DeviceNet Slave Device

CAN-2012D

User's Manual

Warranty

Without contrived damage, all products manufactured by ICP DAS are warranted in one year from the date of delivery to customers.

Warning

ICP DAS revises the manual at any time without notice. However, no responsibility is taken by ICP DAS unless infringement act imperils to patents of the third parties.

Copyright

Trademark

The brand name ICP DAS as a trademark is registered, and can be used by other authorized companies.

Contents

1 Introduction

1.1 Overview

DeviceNet is a network protocol based on the CAN bus, mainly used in embedded network systems, such as industrial computer control, vehicle control systems, factory automation, medical equipment control, remote data acquisition, environmental monitoring, and packaging machine control, etc.

The CAN-2012D complies with the DeviceNet specification (Volume I/II, Release 2.0). Users can obtain analog input data and module configuration through the standard DeviceNet protocol. Establishing communication between the DeviceNet master and the CAN-2012D is no longer a difficult task. The DeviceNet Electronic Data Sheet (EDS) can quickly and easily assist users in achieving this purpose. The CAN-2012D module provides 8 analog input channels in differential mode and 16 analog input channels in single-ended mode, which can be used to measure voltage and current, making it suitable for various applications. By using ICP DAS DeviceNet masters, users can easily build a DeviceNet network to meet their requirements.

1.2 Hardware Specifications

Analog Input:

- Input Channels: 8 (Differential) / 16 (Single-Ended)
- Input Range: ±10V, ±5V, ±2.5V, 0 ~ +10V, ±20mA, 0 ~ +20mA, +4mA ~ +20mA (No external 125 Ω resistor required)
- Resolution: 16-bit
- Accuracy: ±0.1% of Full Scale
- Sampling Rate: 4 k samples/sec (Total Channels)
- Zero Drift: ±0.2μV/°C
- Span Drift: ±5 ppm/°C
- Common Mode Rejection: 84 dB
- Input Impedance: ≥ 1 MΩ
- Overvoltage Protection: 70 Vrms
- Individual Channel Configuration: Yes
- Isolation: 3000 VDC
- ESD Protection: ±4 kV (Contact)

Others:

- Power LED: PWR (red)
- DeviceNet Status LED: NET/MOD (green/orange)
- Upper/Lower Limit/Delta Warning LED Indicators
- LED Indicator for Latchable Alarms
- IEEE-754 Floating Point Data Format
- Power Supply: +10 ~ +30 VDC.
- Power Consumption: 0.5 W.
- Operating Temperature: -25 ~ +75 °C.
- Storage Temperature: -30 ~ +80 °C.
- Humidity: Relative Humidity 10 to 90%, Non-condensing
- Dimensions: 33 mm x 99 mm x 78 mm (W x L x H) .
- DIP Switch Setting for 120 Ω Terminal Resistor

