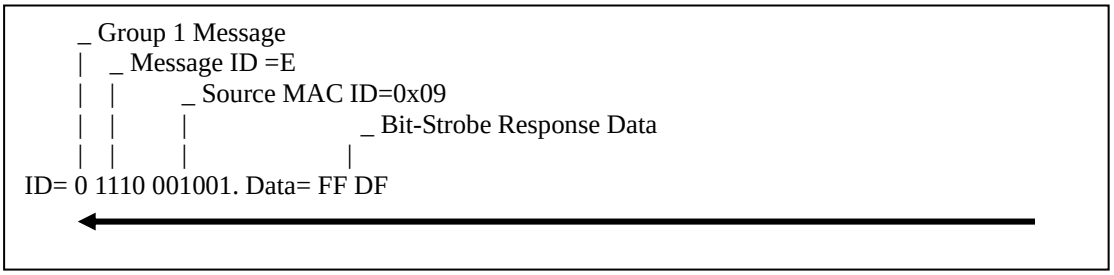




3. Apply the Bit-Strobe IO connection to access the I/O modules

Master (MAC ID =0x0A) Slave (MAC ID =0x09)





7.2.4 COS/Cyclic IO with Acknowledge connection example

The Change of State/Cyclic connection sets use the connection instance 2 (the polled connection instance) for master to slave data production and slave to master acknowledgment. Connection instance 4 is used for slave to master data production and master to slave acknowledgment. If a device does not support the poll, and has no support for output data, connection instance 2 does not need to be installed.

IDENTIFIER BITS											IDENTITY USAGE	HEX
10	9	8	7	6	5	4	3	2	1	0		RANGE
0	1	1	0	1	Source MAC ID						Slave's I/O Change of State or Cyclic 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	
1	0	Destination MAC ID						1	1	0	Group 2 Only Unconnected Explicit Request Messages	
1	0	Destination MAC ID						1	0	0	Master's Explicit Request Messages	

Note: i-7241D: Node ID=0x09, Master node ID=0x0a

1. Requests the use of the Predefined Master/Slave Connection set

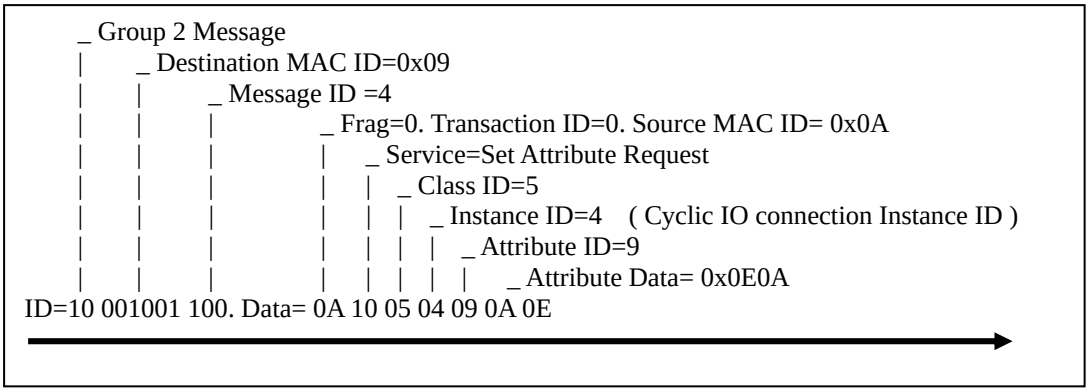
Master (MAC ID =0x0A) Slave (MAC ID =0x09)

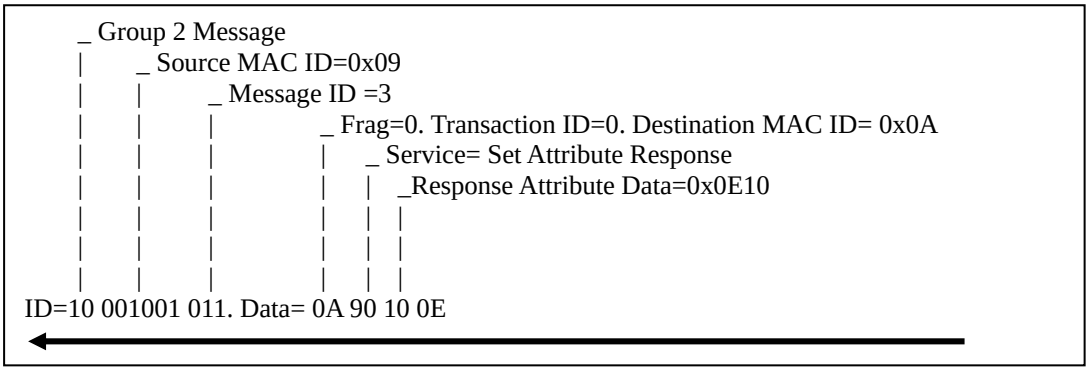
```
_ Group 2 Message
|   _ Destination MAC ID=0x09
|   |   _ Message ID =6
|   |   |   _ Frag=0. Transaction ID=0. Source MAC ID= 0x0A
|   |   |   |   _ Service=Allocate_Master/Slave_Connection_Set Request
|   |   |   |   |   _ Class ID=3
|   |   |   |   |   |   _ Instance ID=1
|   |   |   |   |   |   |   _ Allocation Choice=Explicit &Cyclic
|   |   |   |   |   |   |   |   _ Allocator's MAC ID=0x0A
ID=10 001001 110. Data= 0A 4B 03 01 21 0A
```

```
_ Group 2 Message
|   _ Source MAC ID=0x09
|   |   _ Message ID =3
|   |   |   _ Frag=0. Transaction ID=0. Destination MAC ID= 0x0A
|   |   |   |   _ Service=Allocate_Master/Slave_Connection_Set Response
|   |   |   |   |   _Connection Message Body Format = DeviceNet (8/8)
|   |   |   |   |   |
|   |   |   |   |   |
|   |   |   |   |   |
ID=10 001001 011. Data= 0A CB 00
```

2. Apply the Master's Explicit Request Messages to set the expected_packet_rate attribute for the IO connection and let the I/O Connection Object State established.

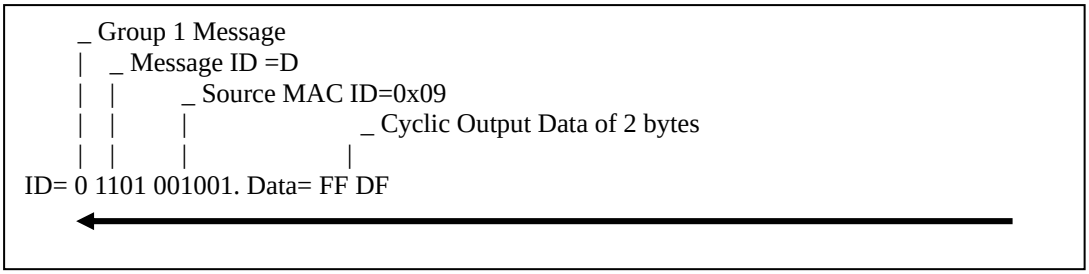
Master (MAC ID =0x0A) Slave (MAC ID =0x09)



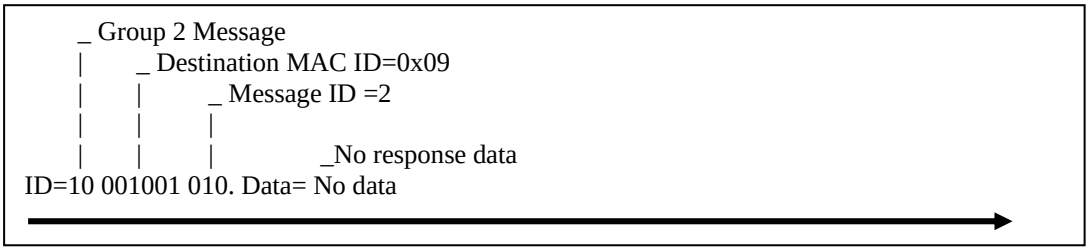


3. Start to transfer IO data via the Cyclic IO connection.

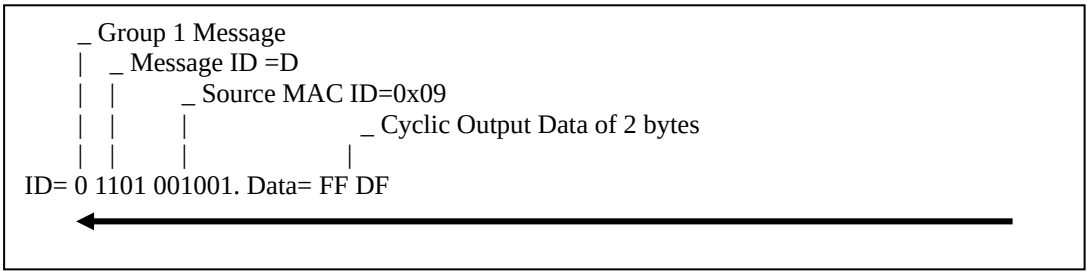
Slave (MAC ID =0x09)



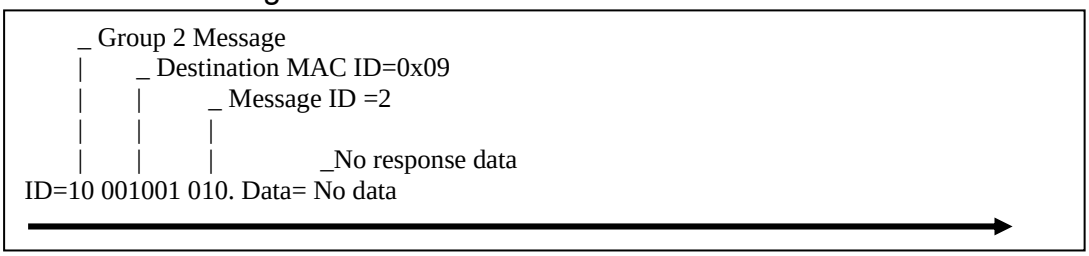
Master Acknowledge



Slave (MAC ID =0x09)



Master Acknowledge

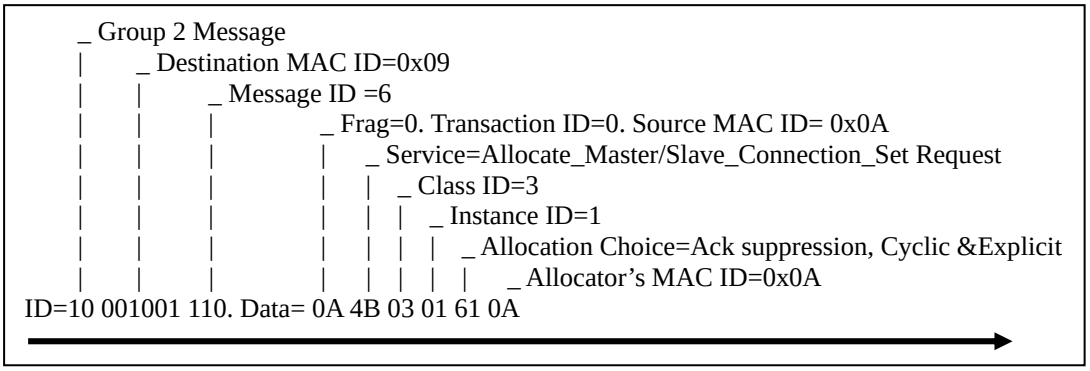


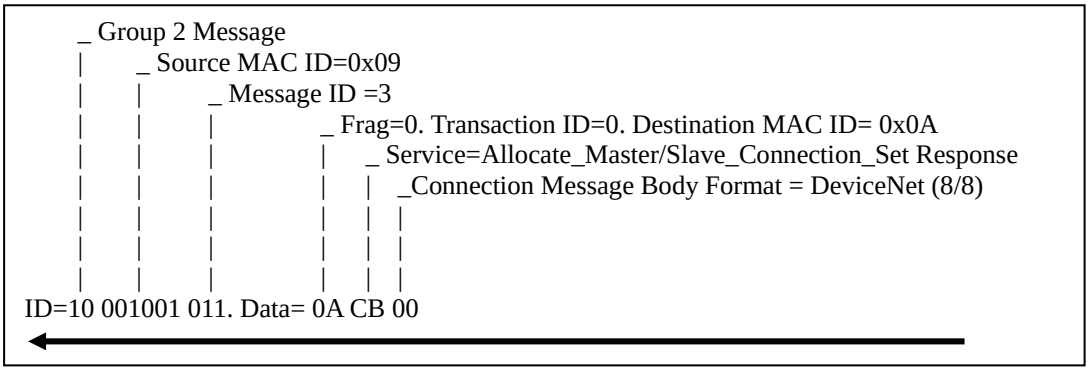
7.2.5 COS/Cyclic IO without Acknowledge connection example

Note: i-7241D: Node ID=0x09, Master node ID=0x0A

1. Requests the use of the Predefined Master/Slave Connection set

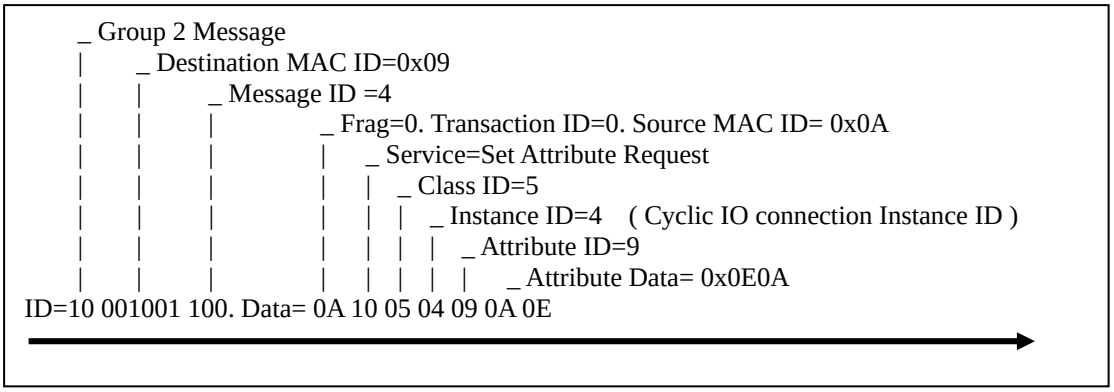
Master (MAC ID =0x0A) Slave (MAC ID =0x09)

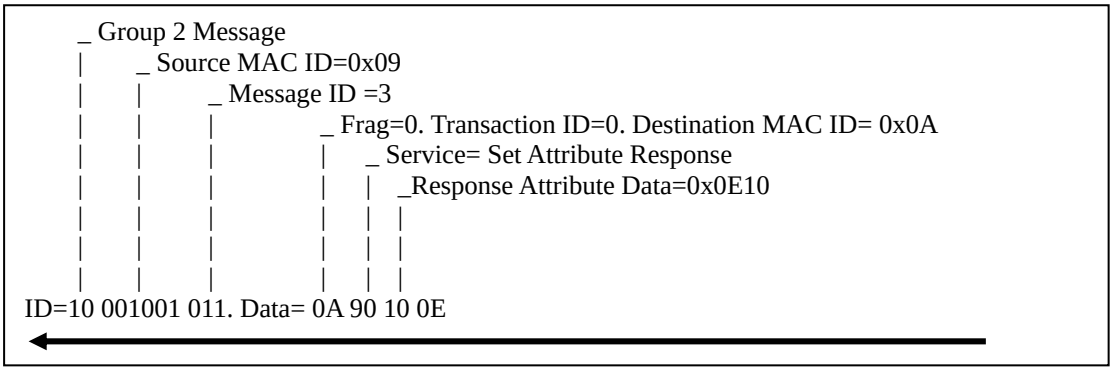




2. Apply the Master's Explicit Request Messages to set the expected_packet_rate attribute for the IO connection and let the I/O Connection Object State established.

