DeviceNet Slave Device

CAN-2019D

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 one kind of the network protocols based on the CAN bus and mainly used for the embedded network of the machine control, such as industrial machine control , aircraft engines monitoring, factory automation, medical equipments control, remote data acquisition, environmental monitoring, and packaging machines control, etc.

The CAN-2019D complies with the DeviceNet Specification Volume I/II, Release 2.0. Users can access analog I/O status and perform configuration through the DeviceNet EDS file. This module features 10 channels of universal AI, which can be used to measure voltage, current (requires an external 125Ω resistor), and thermocouples (J, K, T, E, R, S, B, N, C). If you have a DeviceNet master device from ICP DAS, you can quickly establish a DeviceNet network to meet your requirements.

1.2 Hardware Specifications

Analog Input:

- Input Channels: 10 (Differential).
- Input Type: +/- 15mV, +/- 50mV, +/- 100mV, +/- 500mV, +/- 1V, +/- 2.5V, +/- 5V, +/- 10V
  +/-20mA,(Requires Optional External 125 Ohm Resistor)
  Thermocouple (J, K, T, E, R, S, B, N, C)
- Resolution: 16 -bit
- Sampling Rate: 10 Samples/Sec. (Total)
- Accuracy: +/-0.1% FSR
- Input Impedance: >400 kΩ.
- Common Mode Rejection: 86 dB Min.
- Normal Mode Rejection: 100 dB
- Zero Drift: +/- 10 μV/ ℃
- Span Drift: +/-25 ppm/ ℃
- Intra-module Isolation, Field to Logic: 3000 VDC
- Over voltage Protection: 240 Vrms
- 4KV ESD Protection: Yes, Contact for each terminal.

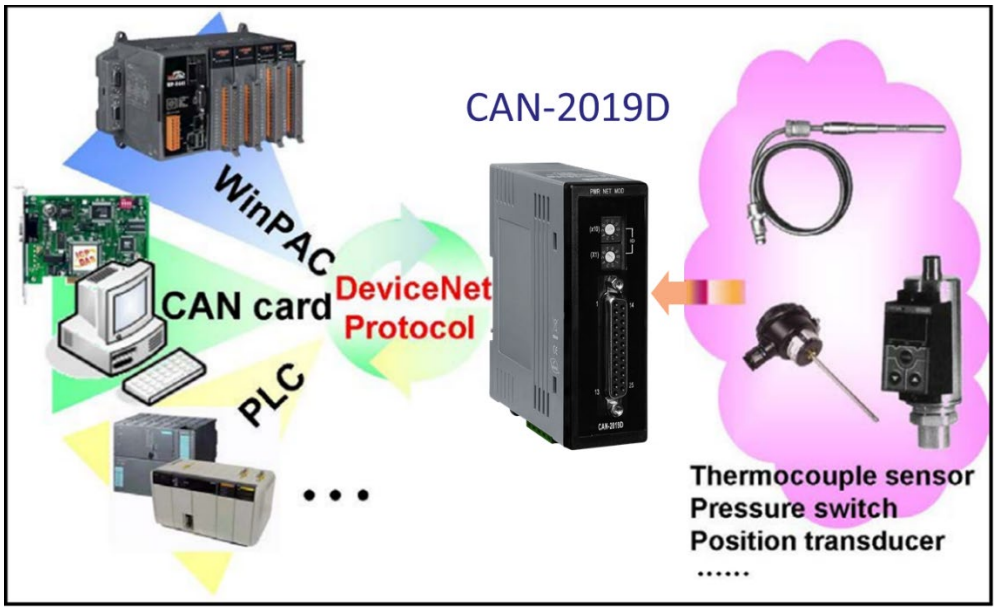
Others:

- DeviceNet Status: 3 LEDs for PWR / NET / MOD.
- Termination Resistor: 120Ω termination resistor DIP switch
- Power Supply: Unregulated +10 ~ +30 VDC.
- Power Consumption：1.5 W
- Operating Temperature: -25 ~ 75 ℃.
- Storage Temperature: -30 ~ 80 ℃.
- Humidity: 10 to 90% RH, Non-condensing.
- Dimensions: 33 mm x 99 mm x 78 mm (W x L x H) .