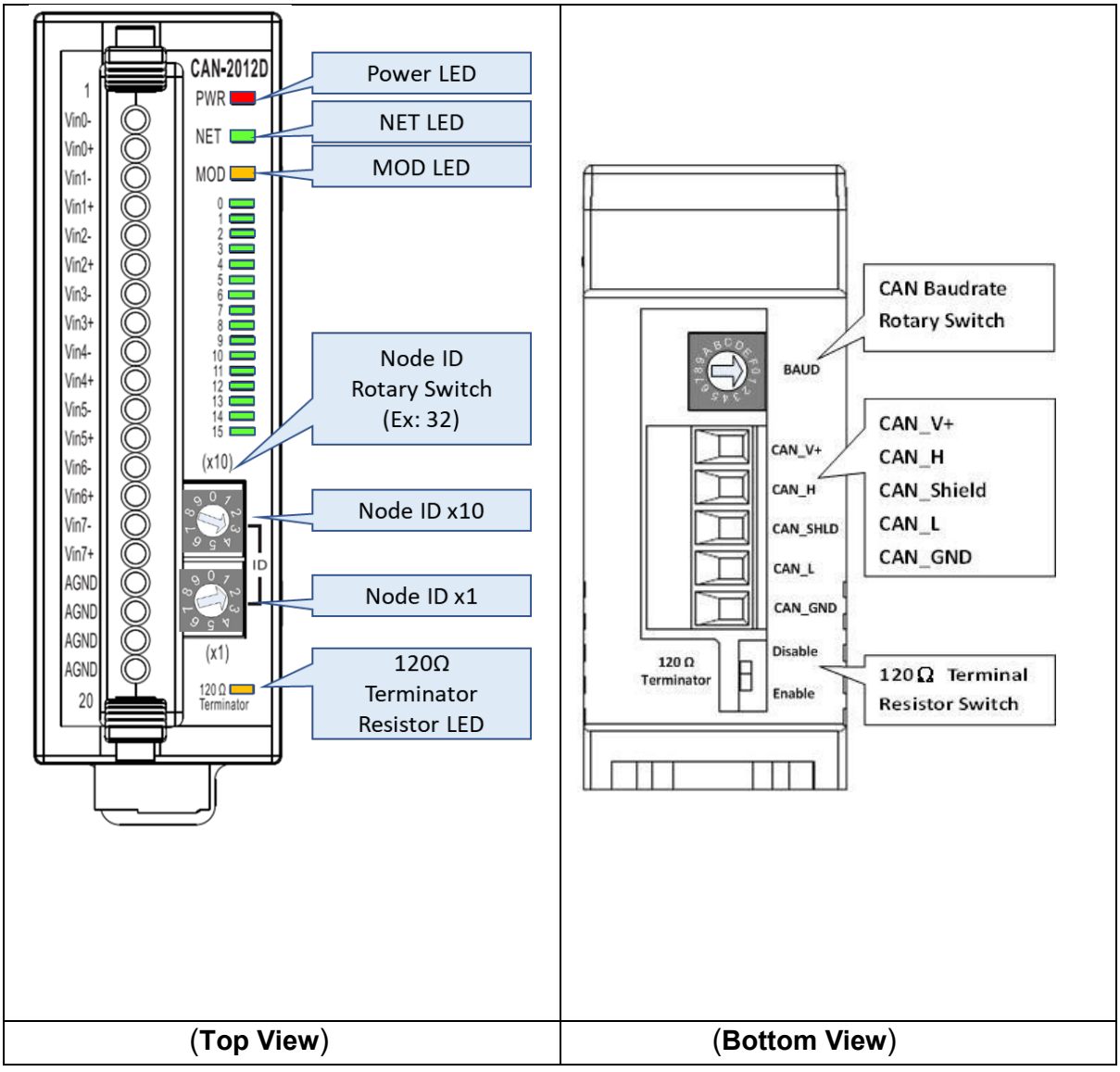
1.3 Features

- DeviceNet Generic I/O Slave Device
- Compliant with DeviceNet Specification (Volume I, Release 2.0 & Volume II, Release 2.0, Errata 5)
- Supports Group 2 Only Server Connection (non-UCMM)
- Supports Predefined Master/Slave Connection
- One Explicit Messaging Connection
- One Polled I/O Connection
- Supports DeviceNet Heartbeat and Shutdown Messages
- Provides Electronic Data Sheet (EDS) for Standard DeviceNet Master
- NET, MOD, and PWR LED Indicators
- Baud Rate: 125k, 250k, 500k (set via Rotary Switch)

1.4 Application

2 Hardware

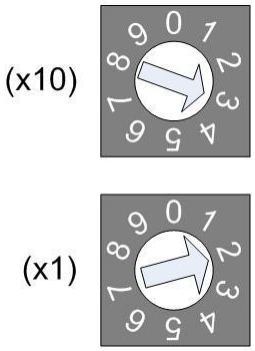
2.1 Structure



(Top View)	(Bottom View)

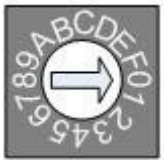
2.2 Node ID & Baud Rate Rotary Switch

The rotary switches of node ID configure the node ID of CAN-2012D module. These two switches are for the tens digit and the units digit of the node ID. The node ID value of this demo picture is 32.



Node ID Rotary Switch

The rotary switch for baud rate handles the CAN baud rate of CAN-2012D module. The relationship between the rotary switch value and the practical baud rate is presented in the following table.



Baud rate rotary switch

Rotary Switch Value	Baud rate (kbps)
0	125
1	250
2	500

2.3 LED Description

POWER LED

The CAN-2012D requires a power supply of 10 ~ 30 VDC. When a proper power source and voltage are correctly connected to the module, the power indicator will turn on and display red.

NET LED

The NET LED indicates the current status of the DeviceNet communication link.

condition	status	indicates
Init Off	Off line	Device is not online
Off	Connection timeout	I/O connection timeout
Flashing	On line	Device is on line, but not communicating
Init solid	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	On line, communicating	Device is online and communicating

MOD LED

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

condition	status	indicates
Off	Normal	
Solid	Critical fault	Device has unrecoverable fault.
Flashing	Non_critical fault	Device has recoverable fault to recover. If users want to fix the problem, reconfiguring device's MAC ID or resetting device may work.

Terminal Resistor Indicator

When the 120 Ω terminal resistor is enabled, the terminal resistor indicator will be illuminated.

Upper/Lower Limit and Delta Range Alarm Indicator

When alarm triggering is enabled, the trigger type and upper/lower limit values

or Delta range are set, and the analog voltage/current input exceeds the defined upper/lower limits or Delta range, the alarm LED indicator will be activated.

