DeviceNet Slave Device

CAN-2084D

User Manual

Warranty Terms

All products manufactured by ICP DAS are provided with a one-year warranty for the product itself. The warranty period begins on the date the product is delivered by our company to the original purchaser.

Notes

ICP DAS makes no warranty for any damage caused by the use of this product and reserves the right to revise this document at any time without prior notice. This document provided by ICP DAS is believed to be accurate and reliable; however, ICP DAS assumes no responsibility for the use of this document and makes no commitment regarding any infringement of patents or other rights of third parties that may result from its use.

Trademark

ICP DAS is a registered trademark of ICP DAS and may be used by other authorized companies.

Contents

DEVICENET SLAVE DEVICE 1

USER MANUAL 1

Warranty Terms *1*

Notes *1*

Copyright *1*

Trademark *1*

1. INTRODUCTION 4

1.1 OVERVIEW 4

1.2 FEATURES 5

1.3 HARDWARE SPECIFICATIONS 5

2. HARDWARE 8

2.1 STRUCTURE 8

2.2 NODE ID & BAUD RATE ROTARY SWITCH 9

2.3 LED DESCRIPTION 10

POWER LED *10*

NET LED *10*

MOD LED *10*

Terminal Resistor Indicator *10*

I/O LED *10*

2.4 PIN ASSIGNMENT 11

2.6 INPUT SIGNAL MODEL 13

Isolated Input (XOR=0) *13*

TTL Input (XOR=1) *13*

2.7 DIGITAL LOW PASS FILTER 14

2.8 COUNTER MODE 15

2.8.1 MODE 0：DIR/PULSE COUNTING(BI-DIRECTION) 16

2.8.2 MODE 1：UP/DOWN COUNTING(BI-DIRECTION) 16

2.8.3 MODE 2：FREQUENCY MODE 16

2.8.4 MODE 3：UP COUNTING 17

2.8.5 MODE 4：QUADRANT COUNTING(A/B PHASE) 18

3.1 DEVICENET STATEMENT OF COMPLIANCE 19

3.2 IDENTITY OBJECT (CLASS ID: 0X01) 20
3.3 CONNECTION OBJECT (CLASS ID: 0X05) 21
3.4 ASSEMBLY OBJECT (CLASS ID: 0X04) 22
3.5 APPLICATION OBJECT1 (CLASS ID: 0X64) 23
3.6 APPLICATION OBJECT2 (CLASS ID: 0X65) 28
3.7 APPLICATION 29
Example 1: *29*
Example 2: *30*

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-2084D complies with the DeviceNet specification (Volume I/II, Release 2.0). Establishing communication between a DeviceNet master and the CAN-2084D is no longer a difficult task. The DeviceNet Electronic Data Sheet (EDS) enables users to accomplish this quickly and easily.

The CAN-2084D provides 4/8-channel counter/frequency inputs, making it suitable for a wide range of applications. By using a DeviceNet master from ICP DAS, users can easily build a DeviceNet network to meet their requirements.

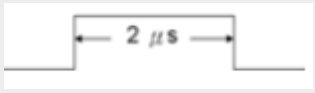
Figure 1-1 CAN-2084D

1.2 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
- Supports I/O connections: Poll、Strobe
- Supports DeviceNet Heartbeat and Shutdown Messages
- Provides Electronic Data Sheet (EDS) for Standard DeviceNet Master