1.3 Features

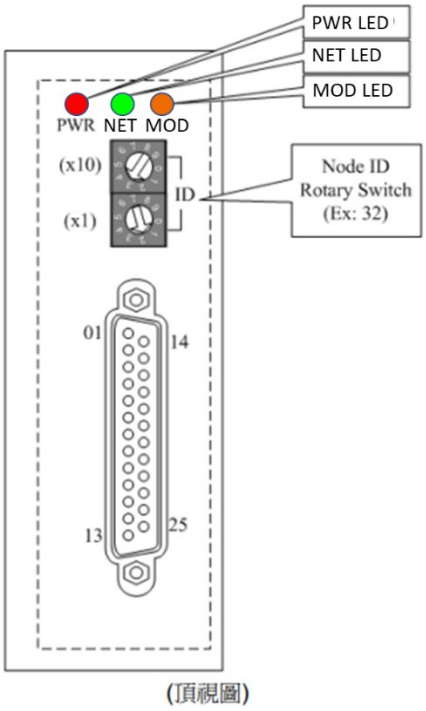
- DeviceNet general I/O slave devices.
- Comply with DeviceNet specification Volume I, Release 2.0 & Volume II, Release 2.0, Errata 5.
- Group 2 Only Server (non UCMM-capable).
- Support Predefined Master/Slave Connection Set.
- Connection supported:
  - 1 connection for Explicit Messaging
  - 1 connection for Polled I/O
  - 1 connection for Bit-Strobe I/O connection
- Support DeviceNet heartbeat and shutdown messages
- Provide EDS file for standard DeviceNet master interface.
- NET, MOD and PWR Led indictors

1.4 Application

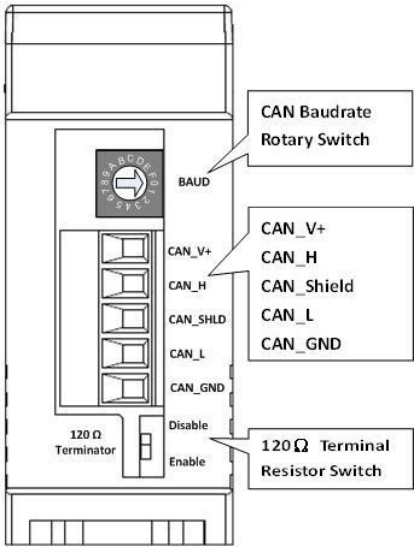


2 Hardware

2.1 Structure



(Top View)

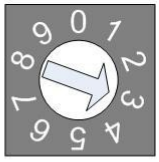


(Bottom View)

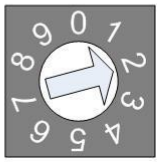
2.2 Node ID & Baud Rate Rotary Switch

The rotary switches of node ID configure the node ID of CAN-2019D 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.

(x10)



(x1)



節點 ID 旋轉開關

The rotary switch for baud rate handles the CAN baud rate of CAN-2019D 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

PWR LED

The CAN-2019D needs the power of 10 ~ 30 VDC. Under a normal connection, a good power supply and a correct voltage selection, as the unit is turned on, the LED will light up in 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