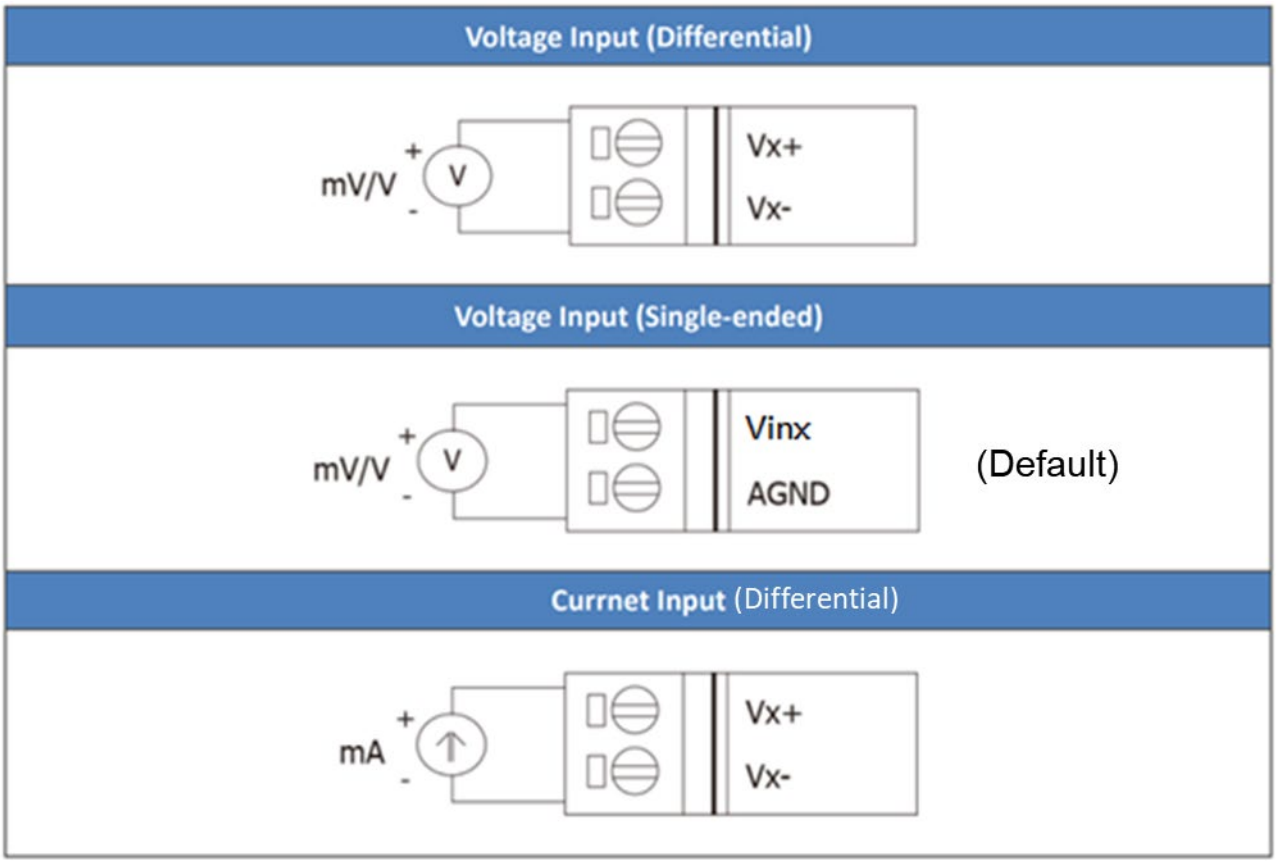
2.4 PIN Assignment

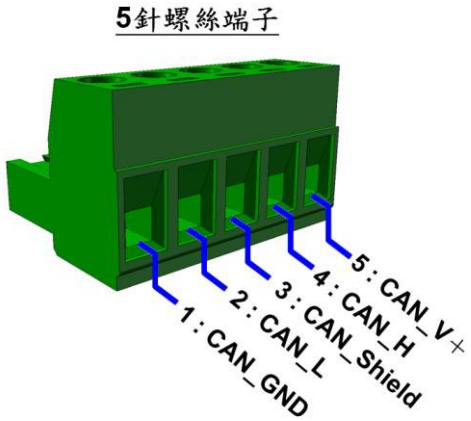
-	Differential	Single-ended
Terminal No.	Pin Assignment	Pin Assignment
1	V0-	Vin0
2	V0+	Vin1
3	V1-	Vin2
4	V1+	Vin3
5	V2-	Vin4
6	V2+	Vin5
7	V3-	Vin6
8	V3+	Vin7
9	V4-	Vin8
10	V4+	Vin9
11	V5-	Vin10
12	V5+	Vin11
13	V6-	Vin12
14	V6+	Vin13
15	V7-	Vin14
16	V7+	Vin15
17	AGND	AGND
18	AGND	AGND
19	AGND	AGND
20	AGND	AGND

2.5 Wire Connection

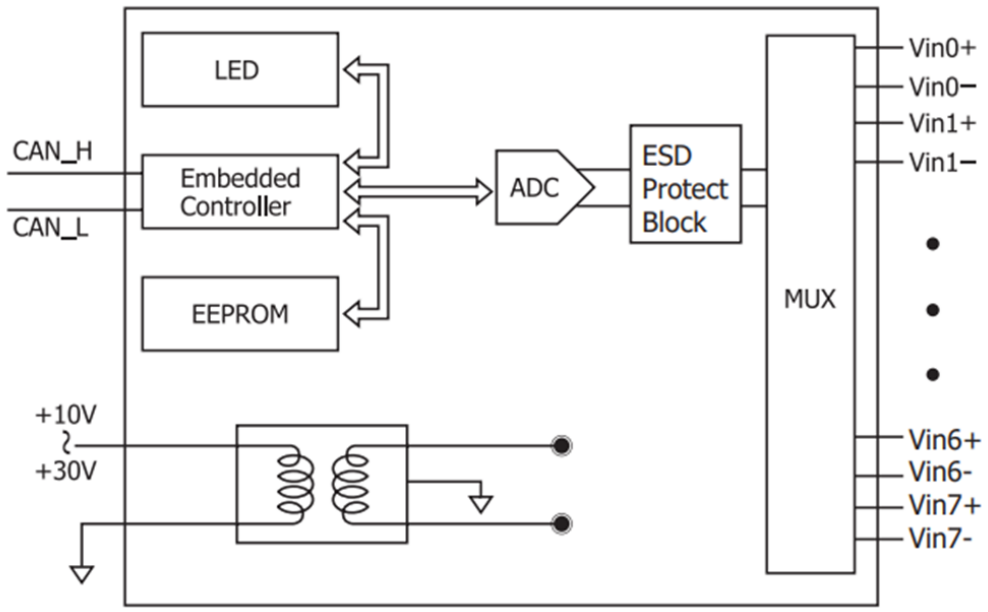
The CAN-2012D input type is set to Single-ended Input by default. To change the input type, please refer to the DeviceNet object (Class:0x64, Instance: x, Attribute:0x9).

Notice: x = channel num





2.6 Internal I/O Block Diagram



3 DeviceNet Profile Area

This section describes the detailed functions for each object class that is implemented in the CAN-2012D DeviceNet network.