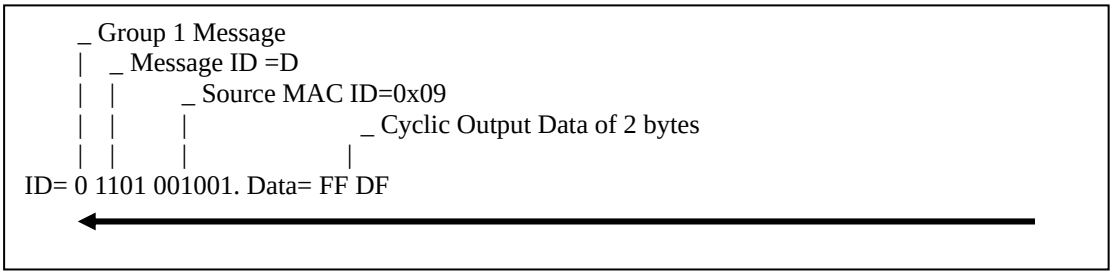
Master (MAC ID =0x0A) Slave (MAC ID =0x09)



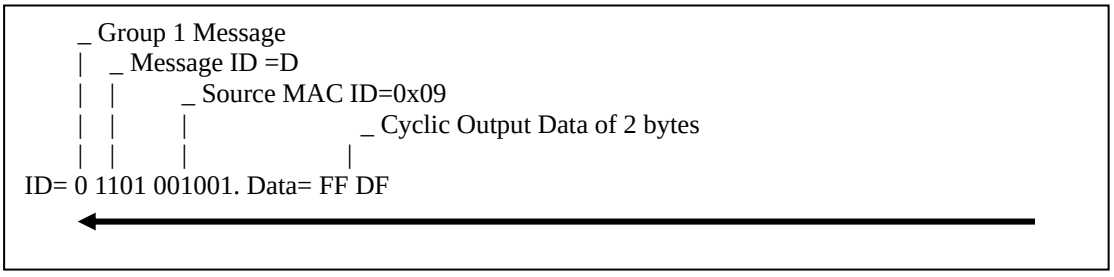


3. Slave transmits data cyclically.

Slave (MAC ID =0x09)



Slave (MAC ID =0x09)



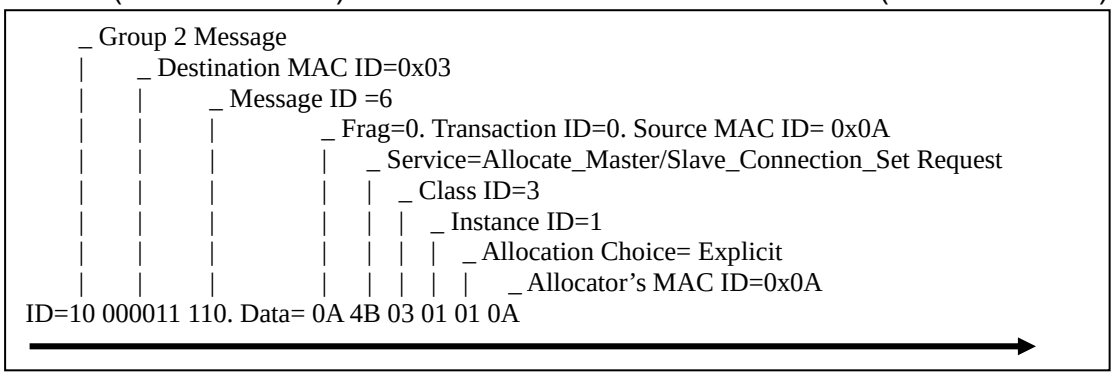
7.2.6 Change Node ID example

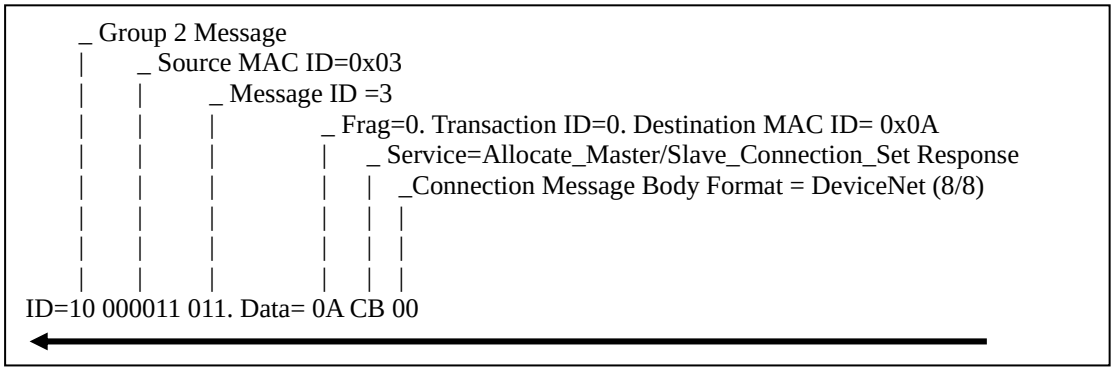
If necessarily, users must change MAC ID of i-7241D. For example, the i-7241D duplicates ID faults. There could be the same ID of device in the network. Therefore, the i-7241D supports the ability of on-line change the baud rate of CAN. Please refer to the following example.

Note: i-7241D: Node ID=0x03, Master node ID=0x0A

1. Requests the use of the Predefined Master/Slave Connection set

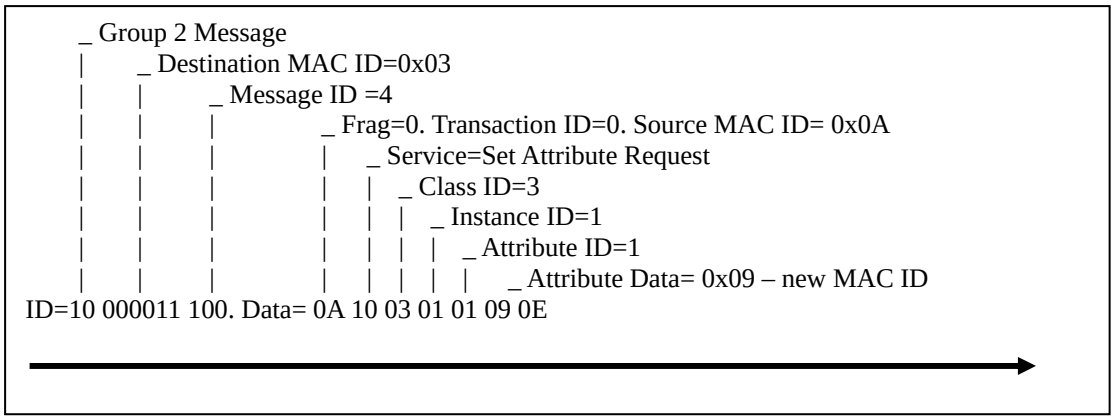
Master (MAC ID =0x0A) Slave (MAC ID =0x03)

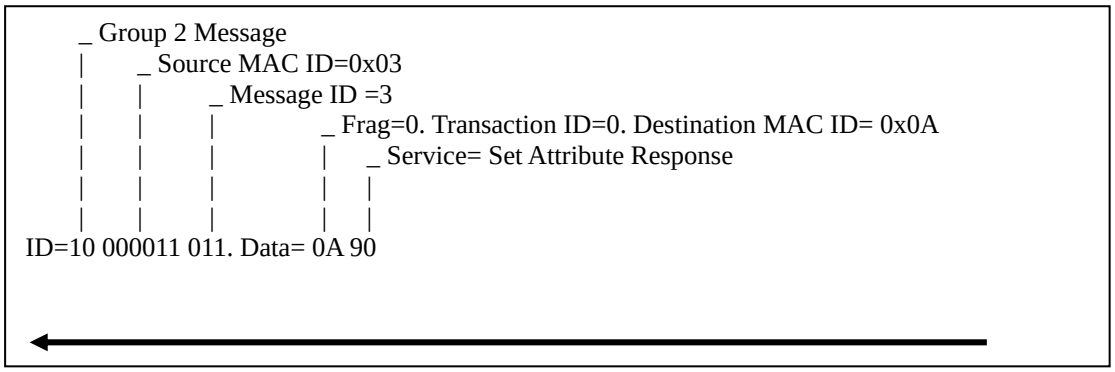




2. Apply the Master's Explicit Request Messages to change the Slave MAC ID.

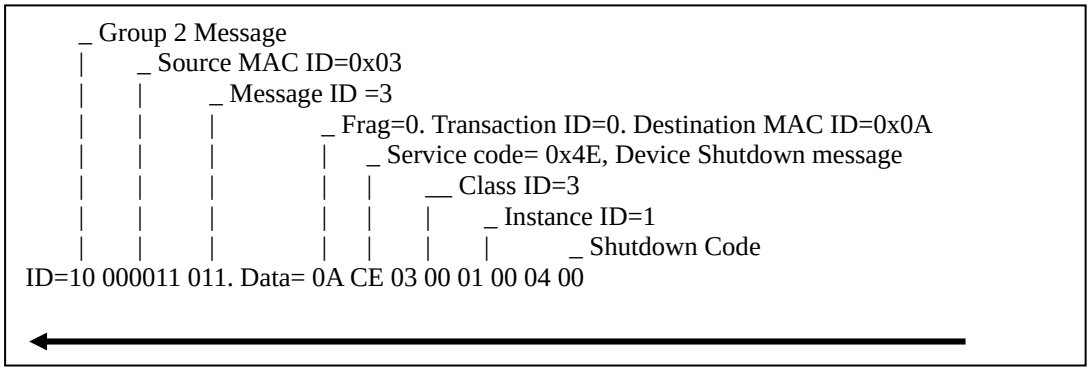
Master (MAC ID =0x0A) Slave (MAC ID =0x03)





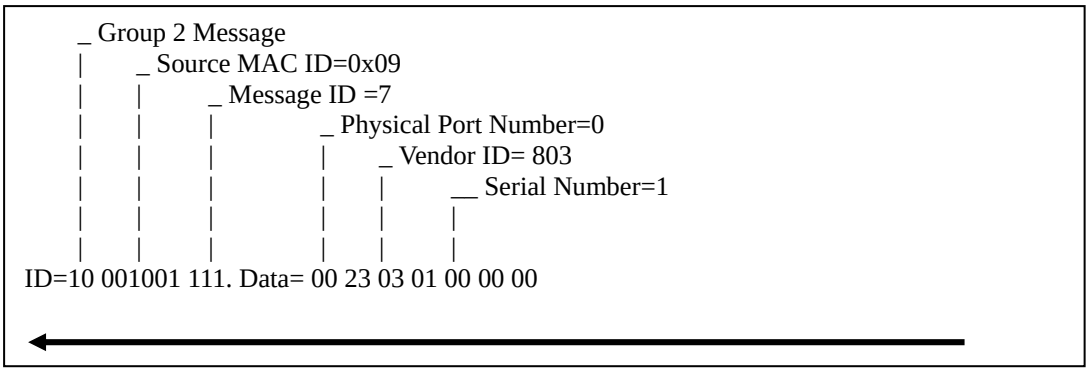
3. After i-7241D changes the MAC ID, it would send out the shutdown message.

Slave (MAC ID =0x03)

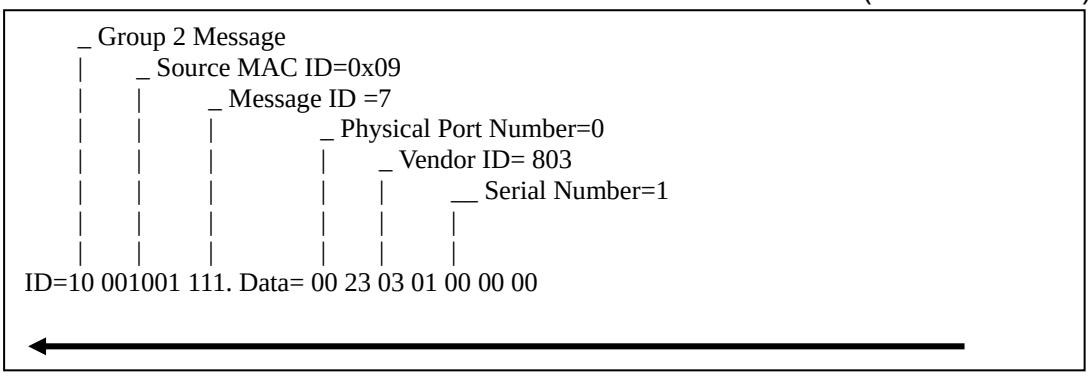


4. Then the i-7241D would send out the duplicated message.

Slave (MAC ID =0x09)



Slave (MAC ID =0x09)

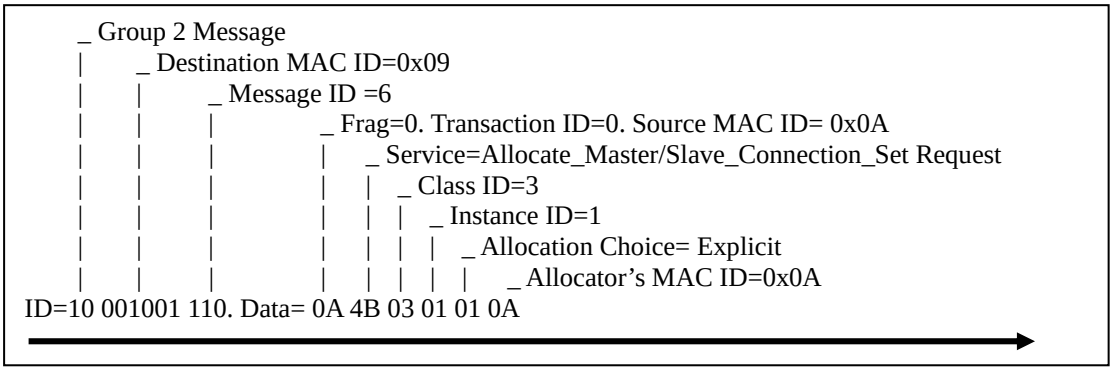


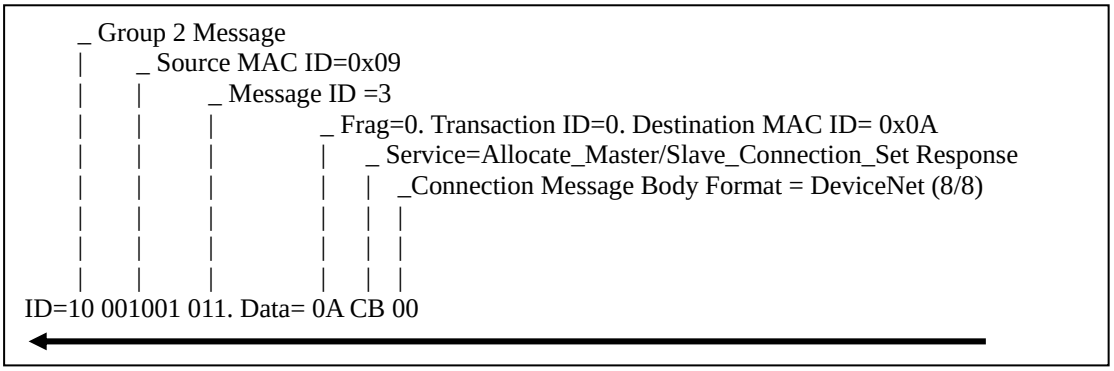
7.2.7 Change CAN Baud Rate

i-7241D provides the ability for changing the baud rate of CAN. But after finishing change baud rate, the new baud rate would be not effective immediately. Must to request reset service to reset i-7241D.