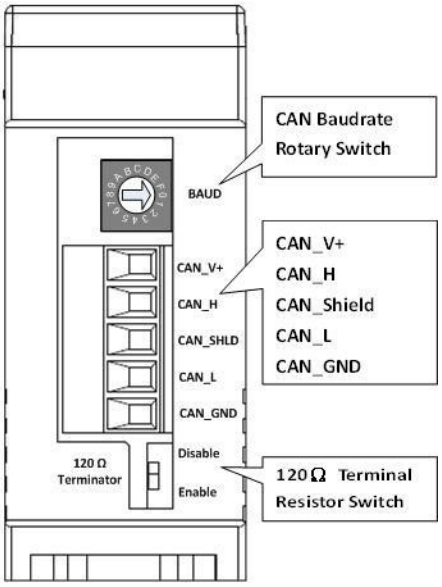
MOD LED

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

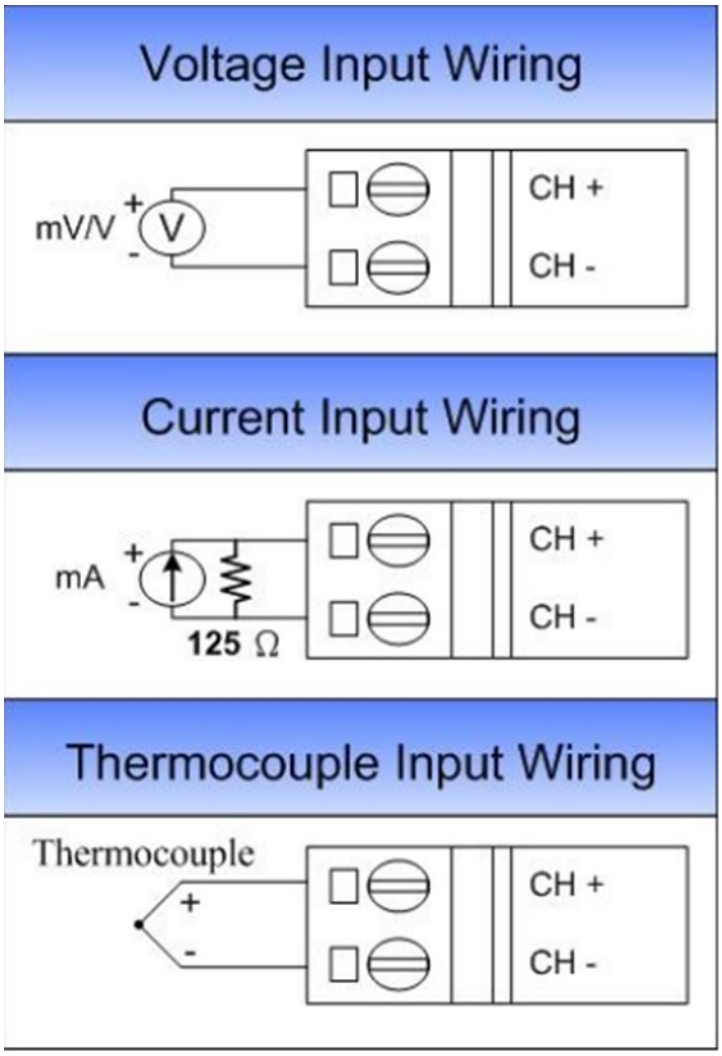
condition	status	indicates
Off	Normal	
Solid	Critical fault	Device has unrecoverable fault.
Flashing	Non_critical fault	Device has recoverable fault to recover. If users want to fix the problem, reconfiguring device's MAC ID or resetting device may work.

2.4 PIN Assignment

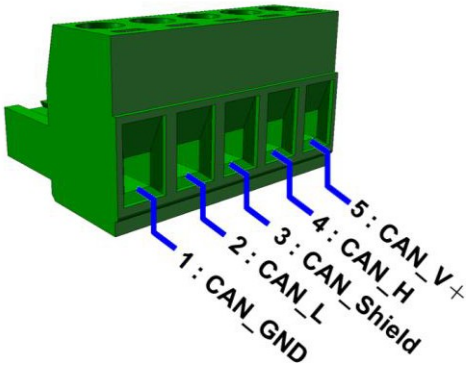
Pin Assignment Name	Terminal No.		Pin Assignment Name
+5V	01	14	DGND
CJC	02	15	CH0+
CH0-	03	16	CH1+
CH1-	04	17	CH2+
CH2-	05	18	CH3+
CH3-	06	19	CH4+
CH4-	07	20	CH5+
CH5-	08	21	CH6+
CH6-	09	22	CH7+
CH7-	10	23	CH8+
CH8-	11	24	CH9+
CH9-	12	25	AGND
AGND	13		
		Shield	F.G.



2.5 Wire Connection



5-pin screw terminal block



3 DeviceNet Profile Area

This section describes the detailed functions for each object class that is implemented in the CAN-2019D 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-2019D
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
Predefined Master/Slave Connection Set	Group 2 Only Server

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
1	Vendor	Get	UINT	803
2	Product type	Get	UINT	0x00
3	Product code	Get	UINT	0x303
4	Major. Minor of firmware version	Get	Struct of USINT USINT	1.1
5	Status	Get	WORD	-
6	Serial number	Get	UDINT	1
7	Product name	Get	Short_String	CAN-2019D
10	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
Bit Strobe Produced	0x64	0x01	0x0D
Bit Strobe Consumed	0x65	0x01	0x01

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

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	1
0x02	Max Instance	UINT	Get	0x02

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 ~ 9 AI Value	IEEE-754 Float	Get	0x00
0x65	Channel 0 ~ 9 Alarm Status	BYTE	Get	0x00

Parameter description of Assembly Object Instance

Instance ID	Data Range	Parameter Description
0x64	Refer to Appendix B.	Channel 0 ~ 9 AI Value
0x65	0x00 ~ 0x07	Bit 0 => Over High Limit Alarm Bit 1 => Below Low Limit Alarm Bit 2 => Out of Hysteresis Range Alarm Bit 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
0x10	Set_Attribute_Single	No

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-2019D module.