3.1 DeviceNet Statement of Compliance

General Device Data

Device Information	Description
Version Description of DeviceNet Specification	Volume I, Release 2.0 & Volume II, Release 2.0
Vendor Name	ICP DAS
Device Profile Name	CAN-2012D
Production Revision	1.1

DeviceNet Physical Conformance Data

Item	Description
LED Support	Yes
MAC ID Setting	Switch (0 ~ 63)
Default MAC ID	1
Communication Baud Rate Setting	Switch (125, 250, 500 kbps)
Default Baud Rate	125 kbps

3.2 Identity Object (Class ID: 0x01)

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

Class Attribute (Instance ID=0)

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 (Instance ID=1)

Attribute ID	Description	Method	DeviceNet Data Type	Value
0x01	Vendor	Get	UINT	803
0x02	Product type	Get	UINT	0x00
0x03	Product code	Get	UINT	0x302
0x04	Major. Minor of firmware version	Get	Struct of USINT USINT	1.1
0x05	Status	Get	WORD	-
0x06	Serial number	Get	UDINT	1
0x07	Product name	Get	Short_String	CAN-2012D
0x0A	Heartbeat Interval	Get/Set	USINT	0(default)

Instance Service

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

Note: Use the Instance Service 0x05 will reboot the device.

3.3 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.

The default IO connection path is as follow.

Connection Path	Class ID	Instance ID	Attribute ID
Poll Produced	0x04	0x64	0x03
Poll Consumed	0x65	0x01	0x01

Connection Instance ID	Description
0x01	References the Explicit Messaging Connection into the Server
0x02	References the Poll I/O Connection

3.4 Assembly Object (Class ID: 0x04)

The Assembly Object binds attributes of multiple objects, which allows data 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 of ”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 (Instance ID=0)

Attribute ID	Attribute name	Data Type	Method	Value
0x01	Revision	UINT	Get	0001
0x02	Max Instance	UINT	Get	0x08

Class Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes

Instance ID

Instance ID	OUTPUT	INPUT
0x64		Get AI Value
0x65		Get Alarm Status

Contents of Each Assembly Object Instance

Instance ID	Description	Type	Method	Default Value
0x64	Channel 0 ~ 15 AI Value	IEEE-754 Float	Get	0x00
0x65	Channel 0 ~ 15 Alarm Status	BYTE	Get	0x00

Parameter description of Assembly Object Instance

Instance ID	Data Range	Parameter Description
0x64	Refer to Appendix B	Channel 0 ~ 15 AI Value
0x65	Trigger Status: 0: Alarm not triggered. 1: Alarm triggered. Bit No. / Alarm Status Bit Definition 0 / Alarm when exceeding upper limit	Channel 0~15 Analog Input Alarm Trigger Status. For the status indication, please refer to the list on the left.

	1	Alarm when below lower limit	
	2	Alarm when exceeding Delta range	
	3~7	Reserved	

Instance Attribute (**Instance** ID=0x64~0x65)

Attribute ID	Description	Method	DeviceNet Data Type	Value
0x03	Data	Get	INPUT	Dependent on instance ID

Instance Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes

3.5 Application Object1 (Class ID: 0x64)

Application objects are the interfaces between an application and the DeviceNet Layer. The attributes of application Objects contain the data for the application, which are accessed and exchanged via DeviceNet. DeviceNet accesses application data by invoking read and write functions. These functions need to be provided by an Application Object. DeviceNet provides Get_Attribute_Single and Set_ Attribute_Single to read and write CAN-2012D module.

Application Object 1 defines the configuration of analog input channels:

Class Attribute (Instance ID=0)

Attribute ID	Attribute name	Data Type	Method	Value
0x01	Revision	UINT	Get	0001
0x02	Max Instance	UINT	Get	0x10

Class Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes

Instance ID

Instance ID	Description
0x01	Accessing Analog Input Configuration of Channel 0
0x02	Accessing Analog Input Configuration of Channel 1
0x03	Accessing Analog Input Configuration of Channel 2
0x04	Accessing Analog Input Configuration of Channel 3
0x05	Accessing Analog Input Configuration of Channel 4
0x06	Accessing Analog Input Configuration of Channel 5
0x07	Accessing Analog Input Configuration of Channel 6
0x08	Accessing Analog Input Configuration of Channel 7
0x09	Accessing Analog Input Configuration of Channel 8
0x0A	Accessing Analog Input Configuration of Channel 9
0x0B	Accessing Analog Input Configuration of Channel 10
0x0C	Accessing Analog Input Configuration of Channel 11
0x0D	Accessing Analog Input Configuration of Channel 12
0x0E	Accessing Analog Input Configuration of Channel 13
0x0F	Accessing Analog Input Configuration of Channel 14
0x10	Accessing Analog Input Configuration of Channel 15

Instance Attribute (Attribute ID=0x01~0x0D)

Attribute ID	Description	Method	Data Type	Default Value
0x01	Channel 0~15 AI Values	Float	Get	0x0000
0x02	Channel 0~15 AI Type	unsigned char	Get/Set	0x01
0x03	Channel 0~15 AI Alarm Enable Status	unsigned char	Get/Set	0x00
0x04	Channel 0~15 AI Alarm Type	unsigned char	Get/Set	0x00
0x05	Channel 0~15 AI Alarm Upper Limit Value	float	Get/Set	0x7F7FFFF
0x06	Channel 0~15 AI Alarm Lower Limit Value	float	Get/Set	0xFF7FFFF
0x07	Channel 0~15 AI Alarm Delta Value	float	Get/Set	0x00
0x08	Channel 0~15 AI Alarm Status	unsigned char	Get	0x00
0x09	Channel 0~15 AI Channel Mode	unsigned char	Get/Set	0x01
0x0A	Channel 0~15 AI Offset	float	Get/Set	0x00
0x0B	Channel 0~15 AI Latched Alarm Status	unsigned char	Get/Set	0x00
0x0C	Channel 0~15 AI Clear Latched Alarm	unsigned char	Set	-
0x0D	Channel 0~15 AI Channel Enable Status	unsigned char	Get/Set	0x01