1. Requests the use of the Predefined Master/Slave Connection set

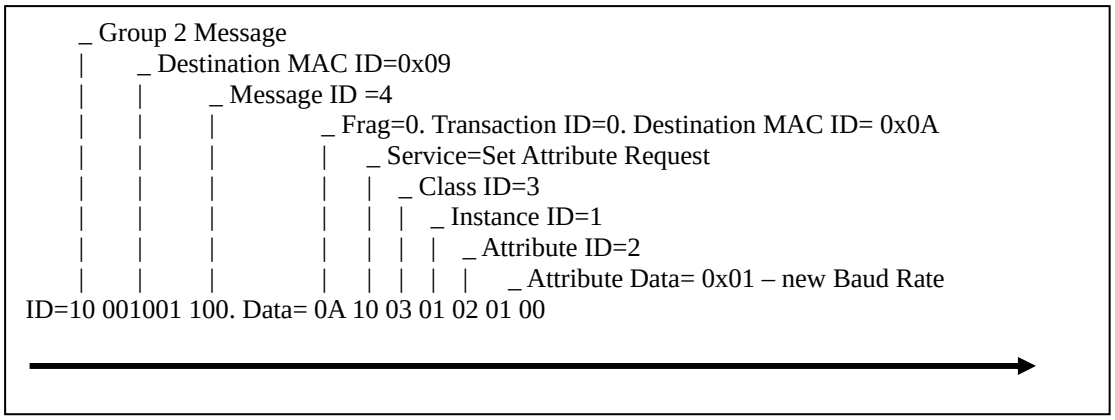
Master (MAC ID =0x0A) Slave (MAC ID =0x09)

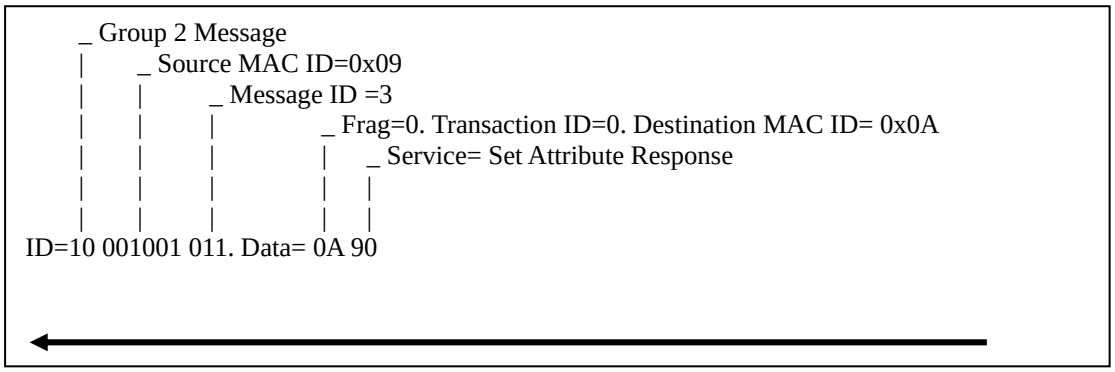




2. Apply the the Master's Explicit Request Messages to change the Slave's Baud Rate.

Master (MAC ID =0x0A) Slave (MAC ID =0x09)





Note: After finishing the changing to the MAC ID on-line, the i-7241D will send a shutdown message and reset. However if users want to change the baud rate of CAN bus, they must send the reset service to reset the i-7241D. Then the new the baud rate of the i-7241D will become effective.

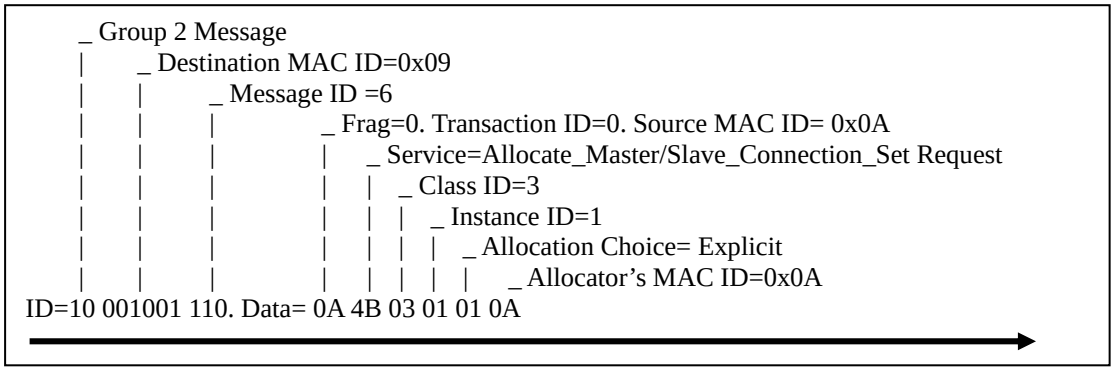
7.2.8 Reset Service

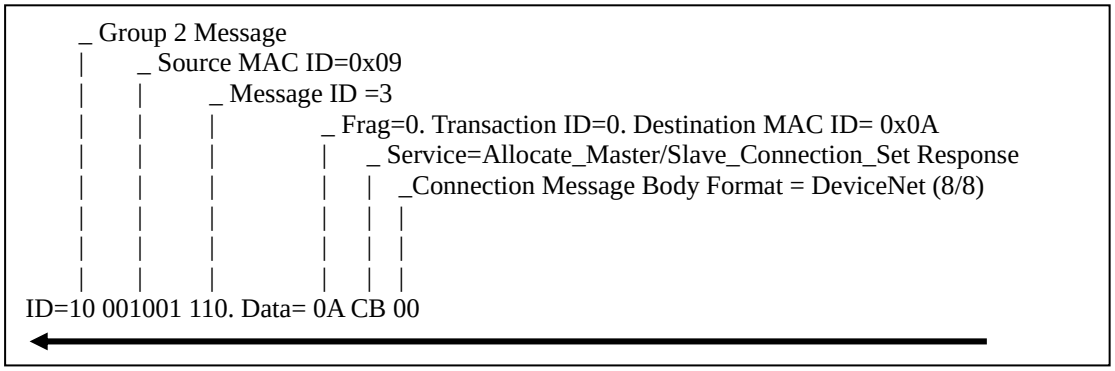
This service can reset device. If users change baud rate of CAN, i-7241D must be reset.

Note: i-7241D: Node ID=0x09, Master node ID=0x0a

1. Requests the use of the Predefined Master/Slave Connection set

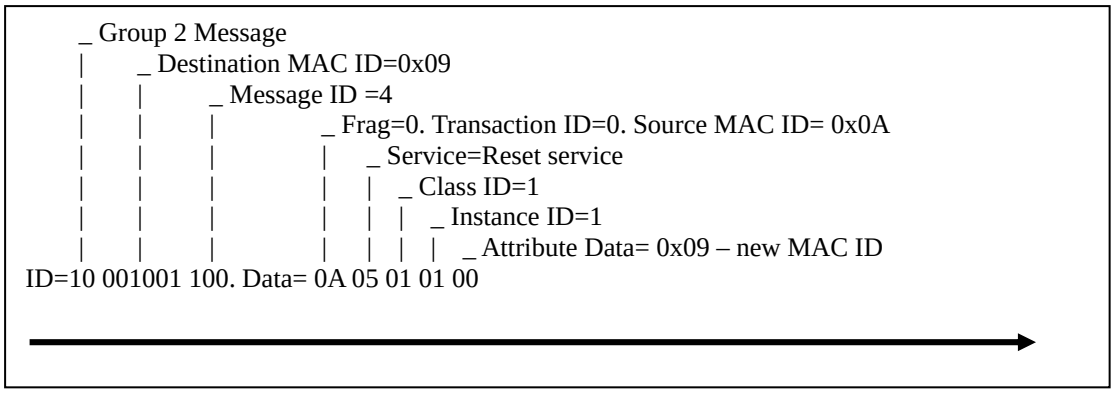
Master (MAC ID =0x0A) Slave (MAC ID =0x09)

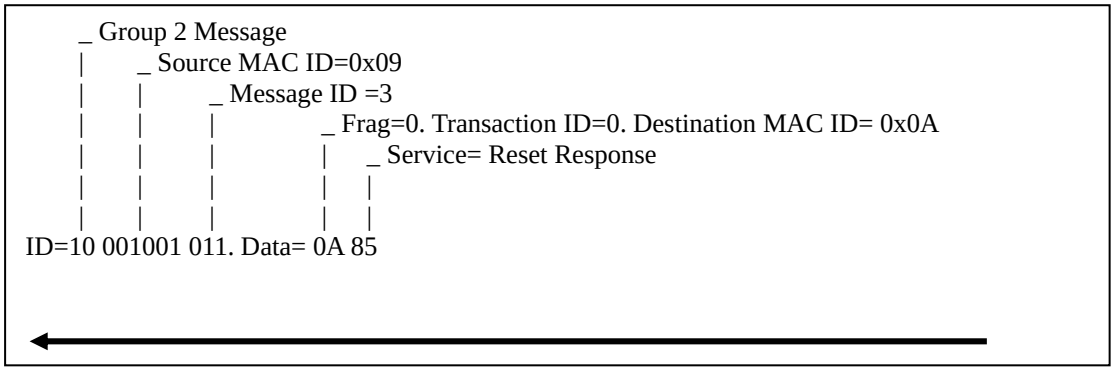




2. Apply the Master's Explicit Request Messages to set the Identify object. The service ID (0x05) is reset service.

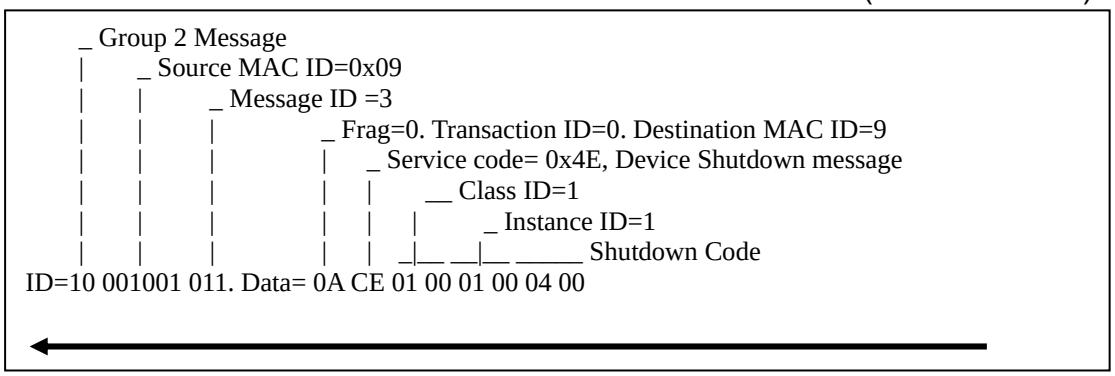
Master (MAC ID =0x0A) Slave (MAC ID =0x09)





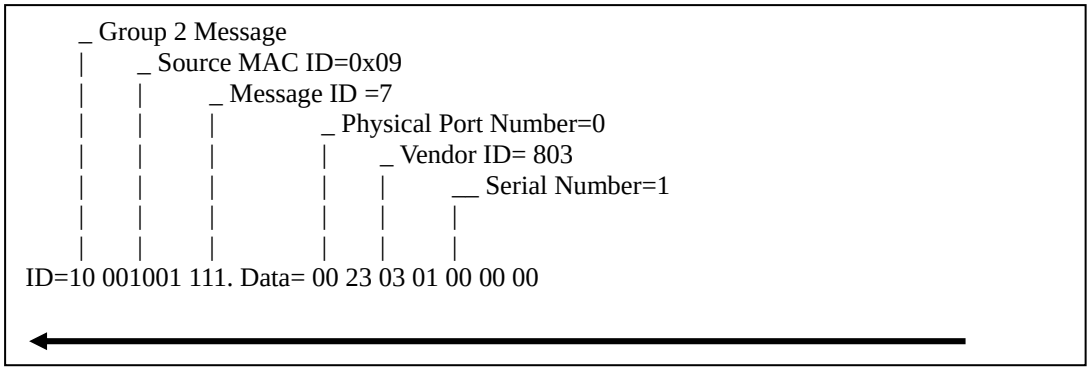
Send the shutdown message to the CAN bus.