Application Object 1 defines the configuration of analog input channels:

Class attribute (Instance =0)

Attribute ID	Attribute name	Data Type	Method	Value
0x01	Revision	UINT	Get	1
0x02	Max Instance	UINT	Get	0x0F

Class service

Service Code	Service name	Support
0x0E	Get_Attribute_Single	Yes

Instance attribute (Instance ID=0x01~0x0F)

Attribute ID	Description	Method	Data Type	Default Value
0x01	AI Value	Get	IEEE-754 Float	0.0f
0x02	AI Type Code	Get/Set	BYTE	0x0E
0x03	AI Offset	Get/Set	IEEE-754 Float	0.0f
0x04	Enable AI Channel	Get/Set	BYTE	0x01
0x05	Alarm Type	Get/Set	BYTE	0x00
0x06	Enable Alarm	Get/Set	BYTE	0x00
0x07	Upper Limit Alarm Value	Get/Set	IEEE-754 Float	Float MAX: 0x7F7FFFFF
0x08	Lower Limit Alarm Value	Get/Set	IEEE-754 Float	Float MIN:

				0xFF7FFFFF
0x09	Hysteresis Range Limit Alarm Value	Get/Set	IEEE-754 Float	Float MAX: 0x7F7FFFFF
0x0A	Alarm Status	Get	BYTE	0x00
0x0B	Latch Alarm	Get/Set	BYTE	0x00
0x0C	Reset Alarm	Set	BYTE	0x00
0x0D	CJC Temperature	Get	UINT16	0x0000
0x0E	CJC Enable	Get/Set	BYTE	0x01
0x0F	CJC Offset	Get/Set	UINT16	0x00

Parameter description of Application Object1 attributes

Attribute ID	Data Range	Parameter Description
0x01	IEEE-754 Float	AI Value。 Little-Endian。 Refer to Appendix B for the measurement range.
0x02	Refer to Appendix B for the measurement range.	AI Measurement Type.
0x03	IEEE-754 Float	AI Offset。 Little-Endian。
0x04	0x00: Disable AI Channel. 0x01: Enable AI Channel.	Allows control over whether a specific AI channel is enabled. ***Note: When disabled, the alarm status of the channel will be cleared.**
0x05	Data Range: 0x00~0x07 Bit 0: Upper Limit Alarm. Bit 1: Lower Limit Alarm. Bit 2: Hysteresis Range Limit Alarm. Bit 3~7: Reserved. Value: 0 => Disable. 1 => Enable.	Set Alarm Type.
0x06	0x00: Disable Alarm. 0x01: Enable Alarm.	Whether to enable the alarm.
0x07	IEEE-754 Float	Alarm high limit value. ***Note: If an invalid IEEE-754 value is entered, the CAN-2019D will reject the operation. To check the type of error, please refer to Appendix C.**
0x08	IEEE-754 Float	Alarm low limit value. ***Note: If an invalid IEEE-754 value is entered, the CAN-2019D will reject the operation. To check the type of error, please refer to Appendix C.**

0x09	IEEE-754 Float	Alarm hysteresis value. ***Note: If an invalid IEEE-754 value is entered, the CAN-2019D will reject the operation. To check the type of error, please refer to Appendix C.**
0x0A	Data Range: 0x00~0x07 Bit 0: Upper Limit Alarm. Bit 1: Lower Limit Alarm. Bit 2: Hysteresis Range Limit Alarm. Bit 3~7: Reserved. Value: 0 => Alarm not triggered. 1 => Alarm triggered.	Alarm trigger status.
0x0B	Data Range: 0x00~0x07 Bit 0: Upper Limit Alarm. Bit 1: Lower Limit Alarm. Bit 2: Hysteresis Range Limit Alarm. Bit 3~7: Reserved. Value: 0 => Disable alarm latching. 1 => Enable alarm latching.	Alarm Latching Function: This function retains the status when an alarm is triggered. For example, if both the high limit alarm and alarm latching are enabled, when the CAN-2019D detects a sudden input that triggers the high limit alarm, the alarm will be latched and will not be cleared by subsequent stable input values. Clearing Latched Alarms: You can use the “Clear Alarm” function to reset the alarm status.
0x0C	0x01	Reset alarm status.
0x0D	0x0000~0xFFFF	CJC Temperature. Unit: 0.1 ℃ For the data format, please refer to Appendix B.
0x0E	0x00: Disable. 0x01: Enable.	Whether to use CJC. Enabled: Uses the measured CJC value. Disabled: Uses the CJC offset value.