Parameter description of Application Object1 attributes

<table>
<tr><th>Attribute ID</th><th>Data Range</th><th>Parameter Description</th></tr>
<tr><td>0x01</td><td>Refer to Appendix B for corresponding input values</td><td>Channel 0~15 Analog Input Values</td></tr>
<tr><td>0x02</td><td>Refer to Appendix B for corresponding input values</td><td>Channel 0~15 Analog Input Type</td></tr>
<tr><td>0x03</td><td>0x00: Disable Alarm Trigger.
0x01: Enable Alarm Trigger.</td><td>Channel 0~15 Analog Input Alarm Trigger Enable Status</td></tr>
<tr><td>0x04</td><td>Trigger Mode:
0: Disable the corresponding alarm.
1: Enable the corresponding alarm.
<table>
<tr><td>Bit No.</td><td>Alarm Status Bit Definition</td></tr>
<tr><td>0</td><td>Alarm when exceeding upper limit</td></tr>
<tr><td>1</td><td>Alarm when below lower limit</td></tr>
<tr><td>2</td><td>Alarm when exceeding Ddelta range</td></tr>
<tr><td>3~7</td><td>Reserved</td></tr>
</table></td><td>Channel 0~15 Alarm Trigger Type.

Please refer to the list on the left for selecting the alarm trigger type.</td></tr>
<tr><td>0x05</td><td>IEEE-754 Float Data</td><td>Channel 0~15 Analog Input Alarm Upper Limit Value.
* Note: If the input format is an invalid IEEE-754 data, the value cannot be set.</td></tr>
<tr><td>0x06</td><td>IEEE-754 Float Data</td><td>Channel 0~15 Analog Input Alarm Lower Limit Value.
* Note: If the input format is an invalid IEEE-754 data, the value cannot be set.</td></tr>
<tr><td>0x08</td><td>IEEE-754 Float Data</td><td>Channel 0~15 Analog Input Alarm Delta Limit Value
* Note: If the input format is an invalid IEEE-754 data, the value cannot be set.</td></tr>
<tr><td>0x07</td><td>Alarm Status:
0: Alarm not triggered.
1: Alarm triggered.
<table>
<tr><td>Bit No.</td><td>Alarm Status Bit Definition</td></tr>
<tr><td>0</td><td>Alarm when exceeding upper limit</td></tr>
<tr><td>1</td><td>Alarm when below lower</td></tr>
</table></td><td>Channel 0~15 Analog Input Alarm Status.

Please refer to the list on the left for the alarm trigger status indication.</td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td></td><td>limit</td></tr>
<tr><td>2</td><td>Alarm when exceeding Delta range</td></tr>
<tr><td>3~7</td><td>Reserved</td></tr>
</table></td><td></td></tr>
<tr><td>0x09</td><td>0x00: Differential
0x01: Single-End</td><td>Channel 0~15 Analog Input Channel Mode.
* Note: If any channel’s analog input code type is set to current measurement mode, switching to Single mode is not allowed.</td></tr>
<tr><td>0x0A</td><td>IEEE-754 Float Data</td><td>Channel 0~15 Analog Input Channel Offset.
* Note: If the input format is an invalid IEEE-754 data, the value cannot be set.</td></tr>
<tr><td>0x0B</td><td>Latch Alarm Status Option:
0: Not latched
1: Latched<table>
<tr><td>Bit No.</td><td>Latched Alarm Status Bit Definition</td></tr>
<tr><td>0</td><td>Alarm when exceeding upper limit</td></tr>
<tr><td>1</td><td>Alarm when below lower limit</td></tr>
<tr><td>2</td><td>Alarm when exceeding fluctuation range</td></tr>
<tr><td>3~7</td><td>Reserved</td></tr>
</table></td><td>Channel 0~15 Analog Input Alarm Latch Status.</td></tr>
<tr><td>0x0C</td><td>1: Clear latched alarm status.</td><td>Clear Latched Alarm Status of Channel 0~15</td></tr>
<tr><td>0x0D</td><td>0: Disable channel
1: Enable channel.</td><td>Channel 0~15 Analog Input Channel Enable</td></tr>
</table>

Instance Service

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

3.6 Application Object2 (Class ID: 0x65)

Application Object 2 defines parameters for saving configurations into EEPROM or loading factory default setting.

Class Attribute (Instance ID=0)

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

Class Service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes

Instance Attribute (**Instance** ID=0x01)

Attribute ID	Description	Method	Data Type	Default Value
0x01	Save all configurations into EEPROM or using factory default configuration setting	USINT	Set	-

Parameter description of Application Object 2 attributes

Attribute ID	Data Range	Parameter Description
0x01	0x01: Use default configuration 0x02: Save all configurations to EEPROM	0x01: After restarting the device, all configurations will become default setting. 0x02: Save all channels configuration into EEPROM.

Instance Service

Service Code	Service name	Support
0x10	Set_Attribute_Single	Yes

4 Application

The application objects (Class ID: 0x64, 0x65) list all parameters of the module. Each Instance ID corresponds to a different channel. By using the Set/Get_Attribute_Single service, users can read/write the parameters of each channel.

Example 1:
Retrieve Analog Input Value of Channel 0.
(Class ID: 0x64, Instance ID: 0x01, Attribute ID: 0x01)
Assume that the CAN-2012D has a node ID of 0x01 and the master (ID: 0x00) is fully connected to the device (Explicit). By accessing the object's Attribute ID (0x01), the user can retrieve the analog input value of Channel 0.

CAN ID											RTR	LEN	Data (HEX)							
		Target MAC ID																		
10	9	8	7	6	5	4	3	2	1	0			0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1	1	0	0	0	5	00	0E	64	01	01	--	--	--

Master → **Slave (CAN-2012D)**

CAN ID											RTR	LEN	Data (HEX)							
		Source MAC ID																		
10	9	8	7	6	5	4	3	2	1	0			0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1	0	1	1	0	6	00	8E	A0	00	20	3A	--	--

Master ← **Slave (CAN-2012D)**

Retrieve the value of Instance ID: 0x01, Attribute ID: 0x01 in Application Object 1. After sending the "Get Attribute Single" service, the slave device will respond with the analog input data. The value 0x3A2000A0 represents the input data obtained from Channel 0. According to IEEE-754 conversion, this is approximately equal to 0.0006103515625.