Slave (MAC ID =0x09)

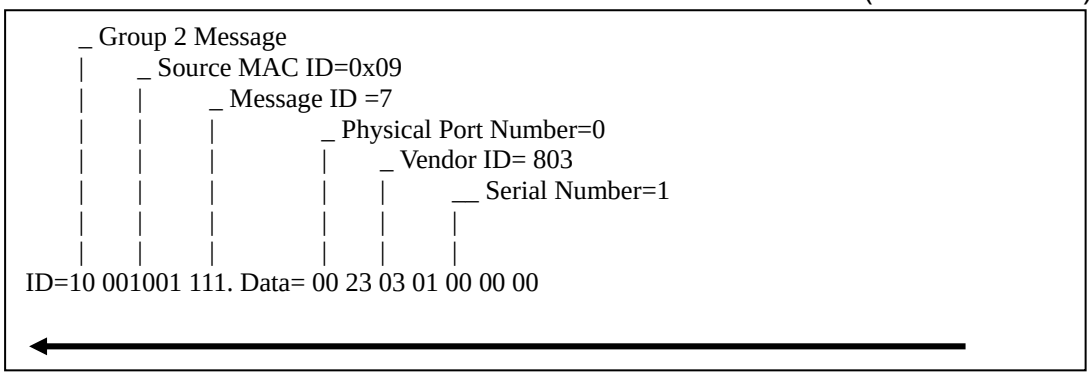


3. After i-7241D sends out the shutdown message, it will reset and send Duplicated ID messages.

Slave (MAC ID =0x09)



Slave (MAC ID =0x09)



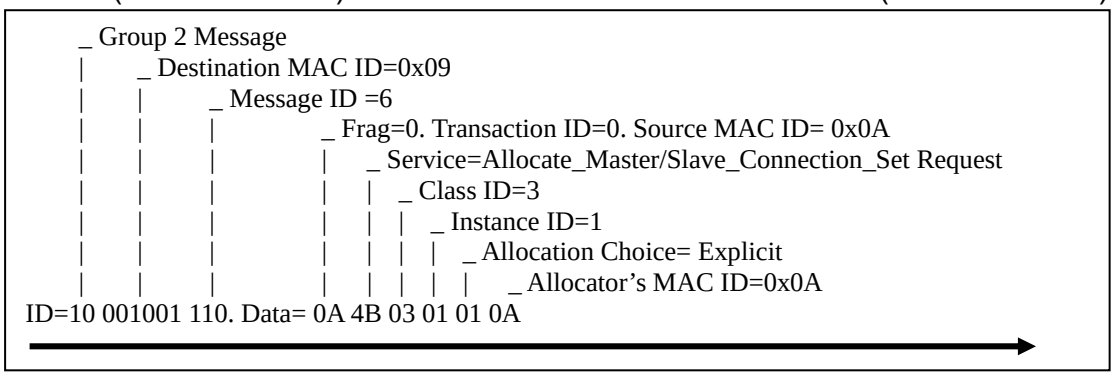
7.2.9 DEVICE HEARTBEAT

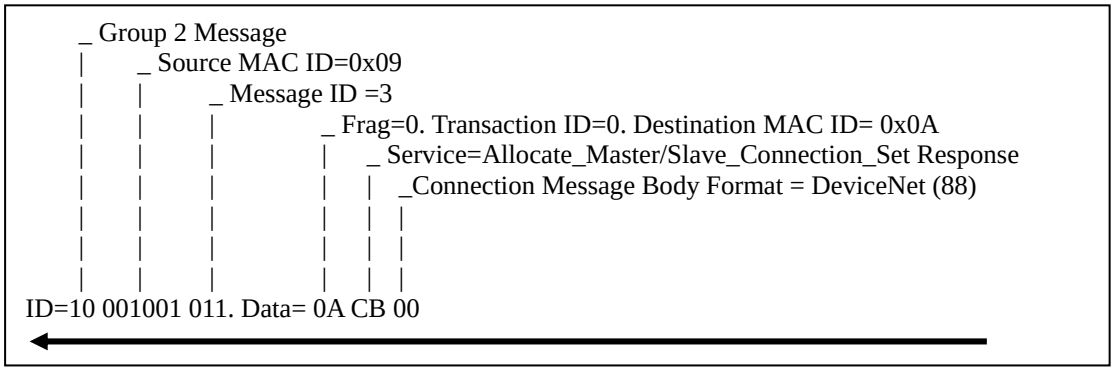
This message broadcasts the current state of the device. This message is transmitted by a group 2 only server as an Unconnected Response Message (Message Group 2, Message ID 3). The Slave Node ID(i-7241D) is 0x09.

Note: i-7241D: Node ID=0x09, Master node ID=0x0A

1. Requests the use of the Predefined Master/Slave Connection set

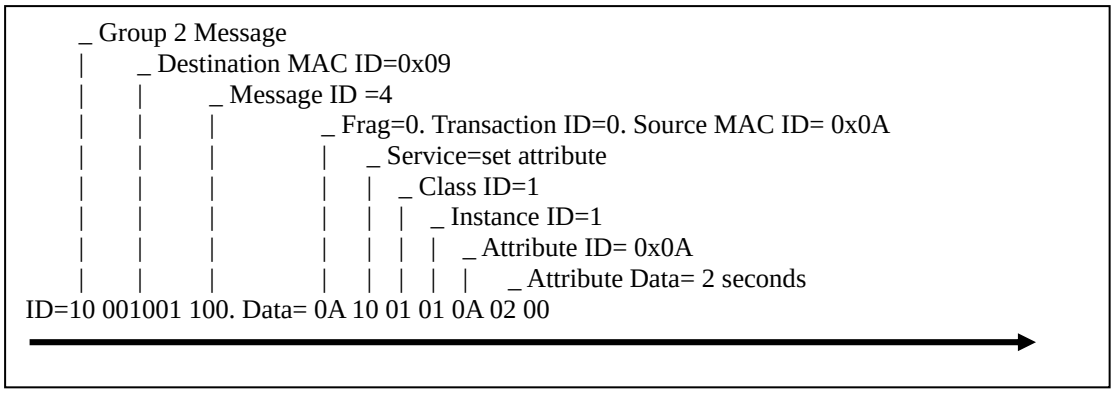
Master (MAC ID =0x0A) Slave (MAC ID =0x09)

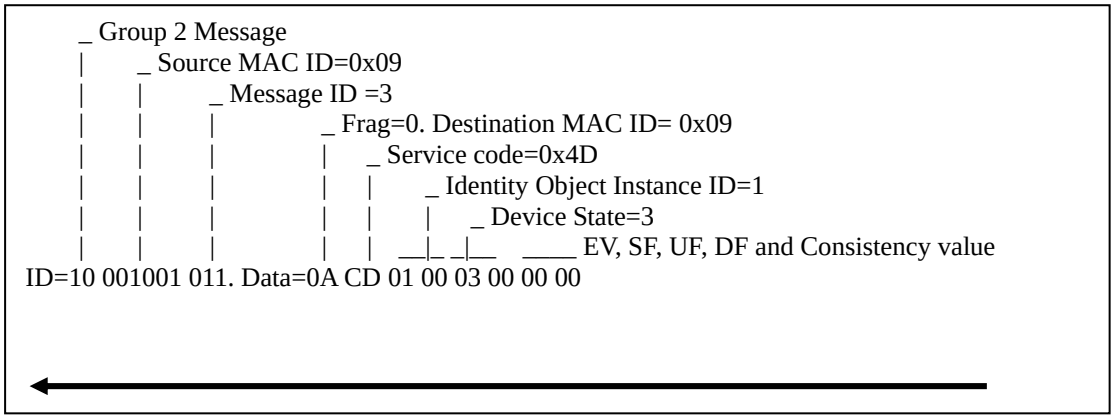




2. Apply the Master's Explicit Request Messages to set the Identify object, service(0x05)=reset.

Master (MAC ID =0x0A) Slave (MAC ID =0x09)





If users want to cancel the heartbeat message, please set the heartbeat interval attribute value of the Identity object instance to zero.

7.2.10 OFFLINE CONNECTION SET

This section describes the Offline Connection Set Messaging Protocol and presents the details associated with the establishment of Offline Connection Set ownership. Support of the Offline Connection Set is optional for all types of devices.

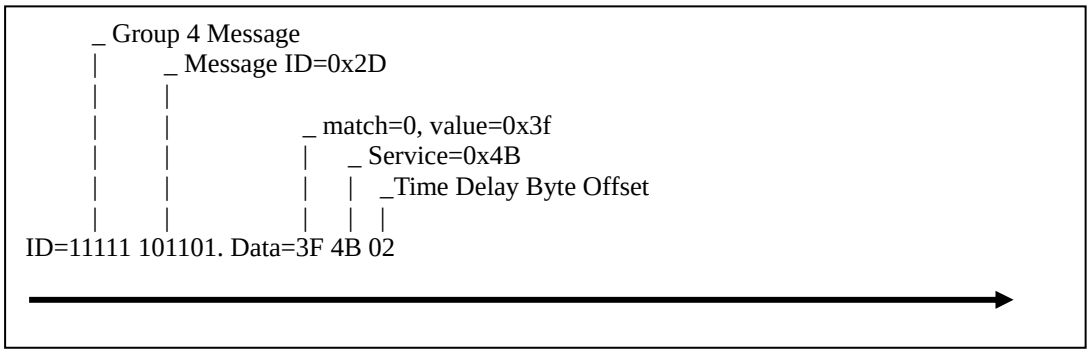
1	1	1	1	1	Group 4 Message ID	Group 4 Messages	000 – 3ff
1	1	1	1	1	2C	Communication Faulted Response Message	
1	1	1	1	1	2D	Communication Faulted Request Message	

Note: i-7241D: Node ID=0x09, Master node ID=0x0a

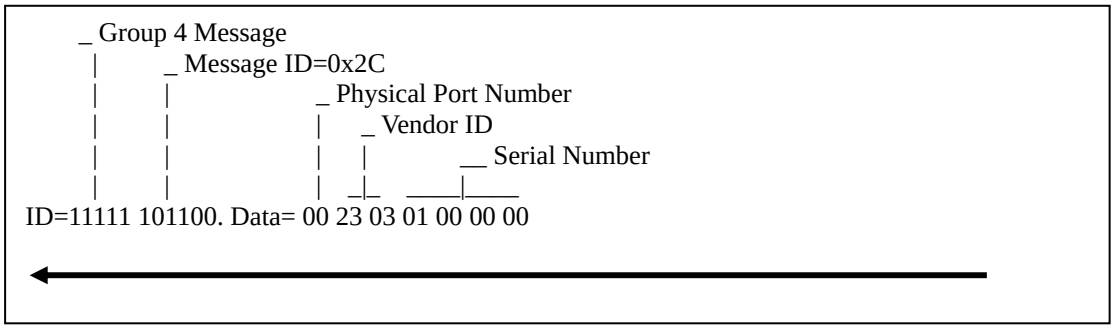
In this example, the i-7241D is set to an off-line state, because it has a duplicated fault. We can then apply the offline connection set to change the i-7241D's baud rate.

1. Apply the Communication Faulted Request message to communicate with offline devices. (Group 4, message 2D , service: 4B)

Master (MAC ID =0x0A)



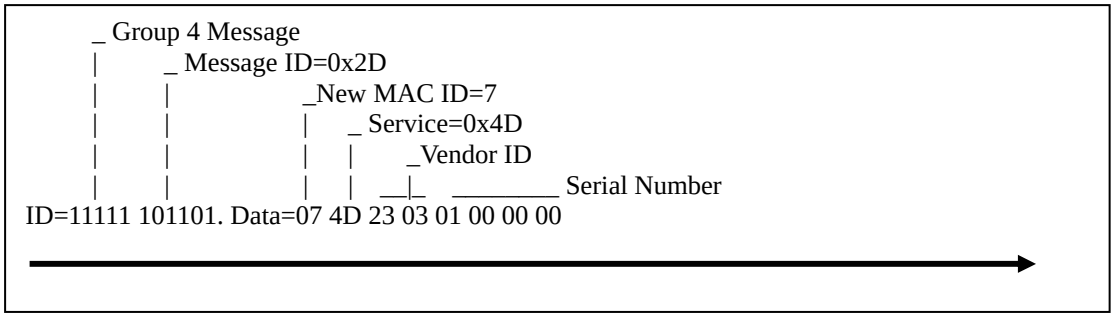
Slave (MAC ID =0x09)



2. Apply the Communication Faulted Request message to change the node ID of i-7241D. (Group 4, message 2D , service: 4D)

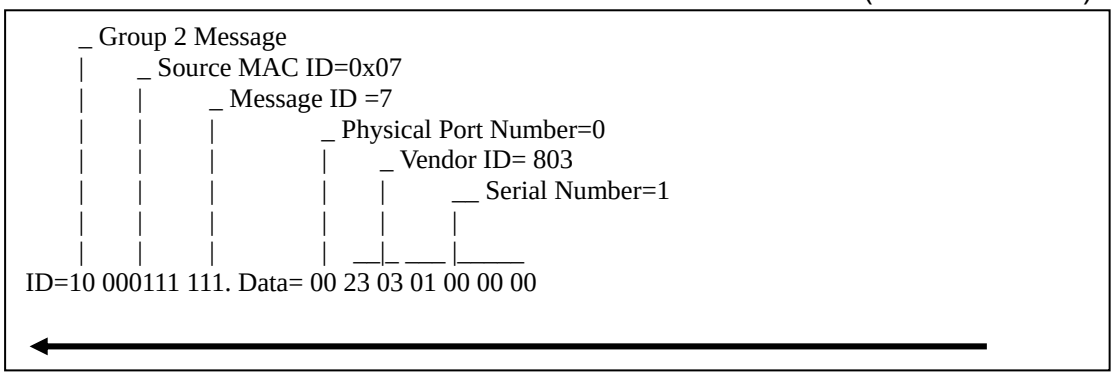
Request message:

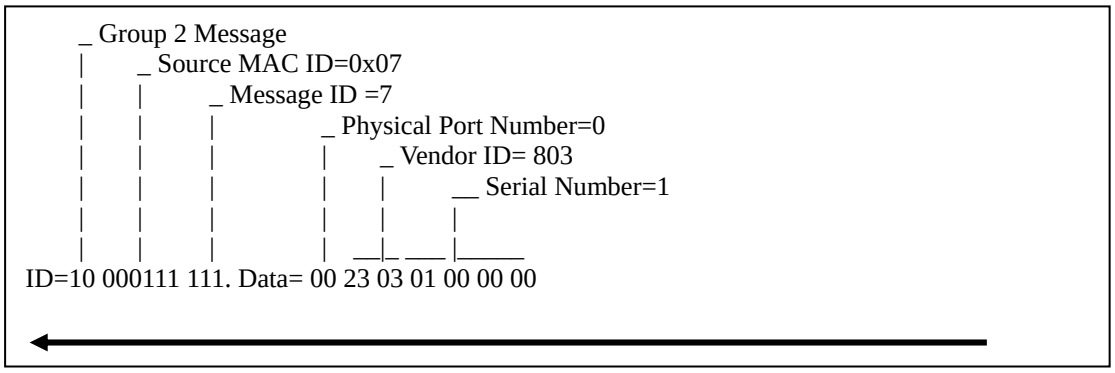
Master (MAC ID =0x0A)



3. When the i-7241D finishes changing the node ID, it will send the duplicated message to the CAN bus.

Slave (MAC ID =0x07)