0x0F	0x03E8~0xFED4	CJC Offset Value. For the data format, please refer to Appendix B. ***Note: If an invalid CJC offset value is entered, the CAN-2019D will reject the operation. To check the type of error, please refer to Appendix C.**

Instance service

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

3.6 Application Object 2 (Class ID: 0x65)

Application Object2 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	0x01
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	Set	USINT	-

Parameter description of Application Object2 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

1 Application

Application Object1 (Class ID:0x64) lists all the parameters of the module. Each Instance ID is corresponding to the different channels. By using “Set/Get Attribute Single” service, user can read/write the parameters of each channel.

Example 1:

Set CJC Offset。

(Class ID: 0x64, Instance ID: 0x01, Attribute ID 0x0F).

If the CAN-2019D's node ID is 1 and the master (ID: 0x0A) has already established an "Explicit" connection with the device, the CJC offset can be set to 10 by setting the value of Attribute ID 0x0F to 0x0A.

IDENTIFIER BITS											RTR	Data Length	8-byte Data (byte) (HEX)							
		Destination 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	7	0A	10	64	01	0F	0A	00	-

Master → **Slave (CAN-2019D)**

IDENTIFIER BITS											RTR	Data Length	8-byte Data (byte) (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	0A	90	--	--	--	--	--	--

Slave (CAN-2019D) → **Master**

Set the value of Application Object 1, Instance ID 0x01, Attribute ID 0x0F to 0x0A.

After sending the “Set Attribute Single” command, the slave device will respond with 0x90 to indicate a successful setting. The CJC offset will then be applied accordingly.

By modifying the Attribute ID of the Application Object, users can configure other parameters of the device.

Example **2:**

Get channel0 AI data (Class ID: 0x64, Instance ID: 0x01, Attribute ID 0x01)。

If the node ID of the CAN-2019D is 1, and the master (ID: 0x0A) has completed “Explicit” connection with the device. By getting the value of the object with attribute ID 0x01, you can get the channel 0 of the AI data.

IDENTIFIER BITS											RTR	Data Length	8-byte Data (byte) (HEX)							
		Destination 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	0A	0E	64	01	01	--	--	--

Master → **Slave (CAN-2019D)**

IDENTIFIER BITS											RTR	Data Length	8-byte Data (byte) (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	0A	8E	00	00	A0	40	-	-

Master ← **Slave (CAN-2019D)**

Obtain the value of Instance ID 0x01 and Attribute ID 0x01 of Application Object 1.

After sending the "Get Attribute Single" command, the slave device responds with AI data (0x40a00000) in bytes 2 to 5.

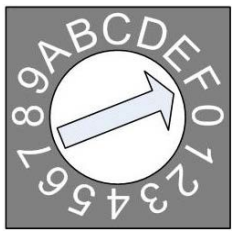
0x40a00000 represents the value 5 in IEEE-754 floating-point format.

For the corresponding data type, please refer to Appendix B. (For example, if Type 0x08 is used, then 0x40a00000 represents 5 V)

2 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.