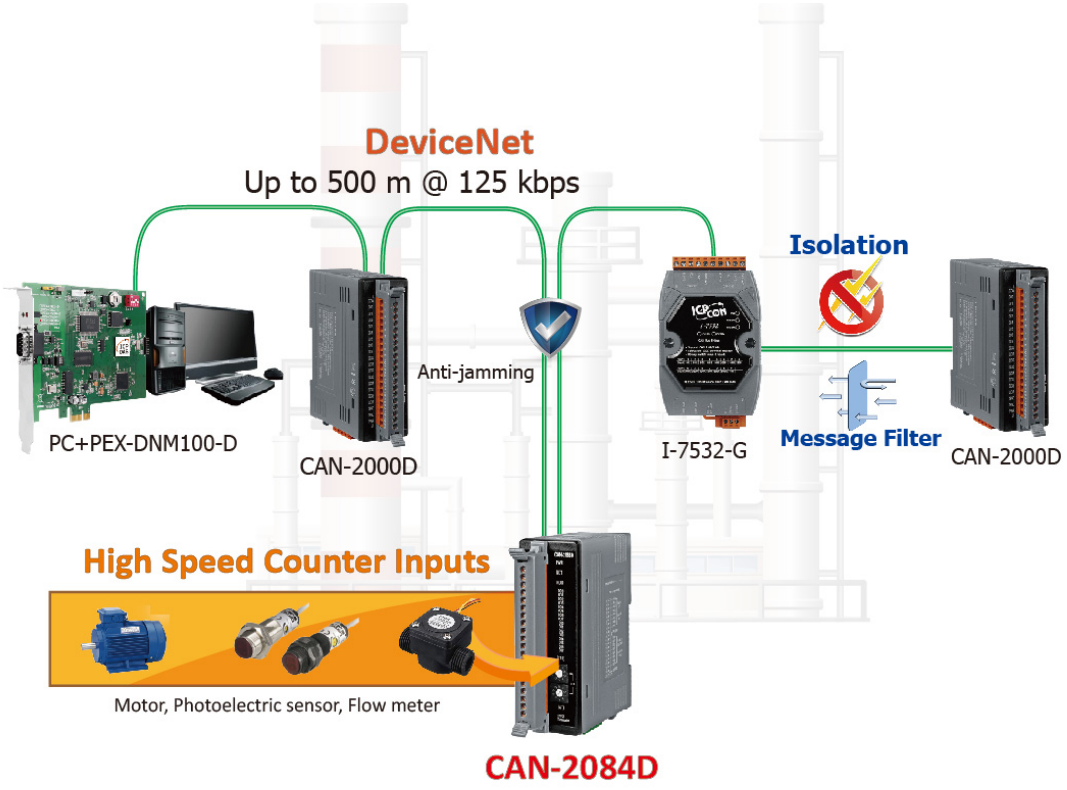
1.3 Hardware Specifications

CAN Interface	
Connector	5-pin screwed terminal block (CAN_GND, CAN_L, CAN_SHLD, CAN_H, CAN_V+)
Baud Rate (bps)	125 k, 250 k, 500 k (selected by rotary switch)
Terminator Resistor	DIP sw itch for the 120 Ω terminator resistor
Node ID	0~63(selected by rotary switch)
Digital Input	
Channel	4/8
Mode	4-channel Up/Dow n Counter (Up/Dow n) 4-channel Dir/Pulse Counter (Bi-direction) 4-channel Quadrant Counting 8-channel Up Counter 8-channel Frequency Programmable Digital Noise Filter : 1 to 32767 μs
Isolated Input Level	Logic Level 0: +1 V Max. Logic Level 1: +4.5 to +30 V

TTL Input Level	Logic Level 0: 0 to +0.8 V Logic Level 1: 2 to +5 V
Minimum Pulse Width	2µs 
Input Frequency	**1 Hz ~ (typically) 250 kHz** for both counter and frequency mode, w here 250 kHz is calculated as follow s: supposed that the duty cycle = 50%, by referring to the Minimum Pulse Duration of the High Level, the pulse period w ill be 2 µs x 2 = 4 µs, w hich is 250 kHz as a maximum. **Maximum Frequency:** Refer to the Minimum Pulse Duration of the High Level, the maximum frequency is highly affected by the duty cycle **Frequency Accuracy** = ±0.4%
Isolated Voltage	1000Vrms
Hardware	
ESD Protection	4 kV (Contact for each Channel)
LED	
DeviceNet Status LED	3 LEDs (PWR LED, NET LED，MOD LED)
Terminal Resister	1 LED to terminal resister indicator
Digital Input	8 LEDs as Digital Input Indicators
Power	
Power Supply	+10 ~ +30 VDC
Power Consumption	1.5W
Mechanism	
Installation Method	DIN-Rail
Dimensions	33 mm x 99 mm x 78 mm (W x L x H)
Environment	
Operating Temp.	-25 ~ 75 ℃
Storage Temp.	-30 ~ 80 ℃
Humidity	10 ~ 90% RH, non-condensing