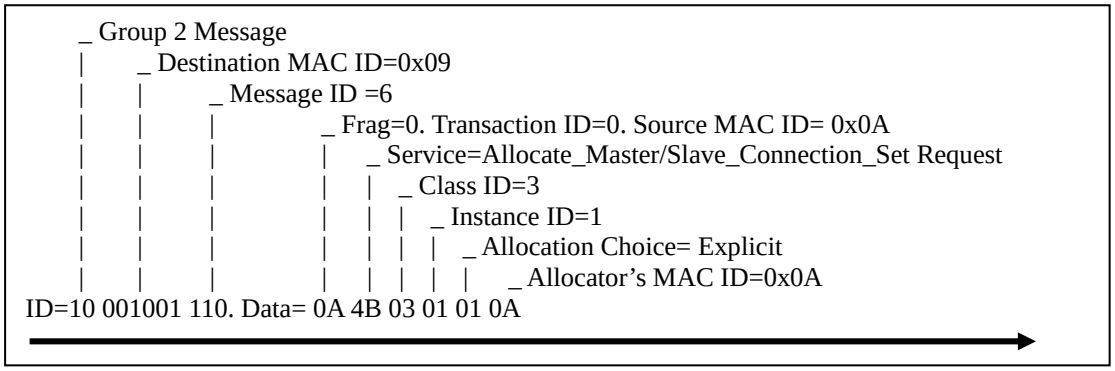
7.2.11 The Watchdog operation of DCON modules

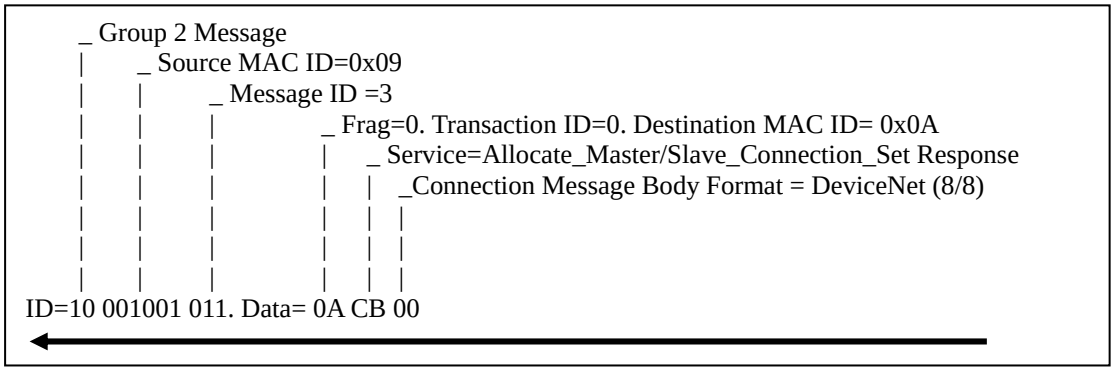
The host watchdog in DCON modules is a software function to monitor the host's operation status. Its purpose is to prevent the network from communication or host halt. When the timeout interval expired, the module will turn all outputs to predefined Safe Value. For detail information, please refer to the DCON manual. In addition, the i-7241D can enable/disable watch dog function of DCON modules via explicit messages. This section will show how to enable/disable the examples.

Note: i-7241D: Node ID=0x09, Master node ID=0x0A

1. Requests the use of the Predefined Master/Slave Connection set

Master (MAC ID =0x0A) Slave (MAC ID =0x09)

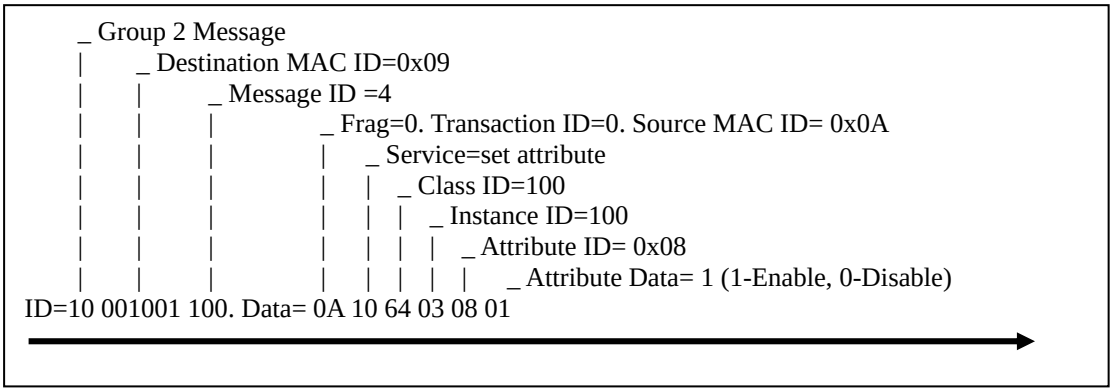


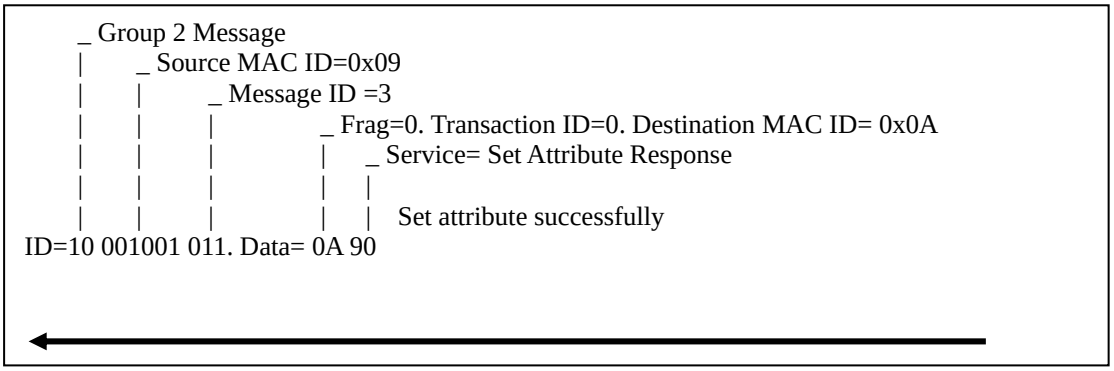


2. Apply the Master's Explicit Request Messages to enable the Watchdog of DCON module.

The instance attribute ID to enable/disable the watchdog of DCON modules is 7. Users can refer to section 3-5 for more information about watchdog function. In addition, the safe value of DCON modules is the current output.

Master (MAC ID =0x0A) Slave (MAC ID =0x09)

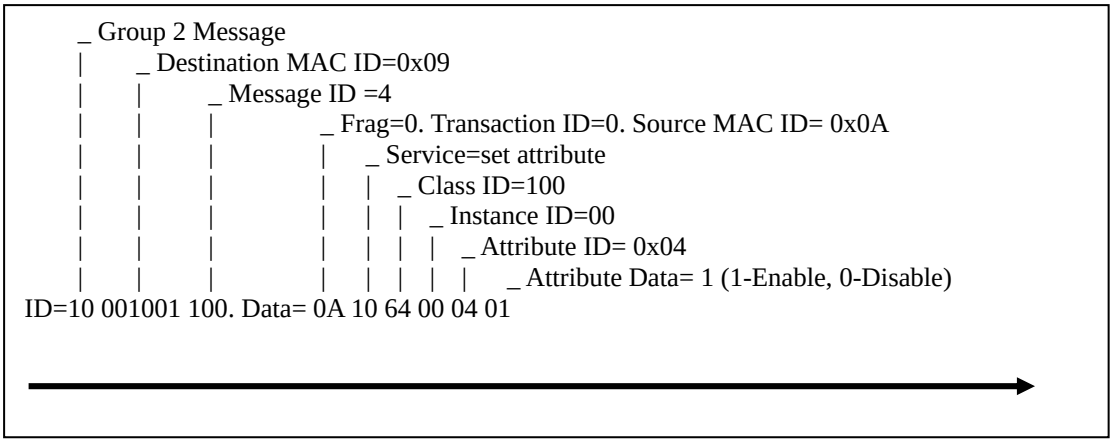


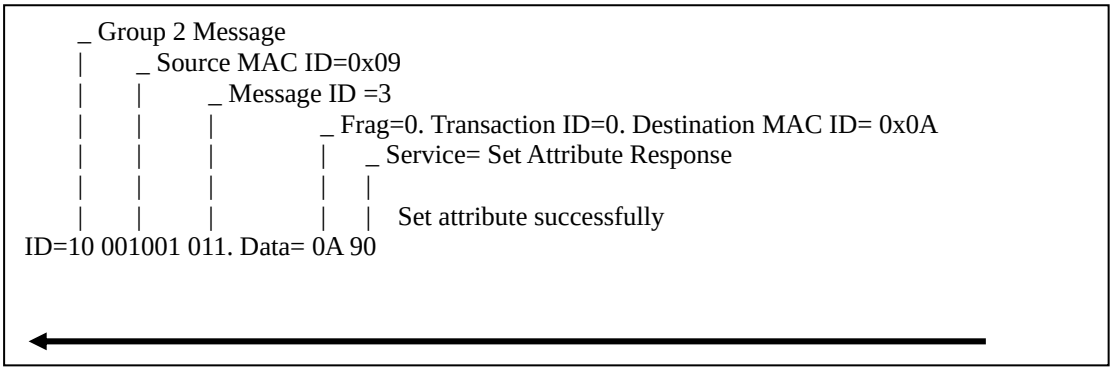


Before watchdog of DCON modules is time out, users can disable the watchdog function of DCON modules via setting the same attribute ID data to 0.

3. Apply the Master's Explicit Request Messages to enable the DNS_DCON gateway to send "Host OK" message to DCON modules.

Master (MAC ID =0x0A) Slave (MAC ID =0x09)

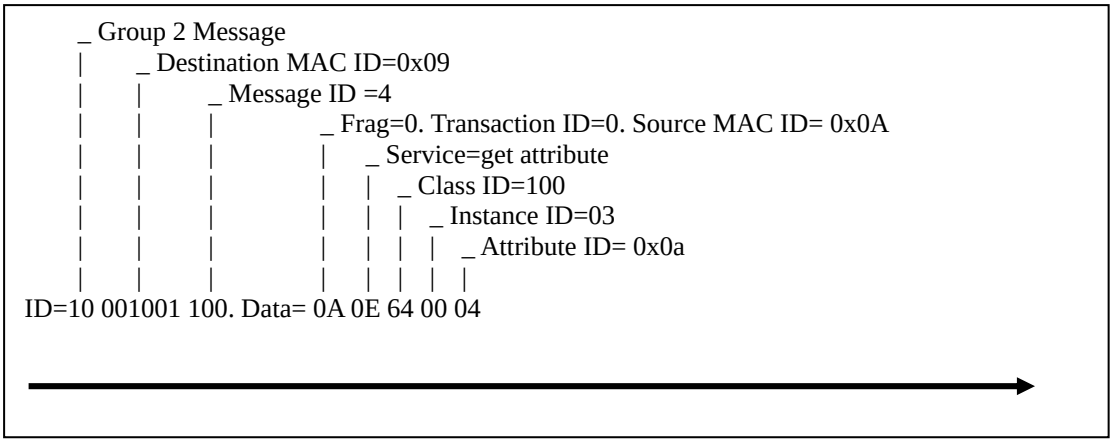


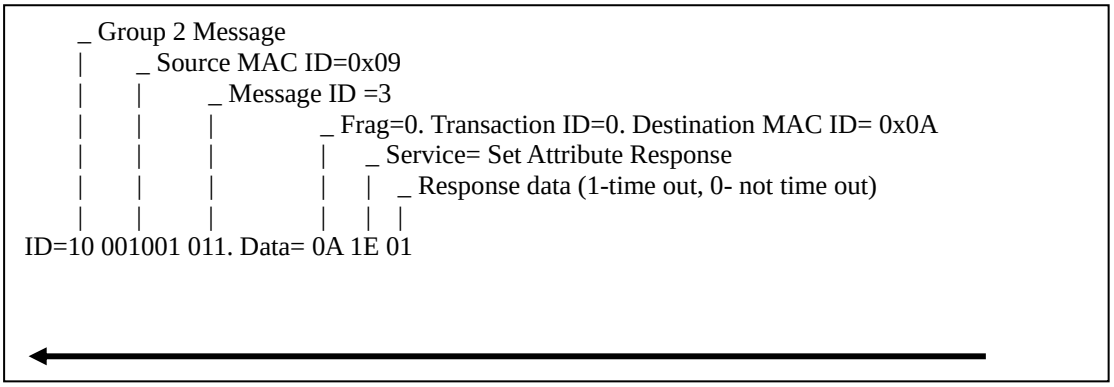


After setting the attribute value as 1, the i-7241D will send "Host OK" message to DCON modules. Also, master can set the value to 0 to disable the i-7241D to send "Host OK" message to DCON modules.

4. Apply the Master's Explicit Request Messages to read the watchdog status of DCON modules.

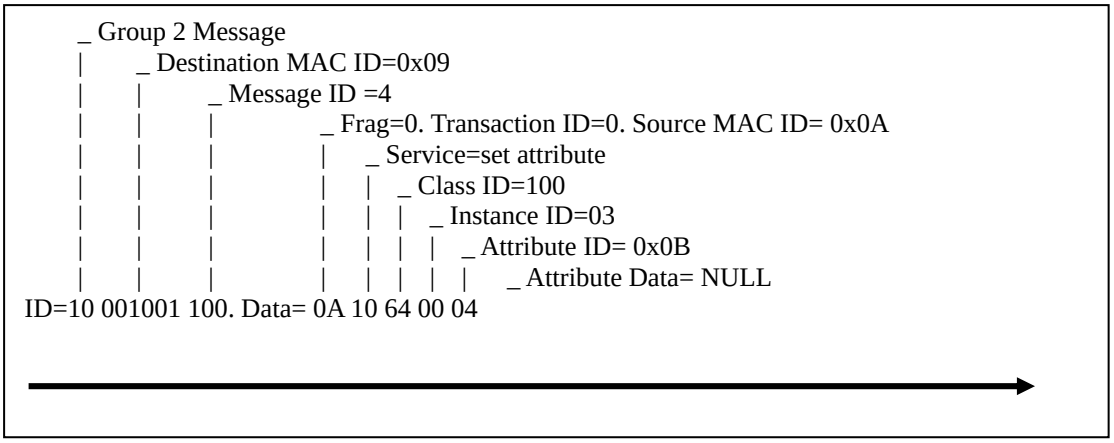
Master (MAC ID =0x0A) Slave (MAC ID =0x09)

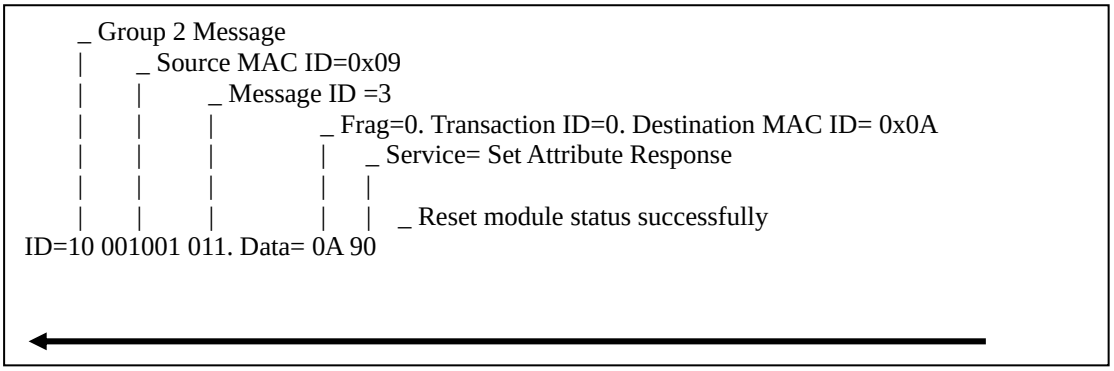




5. Apply the Master's Explicit Request Messages to reset the module status.

Master (MAC ID =0x0A) Slave (MAC ID =0x09)





Before the watchdog is time out, users can not disable the watchdog via this explicit message. But users can disable the watchdog via setting the instance ID (0x08) to 0.