鮑率旋鈕開關

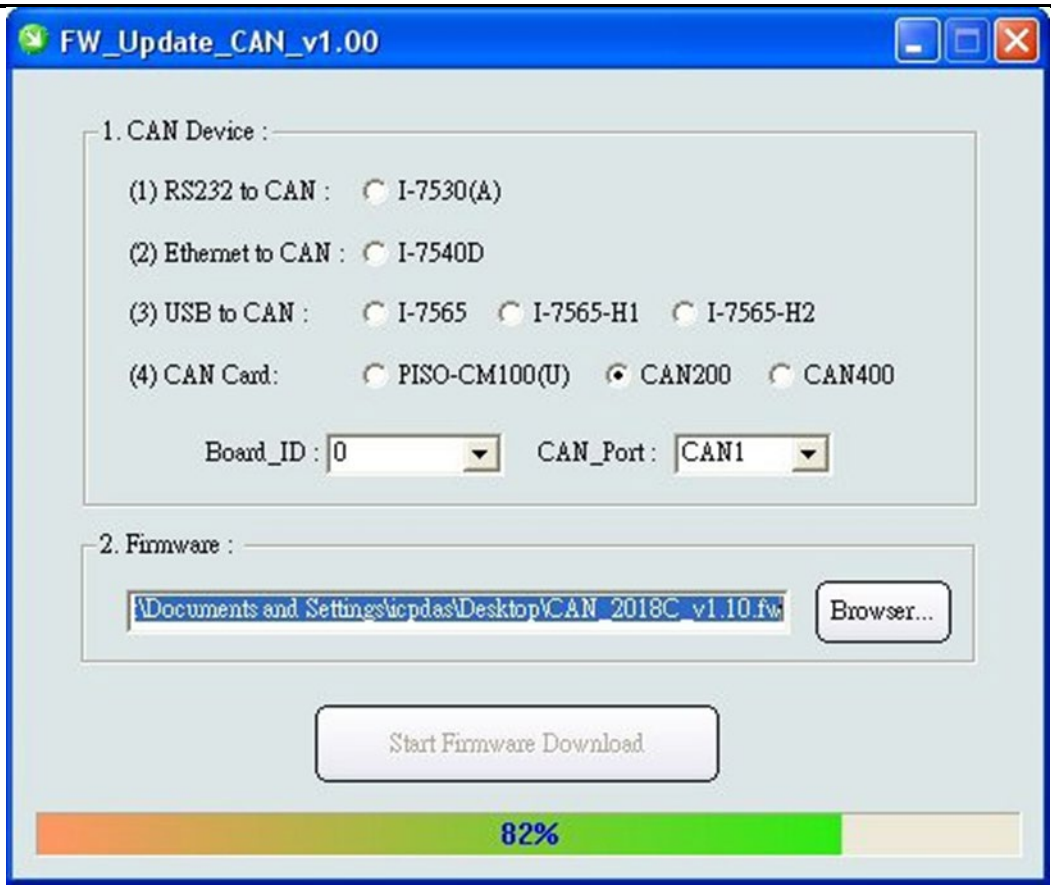
Step 2: Get Module firmware

Module firmware can be downloaded from following link:
https://www.icpdas.com/tw/download/index.php?model=CAN-2019D/S

Step 3: Run FW_Update_CAN Utility

FW_Update_CAN Utility can be downloaded from following link:
https://www.icpdas.com/tw/download/index.php?model=CAN-2019D/S

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.

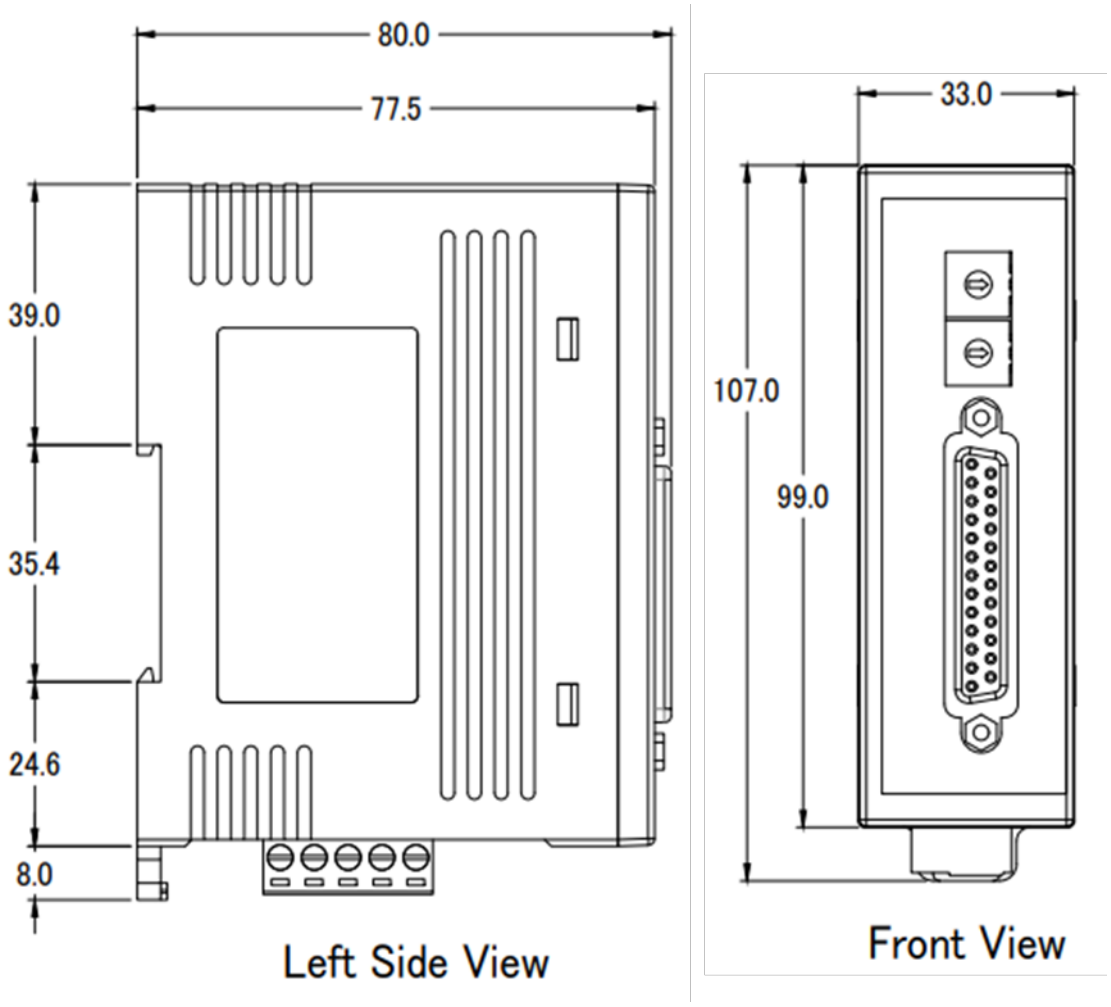
(3) Set ”CAN_Port”.