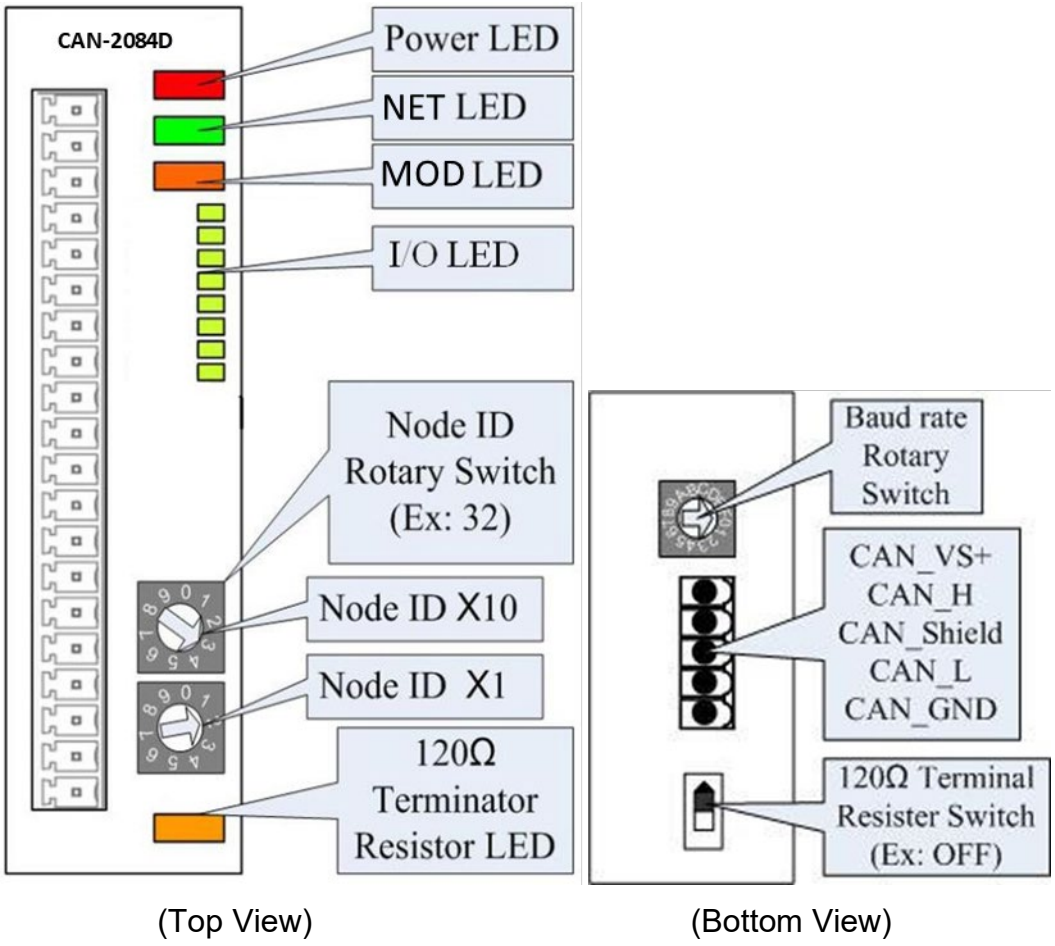
1.4 Application

- Counting events or totalizing
- Detecting frequency
- Monitoring position or speed with quadrature encoder



2. Hardware

2.1 Structure

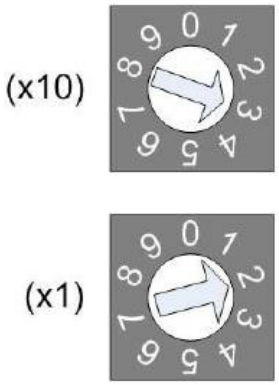


(Top View) (Bottom View)

2.2 Node ID & Baud Rate Rotary Switch

The two rotary switches below are used to adjust the node ID of the CAN-2084D module. The upper switch represents the tens digit of the node ID, while the lower switch represents the units digit of the node ID.