7.2.12 Fragmentation example

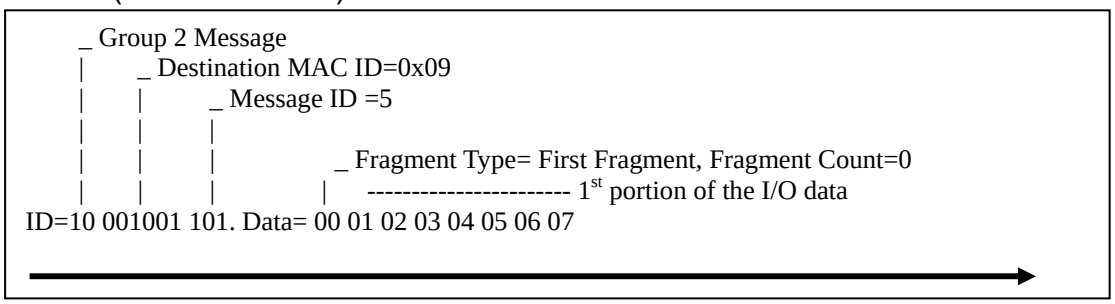
Note: i-7241D: Node ID=0x09, Master node ID=0x0A

Unacknowledged Fragmentation example

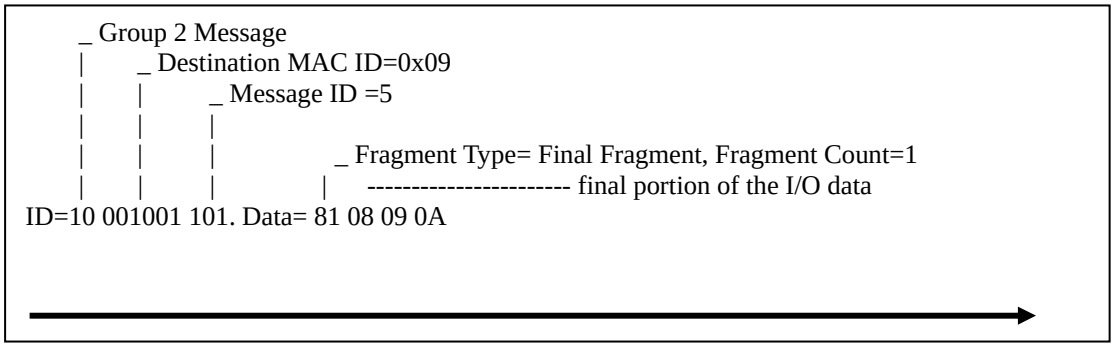
Fragmentation of an I/O message is performed in an Unacknowledged fashion. Unacknowledged fragmentation consists of the back-back transmission of the fragments from the transmitting module. The receiving module(s) returns no acknowledgments (other than the CAN-provided Ack) on a per-fragment basis. The Connection simply invokes the Link Producer's Send service as necessary to move the message without waiting for any specific acknowledgment from the receiving module(s).

In this demo, the polling consumed size is 10 bytes. Master must send fragmented messages. Data=0102030405060708090A. Assume that the I/O Connection has been established.

Master (MAC ID =0x0A)



Master (MAC ID =0x0A)



Acknowledge Fragmentation

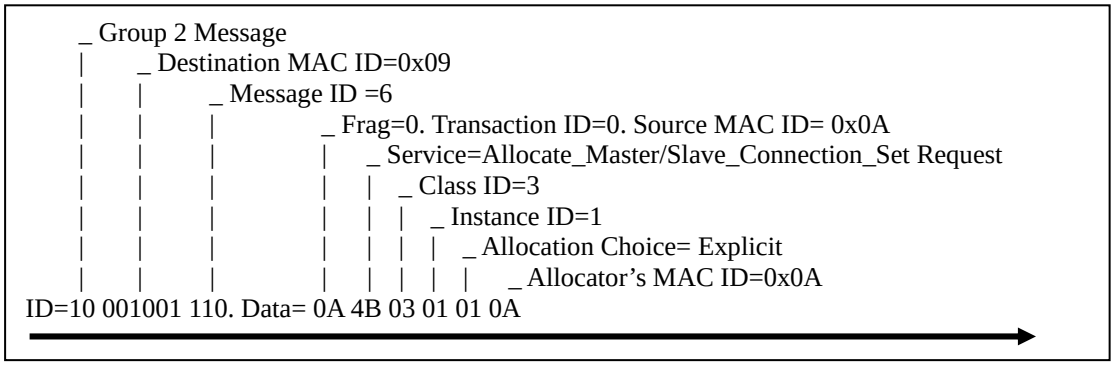
Fragmentation of an Explicit Message is performed in an Acknowledged fashion. Acknowledged fragmentation consists of the transmission of a fragment from the transmitting module followed by the transmission of an acknowledgment by the receiving module. The receiving module acknowledges the reception of each fragment. This provides a degree of flow control. The assumption is that larger bodies of information may be moved across Explicit Messaging Connections (e.g. Upload/Download functions) and, as such, a degree of flow control is necessary.

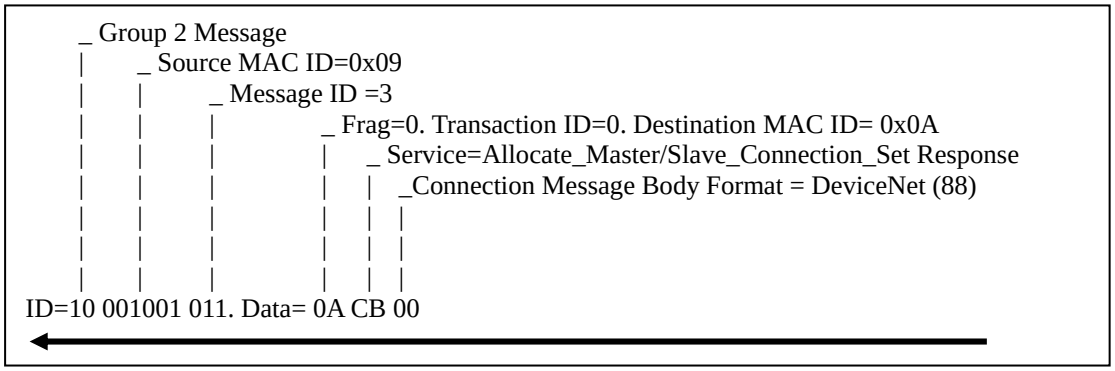
In this demo, assume the attribute data=01020304050607080 90A. The assembly instance ID=4, attribute=3.

Note: i-7241D: Node ID=0x09, Master node ID=0x0A

1. Requests the use of the Predefined Master/Slave Connection set

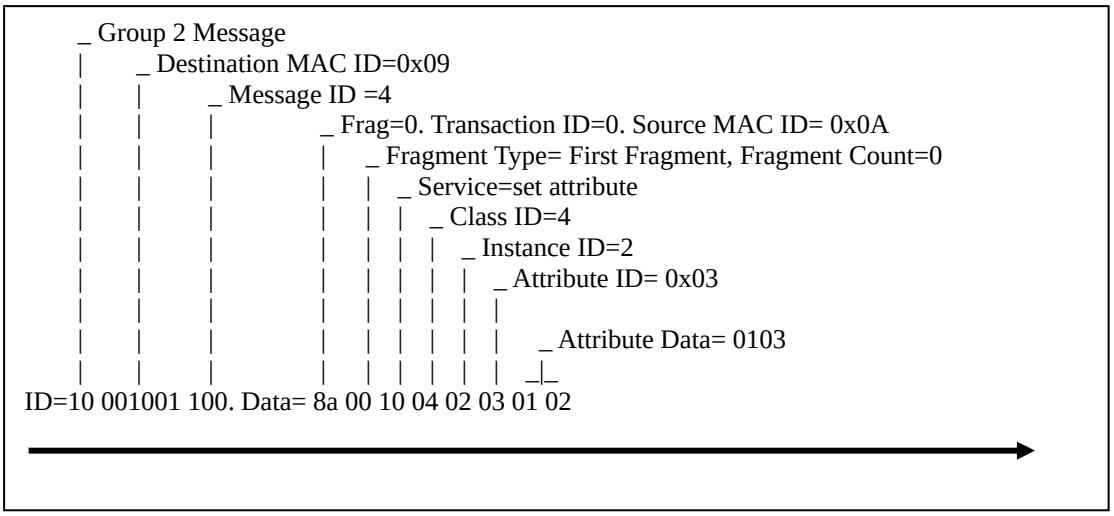
Master (MAC ID =0x0A) Slave (MAC ID =0x09)



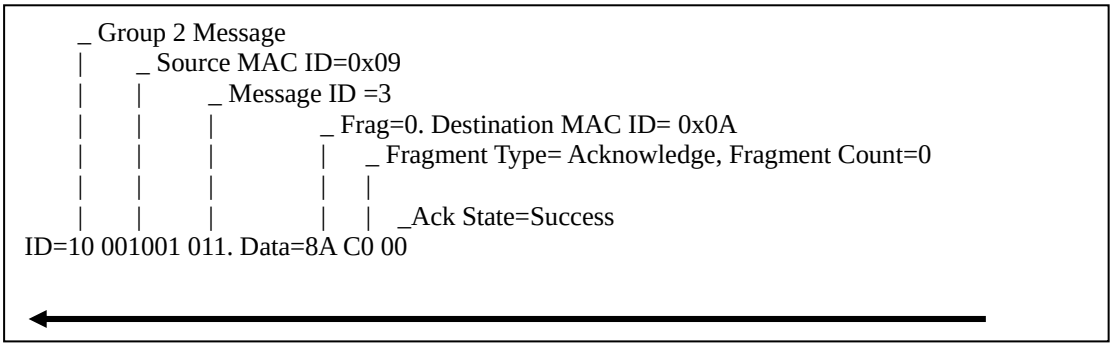


2. Apply the Master's Explicit Request Messages to set the Assembly object, service (0x10)=set attribute service. Data=0102030405060708090A.

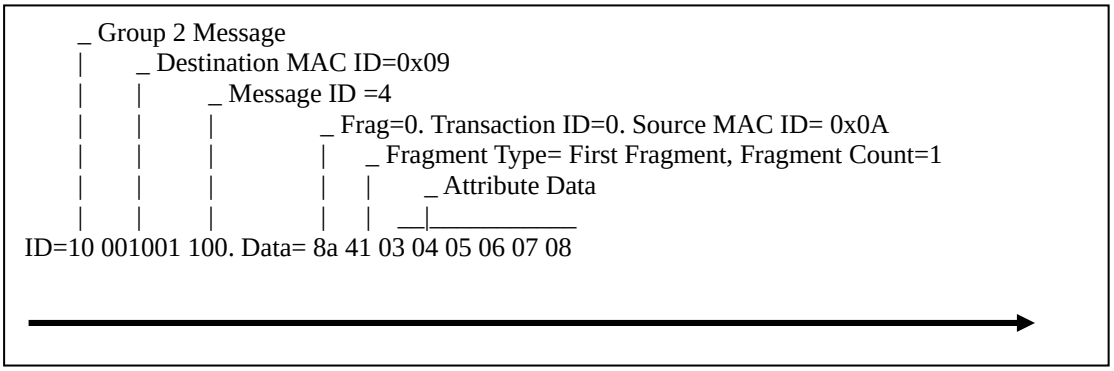
Master (MAC ID =0x0A)



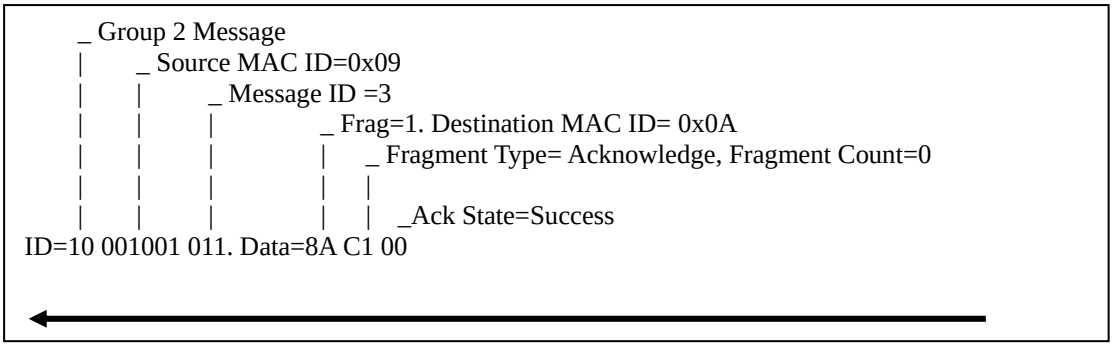
Slave (MAC ID =0x09)



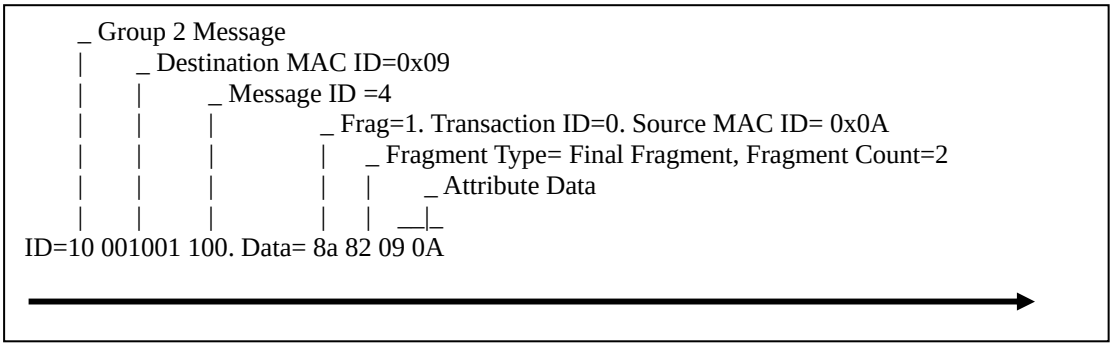
Master (MAC ID =0x0A)