Example 2:

Channel 0 Upper Limit Alarm Trigger.

Assume that the CAN-2012D has a node ID of 0x01 and the master (ID: 0x00) is fully connected to the device (Explicit). When the analog input value of Channel 0 exceeds the upper limit, by configuring the parameters of Attribute IDs 0x03, 0x04, and 0x05 of the object, the user can observe that the upper limit alarm LED indicator is turned on.

1. Enable Alarm Trigger
   Class ID: 0x64, Instance ID: 0x01, Attribute ID: 0x03

CAN ID											RTR	LEN	Data (HEX)							
		Target MAC ID																		
10	9	8	7	6	5	4	3	2	1	0			0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1	1	0	0	0	6	00	10	64	01	03	01	--	--

Master → **Slave (CAN-2012D)**

CAN ID											RTR	LEN	Data (HEX)							
		Source MAC ID																		
10	9	8	7	6	5	4	3	2	1	0			0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1	0	1	1	0	2	00	90	--	--	--	--	--	--

Slave (CAN-2012D) → **Master**

Set the value of Instance ID: 0x01, Attribute ID: 0x03 in Application Object 1 to 0x01.

After sending the “Set_Attribute_Single” service, the slave device will respond with 0x90, which indicates that the setting is successful. Thus, the alarm trigger status is enabled.

2. Set Alarm Trigger Type
   Class ID: 0x64, Instance ID: 0x01, Attribute ID: 0x04

CAN ID											RTR	LEN	Data (HEX)							
		Target MAC ID																		
10	9	8	7	6	5	4	3	2	1	0			0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1	1	0	0	0	6	00	10	64	01	04	01	--	--

Master → **Slave (CAN-2012D)**

CAN ID											RTR	LEN	Data (HEX)							
		Source MAC ID																		
10	9	8	7	6	5	4	3	2	1	0			0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1	0	1	1	0	2	00	90	--	--	--	--	--	--

Slave

Master ←

(CAN-2012D)

Set the value of Instance ID: 0x01, Attribute ID: 0x04 in Application Object 1 to 0x01. After sending the “Set_Attribute_Single” service, the slave device will respond with 0x90, which indicates that the setting is successful. Thus, the trigger type will be set to Upper Limit Trigger Mode. Setting the corresponding value of this Attribute ID to 0x02 will change it to Lower Limit Trigger Mode**.** Setting the value to 0x03 will enable both modes simultaneously. Setting the value to 0x00 will disable triggering.

Bit no.	3~7	2	1	0
Alarm Trigger Type Bit Definition	Reserved	Triggered when exceeding Delta value	Triggered when below lower limit	Triggered when exceeding upper limit

3. Set Upper Limit Value
   Class ID: 0x64, Instance ID: 0x01, Attribute ID: 0x05
   Section 1:

CAM ID											RTR	LEN	Data (HEX)							
		Target MAC ID																		
10	9	8	7	6	5	4	3	2	1	0			0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1	1	0	0	0	8	80	00	10	64	01	05	30	00

Slave

Master →

(CAN-2012D)

CAN ID											RTR	LEN	Data (HEX)							
		Source MAC ID																		
10	9	8	7	6	5	4	3	2	1	0			0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1	0	1	1	0	3	80	C0	00	--	--	--	--	--

Slave

Master ←

(CAN-2012D)

Section 2:

CAN ID											RTR	LEN	Data (HEX)							
		Target MAC ID																		
10	9	8	7	6	5	4	3	2	1	0			0	1	2	3	4	5	6	7

1	0	0	0	0	0	0	1	1	0	0	0	4	80	81	00	00	--	--	--	--

Master → **Slave (CAN-2012D)**

CAN ID											RTR	LEN	Data (HEX)							
		Source MAC ID																		
10	9	8	7	6	5	4	3	2	1	0			0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1	0	1	1	0	3	8A	C1	00	--	--	--	--	--

Master ← **Slave (CAN-2012D)**

CAN ID											RTR	LEN	Data (HEX)							
		Source MAC ID																		
10	9	8	7	6	5	4	3	2	1	0			0	1	2	3	4	5	6	7
1	0	0	0	0	0	0	1	0	1	1	0	2	00	90	--	--	--	--	--	--

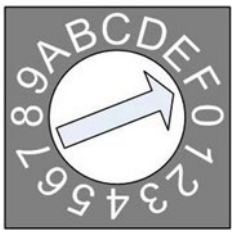
Master ← **Slave (CAN-2012D)**

Set the value of Instance ID: 0x01, Attribute ID: 0x05 in Application Object 1 to 0x00000000. Since it exceeds the maximum data length limit (8 Bytes), it needs to be divided into two stages for processing. After sending all segmented data using the “Set_Attribute_Single” service, the slave device will respond with 0x90, which indicates that the setting is successful. When the analog input value of Channel 0 exceeds 0x00000000, the upper limit alarm LED indicator will be turned on. Attribute ID 0x05 in Application Object 1 can be used to set the threshold value of the upper limit alarm. The value 0x00000000 represents an analog input value of **+**0.

5 Firmware Update

Step 1: Set Module to “Bootloader” mode

Set Module to “Bootloader” mode (set baud rate to 0xF). Then power on the module. After power on, the module’s led (PWR, NET, MOD) will be flashed at the same time. It means that the module have entered into “Bootloader” mode.