In the following example, the node ID of this module is 32.



Node ID rotary switch

The bottom rotary switch (BAUD) is used to adjust the baud rate of the CAN-2084D module. The relationship between the value indicated by the baud rate switch and the actual baud rate is shown in the table below：

Baud rate rotary switch

Rotary Switch Value	Baud Rate (K bps)
0	125
1	250
2	500

Baud rate and rotary switch

2.3 LED Description

POWER LED

The CAN-2084D needs a 10V~30VDC power supply. Under a normal connection, a good power supply and a correct voltage selection, as the unit it 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.

Terminal Resistor Indicator

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

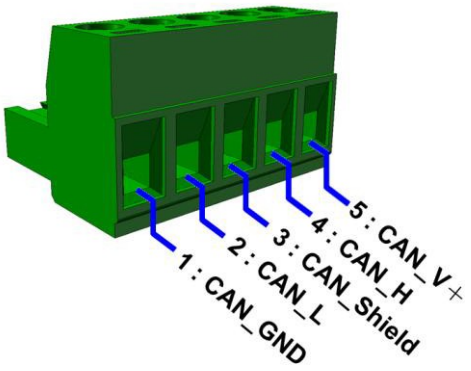
I/O LED

When an input is detected, the I/O indicator lights up; otherwise, it turns off.

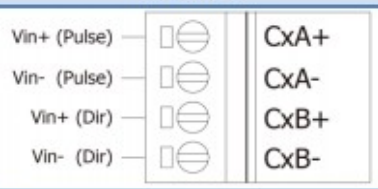
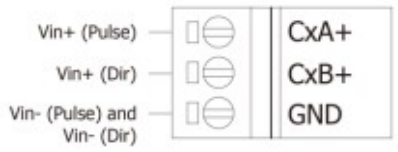
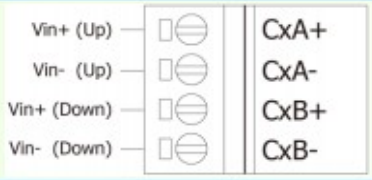
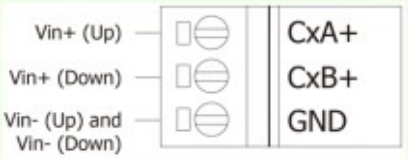
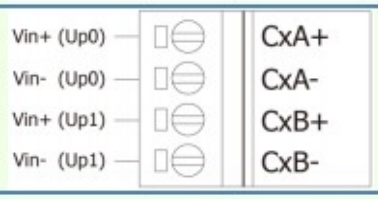
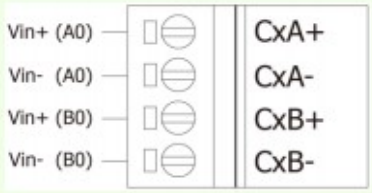
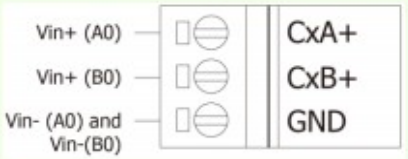
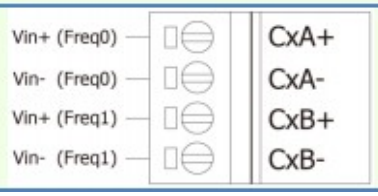
2.4 PIN Assignment

Terminal No	Pin Assignment
01	C0A+
02	C0A-
03	C0B+
04	C0B-
05	C1A+
06	C1A-
07	C1B+
08	C1B-
09	C2A+
10	C2A-
11	C2B+
12	C2B-
13	C3A+
14	C3A-
15	C3B+
16	C3B-
17	GND
18	GND
19	N.C
20	N.C