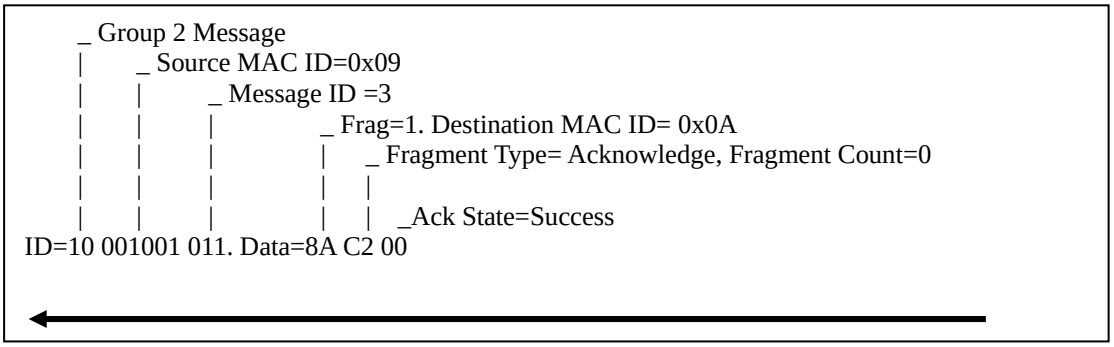
Slave (MAC ID =0x09)



Master (MAC ID =0x0A)



Slave (MAC ID =0x09)



Chapter 8 Interpreting Analog Module Data

8.1 Analog Input Module Data transfer

Because the i-7241D only supports the hex format, all of the AI channels need to transfer to the hex format when storing into this object. The transformation equation is shown below.

$$FloatValue = \left(\frac{HexValue - H\min}{H\max - H\min} \right) * \left(F\max - F\min \right) + F\min$$

The FloatValue is the result after transformation. The HexValue is the value which wants to be transferred. The Hmax and Hmin is the maximum and minimum values of the 2's complement hex range. The Fmax and Fmin is the maximum and minimum value of the float range. User can find out the Hmax, Hmin, Fmax, and Fmin, in the appendix B. For example, The input range of the module I-7017 is set to -10V ~ +10V. According to the table in the appendix B, we can find out the range for hex format is 0x8000 (+32767) ~ 0x7FFF (-32768). Therefore, if the value got from the AI channel of the I-7017 is 0x1234, the AI value with float format can be calculated as follows.

$$\left(\frac{4660 - \left(-32768 \right)}{32767 - \left(-32768 \right)} \right) * \left(10V - \left(-10V \right) \right) + \left(-10V \right) \approx 1.422V$$

By the way, any AI value which is bigger then the maximum value of the input range will be set to the maximum value of the input range automatically. And, the AI value which is small then the minimum value of the input range is also set to the minimum value of the input range automatically.

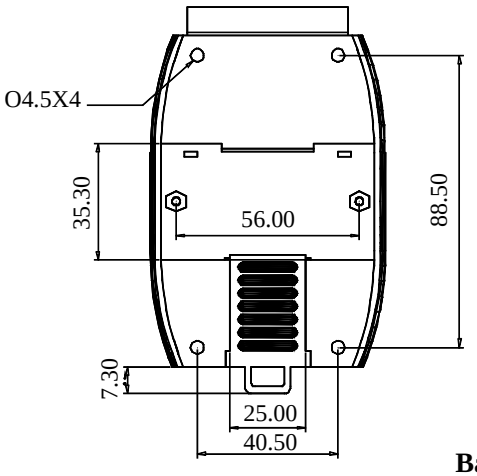
8.2 Analog Output Module Data transfer

Because the i-7241D doesn't support float format, user need to transfer the AO value form float format to hex format. It is similar with the AI situation. The transformation equation is as follows.

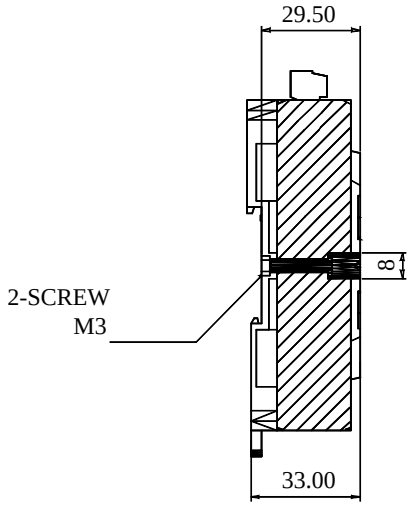
$$HexValue = \left(\frac{FloatValue - F\min}{F\max - F\min} \right) * \left(H\max - H\min \right) + H\min$$

The HexValue is the result after transformation. The FloatValue is the value which wants to be transferred. The Fmax and Fmin is the maximum and minimum values of the float range. The Hmax and Hmin is the maximum and minimum value of the 2's complement hex range. User can find out the Fmax, Fmin, Hmax, and Hmin in the Appendix B.

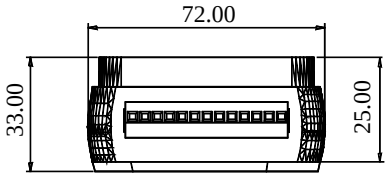
Appendix A: Dimension and Mounting



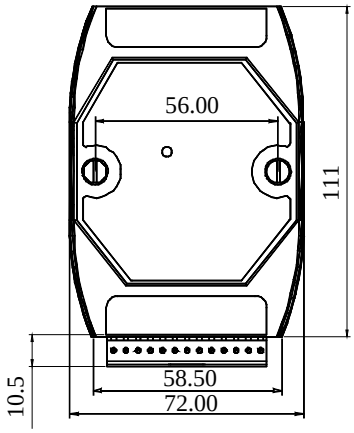
Back View



Side View

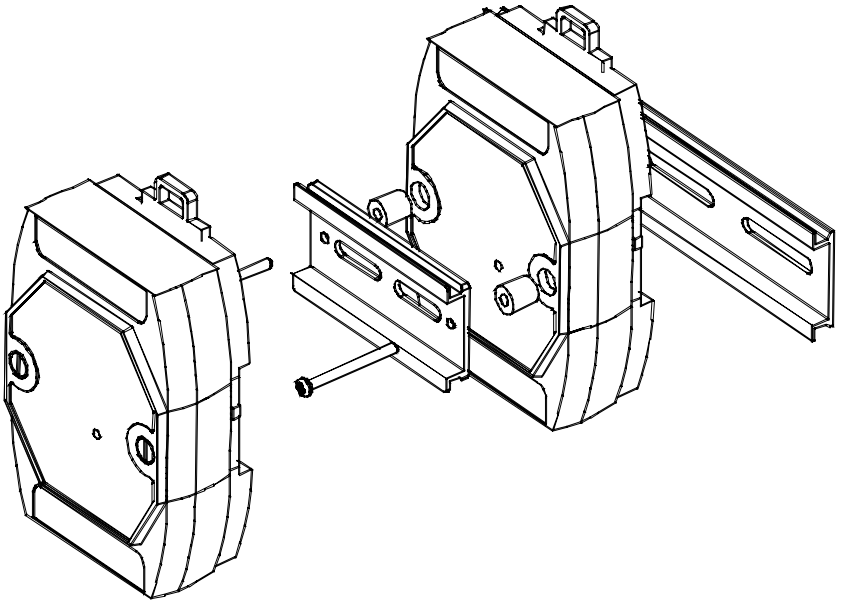


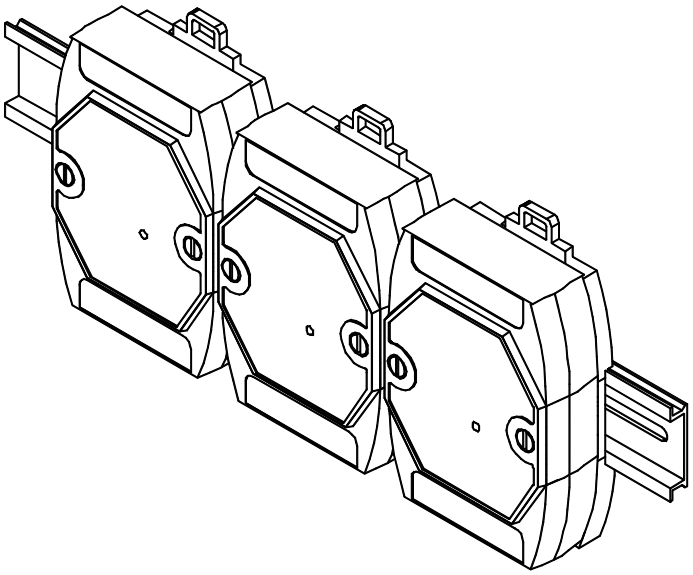
Top View



From View

Unit : mm





Appendix B: Analog I/O Transformation Table

In order to look up the requirement information, we separate the transformation table into several parts according to the DCON module name. And they are shown below.

- I-7005
- I-7012(D), I-7012F(D), I-7014(D), I-7017, I-7017F, I-7017C, I-7017R, I-7017RC, I-87017, I-87017R
- I-87017RC
- I-87017W-A5
- I-7013(D), I-7013(D), I-7015, I-87013
- I-7011(D), I-7011P(D),I-7016(D),7016P(D), I-7018, I-7018P, I-7018R, I-7018BL,I-7018Z, I-7019R, I-87018, I-87018R, I-87018Z
- I-87019R
- I-7021
- I-7022, I-87022
- I-7024, I-87024
- I-87026
- I-7080(D), I-7080B(D), I-87082
- I-7083(D), I-7083B(D)

Note : The i-7241D does not support user-defined types of i-7005.

I-7005

Type Code	Thermistor Type	Data Format	+F.S.	-F.S.
60	PreCon Type III 10K @ 25°C -30 ~ 240°F	Engineering unit	+240.00	-030.00
		% of FSR	+100.00	-012.50
		2's comp HEX	7FFF	F000
		Ohms	+000539.4	+173600.0
61	Fenwell U 2K @ 25°C -50 ~ 150°C	Engineering unit	+150.00	-050.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+000037.2	+134020.0
62	Fenwell U 2K @ 25°C 0 ~ 150°C	Engineering unit	+150.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+000037.2	+006530.0
63	YSI L Mix 100 @ 25°C -80 ~ 100°C	Engineering unit	+100.00	-080.00
		% of FSR	+100.00	-080.00
		2's comp HEX	7FFF	999A
		Ohms	+000014.3	+014470.0
64	YSI L Mix 300 @ 25°C -80 ~ 100°C	Engineering unit	+100.00	-080.00
		% of FSR	+100.00	-080.00
		2's comp HEX	7FFF	999A
		Ohms	+000035.8	+067660.0
65	YSI L Mix 1000 @ 25°C -70 ~ 100°C	Engineering unit	+100.00	-070.00
		% of FSR	+100.00	-070.00
		2's comp HEX	7FFF	A667
		Ohms	+000106.4	+132600.0
66	YSI B Mix 2252 @ 25°C -50 ~ 150°C	Engineering unit	+150.00	-050.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+000041.8	+151000.0
67	YSI B Mix 3000 @ 25°C -40 ~ 150°C	Engineering unit	+150.00	-040.00
		% of FSR	+100.00	-026.67
		2's comp HEX	7FFF	DDDE
		Ohms	+000055.6	+101000.0

Type Code	Thermistor Type	Data Format	+F.S.	-F.S.
68	YSI B Mix 5000 @ 25°C -40 ~ 150°C	Engineering unit	+150.00	-040.00
		% of FSR	+100.00	-026.67
		2's comp HEX	7FFF	DDDE
		Ohms	+000092.7	+168300.0
69	YSI B Mix 6000 @ 25°C -30 ~ 150°C	Engineering unit	+150.00	-030.00
		% of FSR	+100.00	-020.00
		2's comp HEX	7FFF	E667
		Ohms	+000111.5	+106200.0
6A	YSI B Mix 10K @ 25°C -30 ~ 150°C	Engineering unit	+150.00	-030.00
		% of FSR	+100.00	-020.00
		2's comp HEX	7FFF	E667
		Ohms	+000185.9	+177000.0
6B	YSI H Mix 10K @ 25°C -30 ~ 150°C	Engineering unit	+150.00	-030.00
		% of FSR	+100.00	-020.00
		2's comp HEX	7FFF	E667
		Ohms	+000237.0	+135200.0
6C	YSI H Mix 30K @ 25°C -10 ~ 200°C	Engineering unit	+200.00	-010.00
		% of FSR	+100.00	-005.00
		2's comp HEX	7FFF	F99A
		Ohms	+000186.7	+158000.0

Back to Appendix B

Note : The i-7241D does not support user-defined types of i-7005.

I-7012(D), I-7012F(D), I-7014(D), I-7017, I-7017F, I-7017C, I-7017R, I-7017RC, I-87017, I-87017R

Type Code	Input Range	Data Format	+F.S.	Zero	-F.S.
08	-10 to +10 V	Engineer Unit	+10.000	+00.000	-10.000
		% of FSR	+100.00	+000.00	-100.00
		2's complement HEX	7FFF	0000	8000
09	-5 to +5 V	Engineer Unit	+5.0000	+0.0000	-5.0000
		% of FSR	+100.00	+000.00	-100.00
		2's complement HEX	7FFF	0000	8000
0A	-1 to +1 V	Engineer Unit	+1.0000	+0.0000	-1.0000
		% of FSR	+100.00	+000.00	-100.00
		2's complement HEX	7FFF	0000	8000
0B	-500 to +500 mV	Engineer Unit	+500.00	+000.00	-500.00
		% of FSR	+100.00	+000.00	-100.00
		2's complement HEX	7FFF	0000	8000
0C	-150 to +150 mV	Engineer Unit	+150.00	+000.00	-150.00
		% of FSR	+100.00	+000.00	-100.00
		2's complement HEX	7FFF	0000	8000
0D	-20 to +20 mA	Engineer Unit	+20.000	+00.000	-20.000
		% of FSR	+100.00	+000.00	-100.00
		2's complement HEX	7FFF	0000	8000

Back to Appendix B

I-87017RC

Type Code	Input Range	Data Format	Full Scale	Zero	Negative Full Scale
07	-4mA to +20mA	Engineer Unit	+04.000	+00.0000	+20.000
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
0D	-20mA to +20mA	Engineer Unit	+20.000	+00.0000	-20.000
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
1A	+0A to +20mA	Engineer Unit	+20.000	+00.0000	+00.000
		% of FSR	+100.00	+000.00	-000.00
		2's Complement HEX	FFFF	0000	0000

Note:

1. I-87017RC has built-in 125 Ohms resistors for each channel. When connecting to a current source, no add any external resistors required.