Baud Rate Rotary Switch

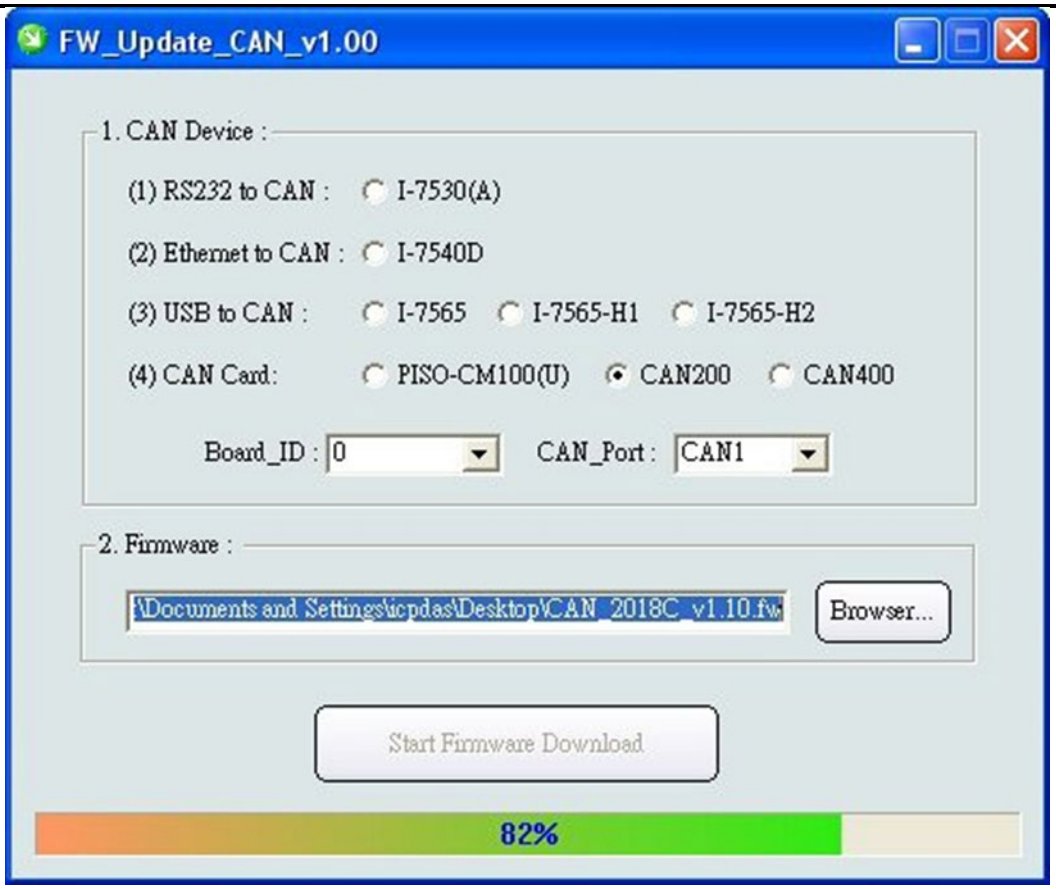
Step 2: Get Module firmware

Module firmware can be downloaded from following link:
https://www.icpdas.com/tw/product/CAN-2012D

Step 3: Run FW_Update_CAN Utility

FW_Update_CAN Utility can be downloaded from following link:
https://www.icpdas.com/tw/product/CAN-2012D

Run FW_Update_CAN Utility



[1] CAN Device:

The below ICP DAS CAN products are supported by FW_Update_CAN utility for firmware update.

(1) RS232 to CAN **:** I-7530
(2) Ethernet to CAN **:** I-7540D
(3) USB to CAN **:** I-7565, I-7565-H1, I-7565-H2
(4) CAN Card**:** PISO-CM100(U), PISO-/PCM-/PEX-CAN200 / CAN400

Before firmware update, users need to set the below parameters:

(1) Select CAN hardware interface.

(2) Set Dev_Port or Board_ID.

(3) Set ”CAN_Port”.

[2] Download Firmware :

(1) Click “Browser…” button to choose firmware file.

(2) Click “Start Firmware Update” button to start firmware update and it will show the total percentage of firmware update in progress bar. After the firmware update finished, it will show the “Firmware Update Success !!” message.

6 Troubleshooting

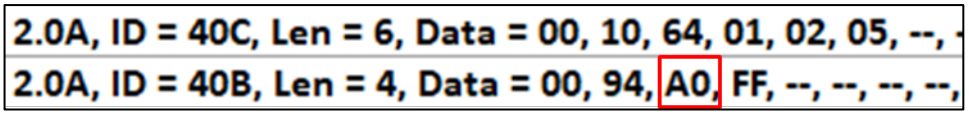
Q: Why does it fail when setting the Type Code to 5/6/7 (current measurement)?

A:

In Single-Ended mode, it is not possible to set the Type Code to 5/6/7 (current measurement). Please switch the channel mode to Differential mode first, and then set the Type Code to 5/6/7 (current measurement).

(Current measurement is only supported in Differential mode.)

You can analyze the message returned by the CAN-2012D to determine the exact cause of the error. If the return code is **0xA0**, it indicates that the device is currently in Single-Ended mode.



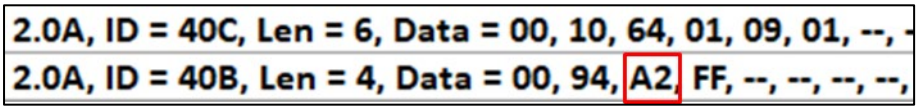
Q: Why does switching to Single-Ended mode fail?

A:

When the CAN-2012D channel mode fails to switch from Differential mode to Single-Ended mode, please verify that none of the channels are set to current measurement mode (Type Code: 5/6/7).

(Current measurement is only supported in Differential mode.)