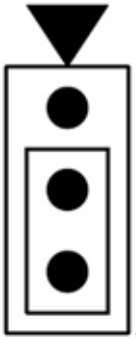
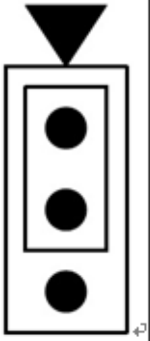
5針螺絲端子



2.5 Wire Connection

Input Mode	Isolated	Non-isolated
Dir/Pulse		
Up/Down		
Up		
A/B Phase (Quadrant)		
Frequency		

The CAN-2084D has two kind of inputs, isolated and non-isolated (TTL) for different input signals. Users can switch jumper setting on the CAN-2084D board for appropriate signal. These jumpers are located within JP1~JP8. The jumper settings are listed in the following table. The isolated input is set by default.

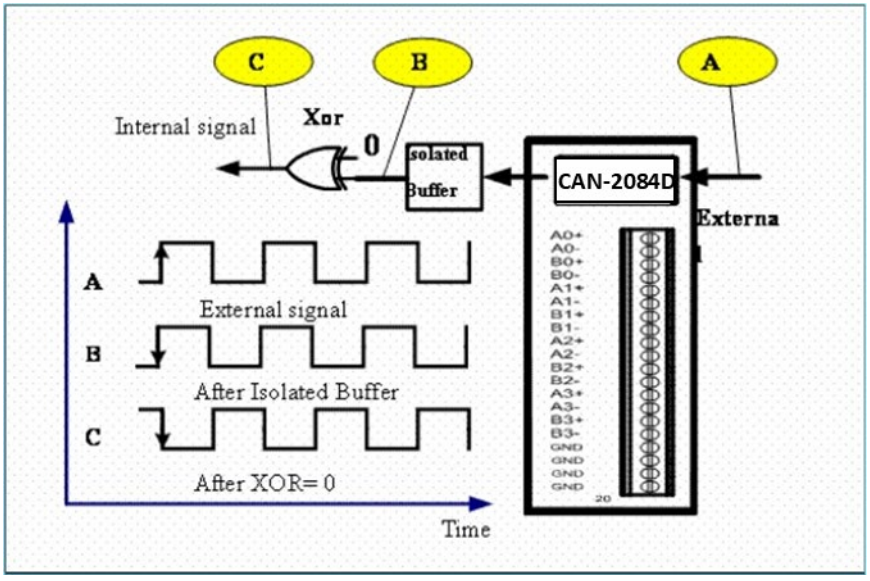
Jumper	Counter	Jumper setting			
JP1	A0	Isolated input (Default)		Non-isolated input (TTL input)	
JP2	B0				
JP3	A1				
JP4	B1				
JP5	A2				
JP6	B2				
JP7	A3				
JP8	B3				

2.6 Input Signal Model

Isolated Input (XOR=0)

The operational logic applied on CAN-2084D modules is a falling edge trigger. (Normal High and Active Low) The external signal is input into a CAN-2084D module through the isolation mechanism, with the signal being reversed from the external signal. This internal signal is the suggested waveform, as the XOR operation (XOR=0) doesn't need to be executed.