I-87017W-A5

Type Code	Input Range	Data Format	Full Scale	Zero	Negative Full Scale
1B	-150V to +150V	Engineer Unit	+150.00	+000.0000	-150.00
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
1C	-50V to +50V	Engineer Unit	+50.000	+00.0000	-50.000
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000

Back to Appendix B

I-7013(D), I-7033(D), I-7015, I-87013

Type Code	RTD Type	Data Format	+F.S.	-F.S.
20	Platinum 100 $\alpha = 0.00385$ -100 ~ 100°C	Engineering unit	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
		Ohms	+138.50	+060.60
21	Platinum 100 $\alpha = 0.00385$ 0 ~ 100°C	Engineering unit	+100.00	+000.00
		% of FSR	+100.00	+100.00
		2's comp HEX	7FFF	0000
		Ohms	+138.50	+100.00
22	Platinum 100 $\alpha = 0.00385$ 0 ~ 200°C	Engineering unit	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+175.84	+100.00
23	Platinum 100 $\alpha = 0.00385$ 0 ~ 600°C	Engineering unit	+600.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+313.59	+100.00
24	Platinum 100 $\alpha = 0.003916$ -100 ~ 100°C	Engineering unit	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
		Ohms	+139.16	+060.60
25	Platinum 100 $\alpha = 0.003916$ 0 ~ 100°C	Engineering unit	+100.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+139.16	+100.00
26	Platinum 100 $\alpha = 0.003916$ 0 ~ 200°C	Engineering unit	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+177.14	+100.00
27	Platinum 100 $\alpha = 0.003916$ 0 ~ 600°C	Engineering unit	+600.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+317.28	+100.00

Type Code	RTD Type	Data Format	+F.S.	-F.S.
28	Nickel 120 -80 ~ 100°C	Engineering unit	+100.00	-080.00
		% of FSR	+100.00	-080.00
		2's comp HEX	7FFF	999A
		Ohms	+200.64	+066.60
29	Nickel 120 0 ~ 100°C	Engineering unit	+100.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+200.64	+120.60
2A[*1]	Platinum 1000 α = 0.00385 -200 ~ 600°C	Engineering unit	+600.00	-200.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+3137.1	+0185.2
2B[*2]	Cu 100 α = 0.00421 -20 ~ 150°C	Engineering unit	+150.00	-020.00
		% of FSR	+100.00	-013.33
		2's comp HEX	7FFF	EEEF
		Ohms	+163.17	+091.56
2C[*2]	Cu 100 α = 0.00427 0 ~ 200°C	Engineering unit	+200.00	+000.00
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
		Ohms	+167.75	+090.34
2D[*2]	Cu 1000 α = 0.00421 -20 ~ 150°C	Engineering unit	+150.00	-020.00
		% of FSR	+100.00	-013.33
		2's comp HEX	7FFF	EEEF
		Ohms	+1631.7	+0915.6
2E[*3]	Platinum 100 α = 0.00385 -200 ~ 200°C	Engineering unit	+200.00	-200.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
		Ohms	+175.84	+018.49
2F[*3]	Platinum 100 α = 0.003916 -200 ~ 200°C	Engineering unit	+200.00	-200.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
		Ohms	+177.14	+017.14

Type Code	RTD Type	Data Format	+F.S.	-F.S.
80*3	Platinum 100 α = 0.00385 -200 ~ 600°C	Engineering unit	+600.00	-200.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+313.59	+018.49
81*3	Platinum 100 α = 0.003916 -200 ~ 600°C	Engineering unit	+600.00	-200.00
		% of FSR	+100.00	-033.33
		2's comp HEX	7FFF	D556
		Ohms	+317.28	+017.14

*1: only available with the I-7013/13D firmware version B1.3 and later, I-7015, I-7033/33D, M-7015 and M-7033/33D.
*2: only available with the I-7015 and M-7015.
*3: only available with the I-7013/13D firmware version B1.3 and later, I-7015 firmware version A1.10 and later, I-7033/33D firmware version B1.3 and later, M-7015 and M-7033/33D.

Over Range	Under Range
7FFFh	8000h

Back to Appendix B

I-7011(D), I-7011P(D), I-7016(D), I-7016P(D), I-7018, I-7018P, I-7018R, I-7018BL,I-7018Z, I-7019R, I-87018, I-87018R, I-87018Z

Type code	Input Type	Data Format	+F.S	-F.S.
00*1	-15 to +15 mV	Engineering unit	+15.000	-15.000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
01*1	-50 to +50 mV	Engineering unit	+50.000	-50.000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
02*1	-100 to +100 mV	Engineering unit	+100.00	-100.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
03*1	-500 to +500 mV	Engineering unit	+500.00	-500.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
04*1	-1 to +1 V	Engineering unit	+1.0000	-1.0000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
05*1	-2.5 to +2.5 V	Engineering unit	+2.5000	-2.5000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
06*1	-20 to +20 mA	Engineering unit	+20.000	-20.000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
07*5	+4 to +20 mA	Engineering unit	+20.000	+04.000
		% of FSR	+100.00	+000.00
		2's comp HEX	FFFF	0000
08*2	-10 to +10 V	Engineering unit	+10.000	-10.000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
09*2	-5 to +5 V	Engineering unit	+5.0000	-5.0000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000

Type code	Input Type	Data Format	+F.S	-F.S.
0A[*2]	-1 to +1 V	Engineering unit	+1.0000	-1.0000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
0B[*2]	-500 to +500 mV	Engineering unit	+500.00	-500.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
0C[*2]	-150 to +150 mV	Engineering unit	+150.00	-150.00
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
0D[*2]	-20 to +20 mA	Engineering unit	+20.000	-20.000
		% of FSR	+100.00	-100.00
		2's comp HEX	7FFF	8000
0E[*1]	Type J Thermocouple -210 ~ 760°C	Engineering unit	+760.00	-210.00
		% of FSR	+100.00	-027.63
		2's comp HEX	7FFF	DCA2
0F[*1]	Type K Thermocouple -270 ~ 1372°C	Engineering unit	+1372.0	-0270.0
		% of FSR	+100.00	-019.68
		2's comp HEX	7FFF	E6D0
10[*1]	Type T Thermocouple -270 ~ 400°C	Engineering unit	+400.00	-270.00
		% of FSR	+100.00	-067.50
		2's comp HEX	7FFF	A99A
11[*1]	Type E Thermocouple -270 ~ 1000°C	Engineering unit	+1000.0	-0270.0
		% of FSR	+100.00	-027.00
		2's comp HEX	7FFF	DD71
12[*1]	Type R Thermocouple 0 ~ 1768°C	Engineering unit	+1768.0	+0000.0
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
13[*1]	Type S Thermocouple 0 ~ 1768°C	Engineering unit	+1768.0	+0000.0
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000

Type code	Input Type	Data Format	+F.S	-F.S.
14*1	Type B Thermocouple 0 ~ 1820°C	Engineering unit	+1820.0	+0000.0
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
15*1	Type N Thermocouple -270 ~ 1300°C	Engineering unit	+1300.0	-0270.0
		% of FSR	+100.00	-020.77
		2's comp HEX	7FFF	E56B
16*1	Type C Thermocouple 0 ~ 2320°C	Engineering unit	+2320.0	+0000.0
		% of FSR	+100.00	+000.00
		2's comp HEX	7FFF	0000
17*3	Type L Thermocouple -200 ~ 800°C	Engineering unit	+800.00	-200.00
		% of FSR	+100.00	-025.00
		2's comp HEX	7FFF	E000
18*3	Type M Thermocouple -200 ~ 100°C	Engineering unit	+100.00	-200.00
		% of FSR	+050.00	-100.00
		2's comp HEX	4000	8000
19*4	Type L DIN43710 Thermocouple -200 ~ 900°C	Engineering unit	+900.00	-200.00
		% of FSR	+100.00	-022.22
		2's comp HEX	7FFF	E38E
1A*5	0 to +20 mA	Engineering unit	+20.000	+00.000
		% of FSR	+100.00	+000.00
		2's comp HEX	FFFF	0000

*1: only available with the I-7018 and I-7019 series

*2: only available with the I-7017 and I-7019 series

*3: only available with the I-7018P, I-7018Z, I-7019 and I-7019R series

*4: only available with the I-7019 series

*5: only available with the I-7017 firmware B2.2 and later, I-7018Z and I-7019R firmware version B2.7 and later.

Back to Appendix B

I-87019R

Type Code	Input Range	Data Format	Full Scale	Zero	Negative Full Scale
00	-15mV to +15mV	Engineer Unit	+15.000	+00.000	-15.000
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
01	-50mV to +50mV	Engineer Unit	+50.000	+00.000	-50.000
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
02	-100mV to +100mV	Engineer Unit	+100.00	+000.00	-100.00
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
03	-500mV to +500mV	Engineer Unit	+500.00	+000.00	-500.00
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
04	-1V to +1V	Engineer Unit	+1.0000	+0.0000	-1.0000
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
05	-2.5V to +2.5V	Engineer Unit	+2.5000	+0.0000	-2.5000
		% of FSR	+100.00	+000.00	-100.00
		2's Complement	7FFF	0000	8000

		HEX			
06	-20mA to +20mA with 125 ohms resistor	Engineer Unit	+20.000	+00.0000	-20.000
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
08	-10V to +10V	Engineer Unit	+10.000	+00.000	-10.000
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
09	-5V to +5V	Engineer Unit	+5.0000	+0.0000	-5.0000
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
0A	-1V to +1V	Engineer Unit	+1.0000	+0.0000	-1.0000
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
0B	-500mV to +500mV	Engineer Unit	+500.00	+000.00	-500.00
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
0C	-150mV to +150mV	Engineer Unit	+150.00	+000.00	-150.00
		% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
0D	-20mA to +20mA	Engineer Unit	+20.000	+00.0000	-20.000

	with 125 ohms resistor	% of FSR	+100.00	+000.00	-100.00
		2's Complement HEX	7FFF	0000	8000
0E	J Type	Engineer Unit	+760.00	+00.000	-210.00
		% of FSR	+100.00	+000.00	-027.63
		2's Complement HEX	7FFF	0000	DCA2
0F	K Type	Engineer Unit	+1372.0	+00.000	-0270.0
		% of FSR	+100.00	+000.00	-019.68
		2's Complement HEX	7FFF	0000	E6D0
10	T Type	Engineer Unit	+400.00	+000.00	-270.00
		% of FSR	+100.00	+000.00	-067.50
		2's Complement HEX	7FFF	0000	A99A
11	E Type	Engineer Unit	+1000.0	+000.00	-0270.0
		% of FSR	+100.00	+000.00	-027.00
		2's Complement HEX	7FFF	0000	DD71
12	R Type	Engineer Unit	+1768.0	+00.000	+0000.0
		% of FSR	+100.00	+000.00	+000.00
		2's Complement HEX	7FFF	0000	0000
13	S Type	Engineer Unit	+1768.0	+00.000	+0000.0
		% of FSR	+100.00	+000.00	+000.00
		2's Complement HEX	7FFF	0000	0000

14	B Type	Engineer Unit	+1820.0	+000.00	+0000.0
		% of FSR	+100.00	+000.00	+000.00
		2's Complement HEX	7FFF	0000	0000
15	N Type	Engineer Unit	+1300.0	+000.00	-0270.0
		% of FSR	+100.00	+000.00	-020.77
		2's Complement HEX	7FFF	0000	E56B
16	C Type	Engineer Unit	+2320.0	+00.000	+0000.0
		% of FSR	+100.00	+000.00	+000.00
		2's Complement HEX	7FFF	0000	0000
17	L Type	Engineer Unit	+800.00	+00.000	-200.00
		% of FSR	+100.00	+000.00	-025.00
		2's Complement HEX	7FFF	0000	E000
18	M Type	Engineer Unit	+100.00	+000.00	-200.00
		% of FSR	+050.00	+000.00	-100.00
		2's Complement HEX	4000	0000	8000
19	L Type DIN43710	Engineer Unit	+900.00	+000.00	-200.00
		% of FSR	+100.00	+000.00	-022.22
		2's Complement HEX	7FFF	0000	E38F

Back to Appendix B

I-7021

Type Code	Output Range	Data Format	Max.	Min.
30	0 to 20 mA	Engineer Unit	20.000	00.000
		% of Span	+100.00	+000.00
		Hexadecimal	FFF	0000
31	4 to 20 mA	Engineer Unit	20.000	04.000
		% of Span	+100.00	+000.00
		Hexadecimal	FFF	0000
32	0 to 10 V	Engineer Unit	10.000	00.000
		% of Span	+100.00	+000.00
		Hexadecimal	FFF	0000

Back to Appendix B

I-7022, I-87022

Output Type	Output Range	Data Format	Max.	Min.
0	0 to 20 mA	Engineer Unit	20.000	00.000
		% of Span	+100.00	+000.00
		Hexadecimal	FFF	0000
1	4 to 20 mA	Engineer Unit	20.000	04.000
		% of Span	+100.00	+000.00
		Hexadecimal	FFF	0000
2	0 to 10 V	Engineer Unit	10.000	00.000
		% of Span	+100.00	+000.00
		Hexadecimal	FFF	0000

Back to Appendix B

I-7024, I-87024

Type Code	Output Range	Data Format	Max Value	Min Value
30	0 to 20mA	Engineer Unit	20.000	00.000
		% of FSR	+100.00	+000.00
		Hexadecimal	7FFF	0000
31	4 to 20mA	Engineer Unit	20.000	04.000
		% of FSR	+100.00	+000.00
		Hexadecimal	7FFF	0000
32	0 to 10V	Engineer Unit	10.000	00.000
		% of FSR	+100.00	+000.00
		Hexadecimal	7FFF	0000
33	-10 to +10V	Engineer Unit	+10.000	-10.000
		% of FSR	+100.00	+000.00
		Hexadecimal	7FFF	8000
34	0 to 5V	Engineer Unit	5.000	00.000
		% of FSR	+100.00	+000.00
		Hexadecimal	7FFF	0000
35	-5 to +5V	Engineer Unit	+5.000	-5.000
		% of FSR	+100.00	+000.00
		Hexadecimal	7FFF	8000

Back to Appendix B

I-87026

Type Code	Output Range	Data Format	Max Value	Min Value
0	0 to 20mA	Engineer Unit	20.000	00.000
		% of FSR	+100.00	+000.00
		Hexadecimal	FFFF	0000
1	4 to 20mA	Engineer Unit	20.000	04.000
		% of FSR	+100.00	+000.00
		Hexadecimal	FFFF	0000
2	0 to 10V	Engineer Unit	10.000	00.000
		% of FSR	+100.00	+000.00
		Hexadecimal	FFFF	0000

Back to Appendix B

I-7080, I-7080B, I-87082

Type Code	Description
50	Counter Mode
51	Frequency Measurement

Back to Appendix B

I-7083

Type Code	Description
53	Encoder

Back to Appendix B