[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.

Appendix A: Dimension



Appendix B: Type Code Definitions

Type code definitions for analog input of CAN-2019D

Type Code	Input Range	Date Format	Max	Min
00h	-15 to +15mV	IEEE-754	+15	-15
01h	-50 to +50mV	IEEE-754	+50	-50
02h	-100 to +100mV	IEEE-754	+100	-100
03h	-500 to +500mV	IEEE-754	+500	-500
04h	-1 to +1V	IEEE-754	+1	-1
05h	-2.5 to +2.5V	IEEE-754	+2.5	-2.5
06h	-20 to +20mA (with 125Ω resistor)	IEEE-754	+20	-20
08h	-10 to +10V	IEEE-754	+10	-10
09h	-5 to +5V	IEEE-754	+5	-5
0Eh (Default)	J Type (℃)	IEEE-754	+1200	-210
0Fh	K Type (℃)	IEEE-754	+1372	-270
10h	T Type (℃)	IEEE-754	+400	-270
11h	E Type (℃)	IEEE-754	+1000	-270
12h	R Type (℃)	IEEE-754	+1765	-50
13h	S Type (℃)	IEEE-754	+1765	-50
14h	B Type (℃)	IEEE-754	+1820	0
15h	N Type (℃)	IEEE-754	+1300	-270
16h	C Type (℃)	IEEE-754	+2320	0

***Open-circuit detection is supported only in thermocouple mode. When an open circuit is detected, the CAN-2019D will return the maximum measurement value for the currently selected thermocouple type, in order to avoid confusion with normal measurement results.**
(For example: when an open circuit occurs with J type, the returned value will be 1200.)

CJC (Cold Junction Compensation) of CAN-2019D

Input Range	Data Format	Max	Min
-30 to +100 (℃)	Engineer Unit	+100	-30
	2's Complement HEX	03E8h	FED4h

Appendix C: Troubleshooting

Q1: Why does setting the attribute Upper_Limit / Lower_Limit / Delta / Offset fail?

A: Please verify that the data you entered is in a valid IEEE-754 format, or you can use the method described in Q2 to obtain the error type of the issue.

Q2: How can I determine whether the failure of Set Attribute is due to an invalid IEEE-754 formatted input?

A: You can observe the CAN BUS data to obtain the Explicit Error Response information. In the error code, the CAN-2019D uses a custom error code "0xA1" to indicate that the IEEE-754 data is invalid.