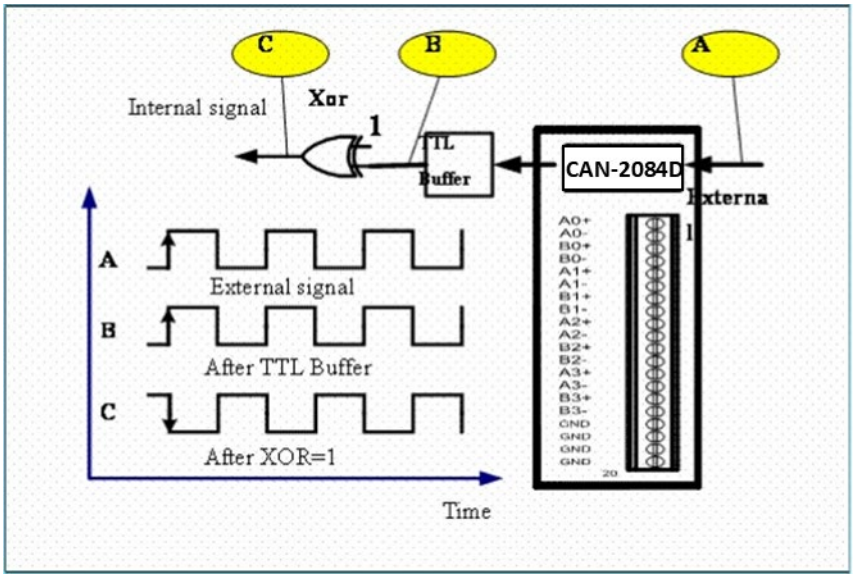
The solution is shown below：



TTL Input (XOR=1)

When an external TTL signal is input into a CAN-2084C module through the TTL mechanism, the signal will be the same as the external signal. This internal signal isn't the recommended waveform as the exclusive OR (XOR=1) operation must be executed to invert the waveform.

The solution is shown below：



2.7 Digital Low Pass Filter

The CAN-2084D includes three independent 2nd-order digital noise filters that can be used to remove noise, and are implemented as follows:

Channel	Low Pass Filter
0 (A0)	Low Pass Filter 0
1 (B0)	
2 (A1)	Low Pass Filter 1
3 (B1)	
4 (A2)	Low Pass Filter 2
5 (B2)	
6 (A3)	
7 (B3)	

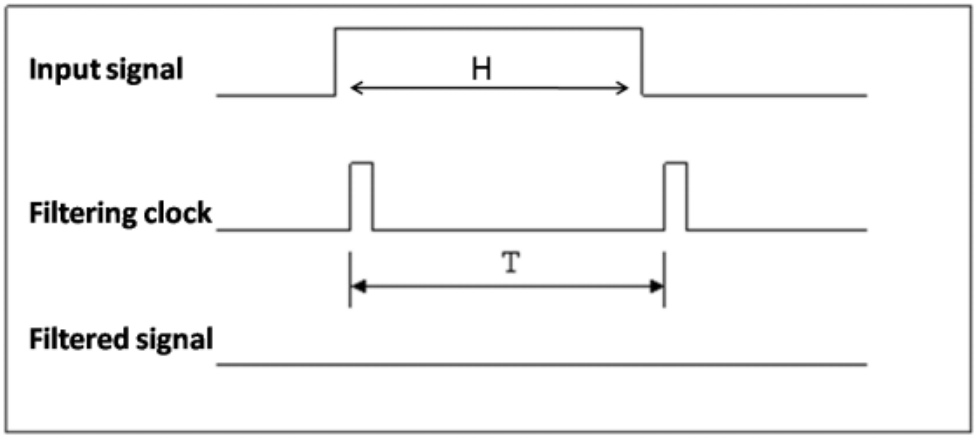
- The Low Pass Filter can be set to either enabled or disabled. The width of the Low Pass Filter is programmable and can be set within a range from 1 μs to 32767 μs.
- The Low Pass Filter can be applied to all working modes, both counter and frequency.
- All three Low Pass Filters are disabled by default when the module is shipped.

Refer to the following table for details of how to set the Low Pass Filter：

H = the **HIGH** width of the input signal T = the period of the filtering clock	
Case	**Filtering status**
H > 2T	The signal will be **PASSED**
T □ H □ 2T	The signal may be **FILTERED** or **PASSED**
H < T	The signal will be **FILTERED**

An example is illustrated in the following figure：

n this example, H < T and the high width of the input signal < the period of the filtering clock, and so the signal will be filtered.



Supposed T = 1 ms, that is the filtering clock has a frequency of 1 kHz. Now, if the duty cycle of the input signal is 50%, that is the high width is equal to the low width, then:

H < T ➞ H < 1 ms

➞ input signal period < 2 ms (duty cycle = 50%)

➞ input frequency > 500 Hz,

Consequently, the input signal will be filtered.