You can analyze the message returned by the CAN-2012D to determine the exact cause of the error. If the return code is **0xA2**, it indicates that one or more channels are currently in current measurement mode.

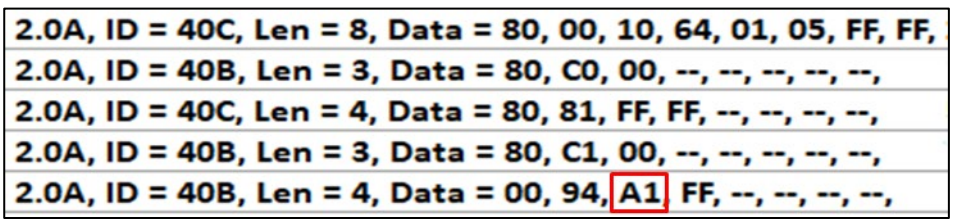


Q: Why does setting the upper limit, lower limit, delta range, or offset fail?

A:

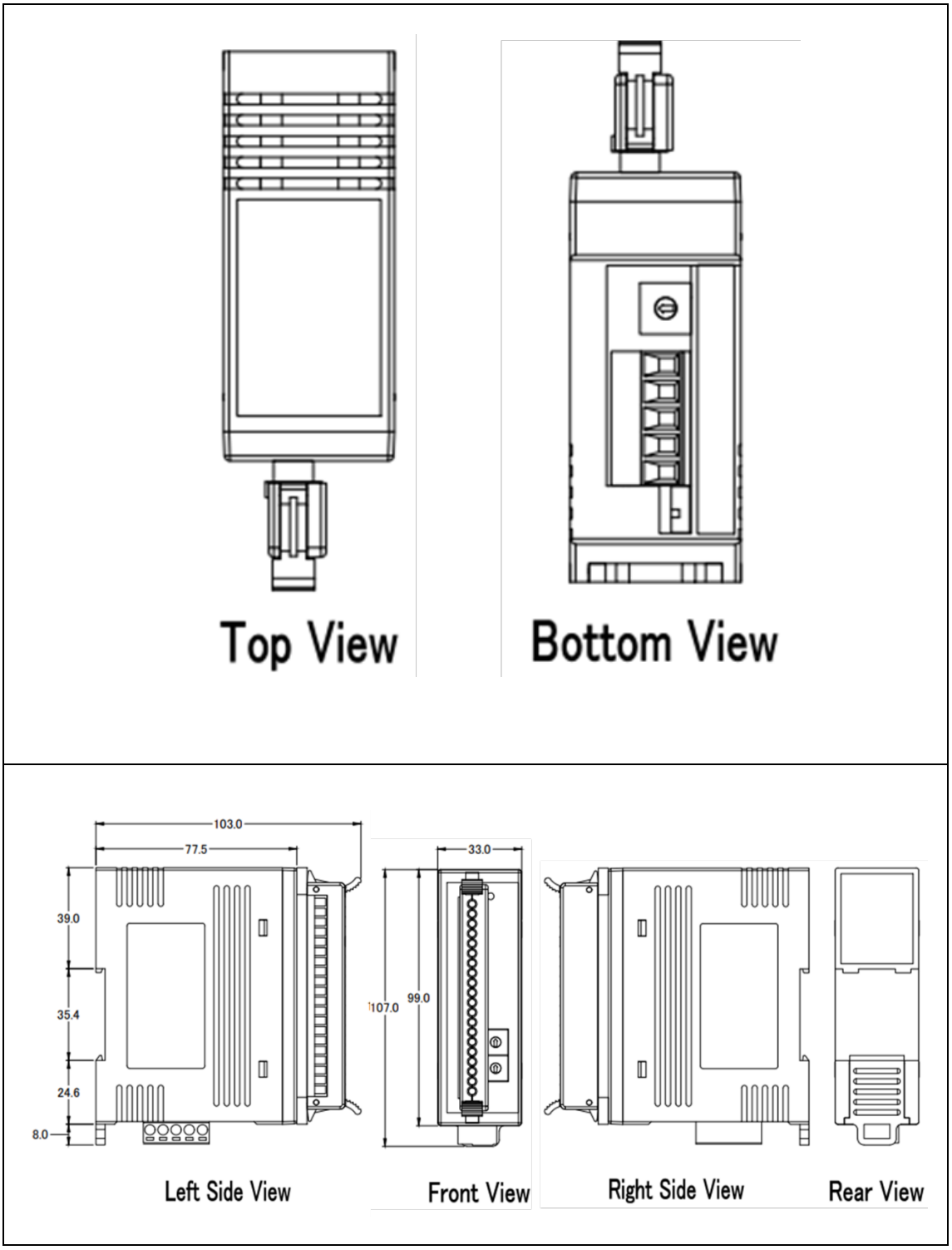
When an error occurs while setting the upper limit, lower limit, delta range, or offset of the CAN-2012D, please check whether the input data complies with the IEEE-754 floating-point format (Note: little-endian).

You can analyze the message returned by the CAN-2012D to determine the exact cause of the error. If the return code is **0xA1**, it indicates that the input value does not comply with the IEEE-754 standard.



Appendix A: Dimensions

CAN-2012D Dimensions:



Unit: mm

Appendix B: Type Code Definitions

Type code definitions for analog input of CAN-2012D

Type Code	Input Range	Date Format	Max	Min	**Channel mode**
01	-10 to +10V	IEEE-754 float	-10V	+10V	Single or Differential
02	-5 to +5V		-5V	+5V	Single or Differential
03	-2.5 to +2.5V		-2.5V	+2.5V	Single or Differential
04	0 to +10 V		0V	+10V	Single or Differential
05	-20 to +20 mA		-20mA	-20mA	Differential
06	0 to +20 mA		0mA	+20mA	Differential
07	-4 to +20 mA		+4mA	+20mA	Differential