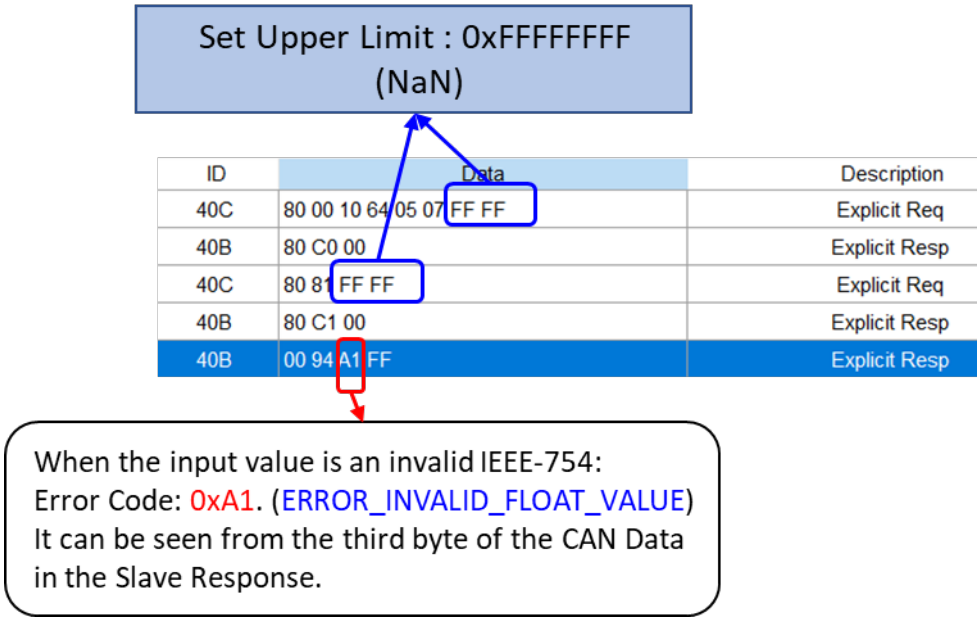


Figure: Example of Invalid IEEE-754 Format

Q3: Why does Set CJC Offset fail?

A: If the value you entered is outside the valid CJC range (please refer to Appendix B), the CAN-2019D will reject the current operation. You can observe the CAN BUS data to obtain the Explicit Error Response information. In the error code, the CAN-2019D uses a custom error code "0xA2" to indicate an invalid CJC offset value.

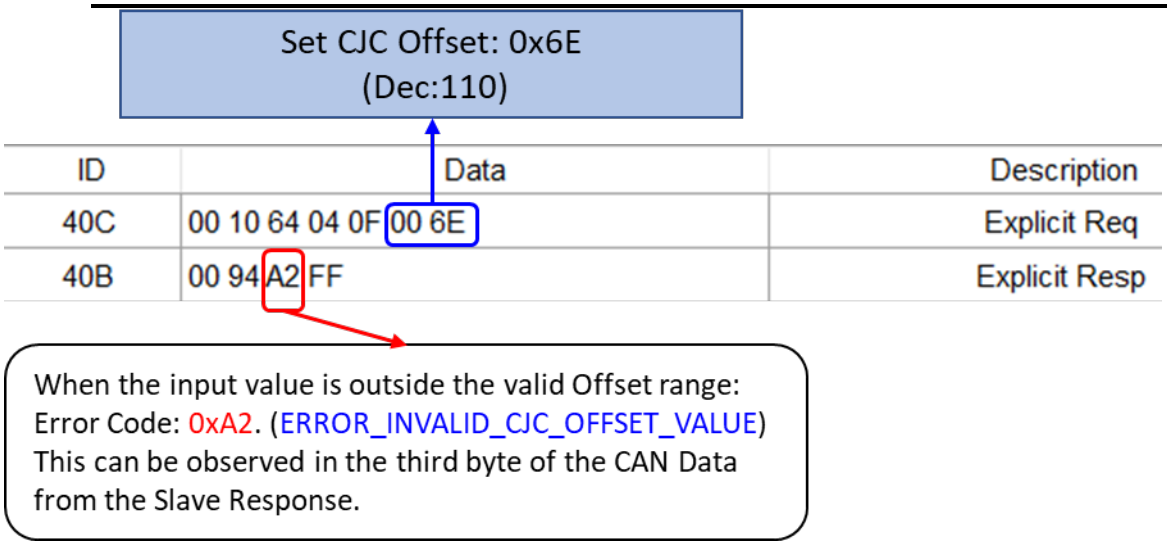


Figure: Example of Invalid CJC Offset