Now, if the duty cycle of the input signal is 25%, that is the low width is three times the high width, then:

H < T ➞ H < 1 ms

➞ Input signal period < 4 ms (duty cycle = 25%)

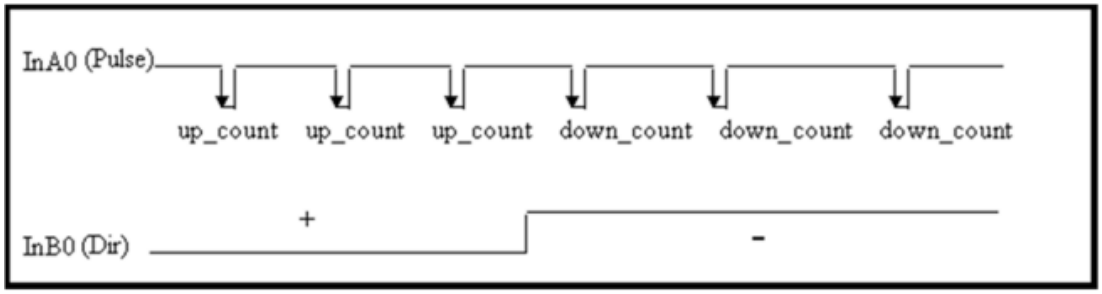
➞ Input frequency > 250 Hz,

Consequently, the input signal will be filtered.

Similarly, the maximum period of the filtering clock can be calculated to allow the input signal to be passed using the formula H > 2T.

2.8 Counter Mode

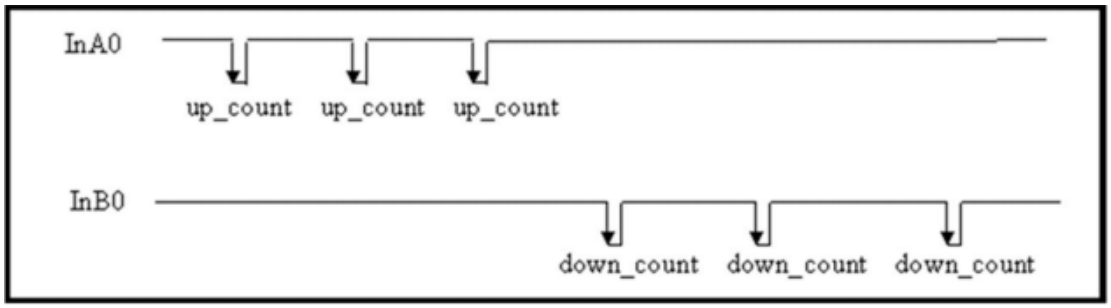
2.8.1 Mode 0：Dir/Pulse Counting(Bi-direction)



When InB0 is used as Dir：

- If InB0 is Low, counter will be increased by one for every falling edge of InA0
- If InB0 is High, counter will be decreased by one for every falling edge of InA0.

2.8.2 Mode 1：Up/Down Counting(Bi-direction)

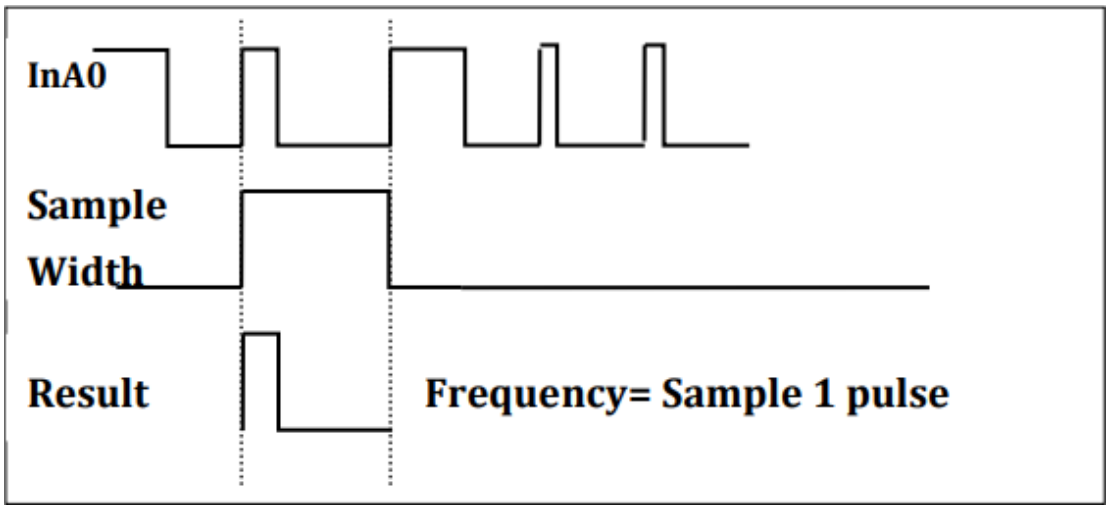


When InA0 is used as an UP_clock and InB0 is used as a DOWN_clock, counter_0 will be increased by one for every falling edge of InA0 and decreased by one for every falling edge of InB0.

2.8.3 Mode 2：Frequency Mode

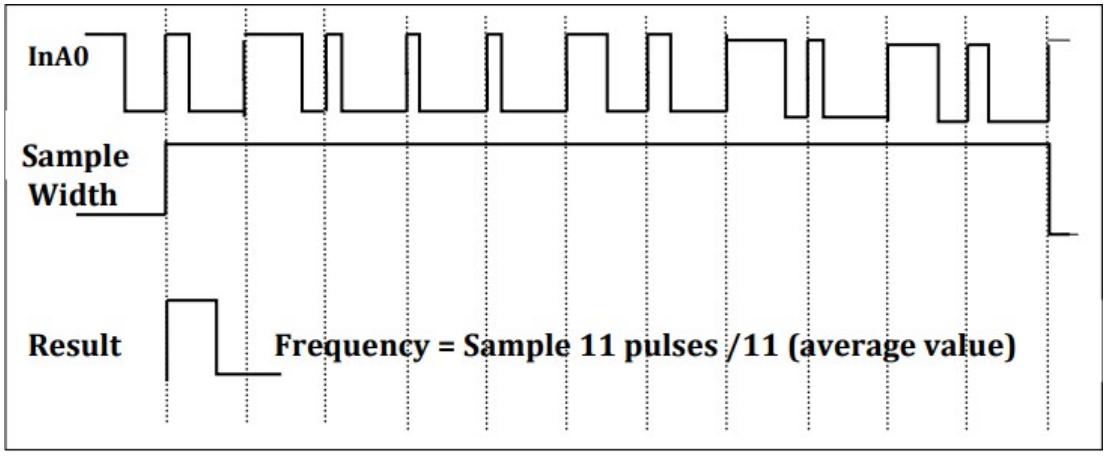
We calculate frequency of a certain channel by how many counts in a period of time. There are two modes: Normal Mode and High Speed Mode for measurement frequency

Normal Mode：



Normal Mode will read 1 pulse, this pulse will be calculated frequency value.

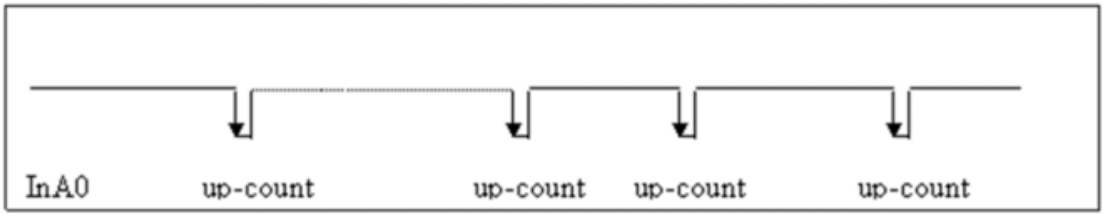
High Speed Mode：



High Speed Mode will read 11 pulses to calculate average value of those 11 pluses.

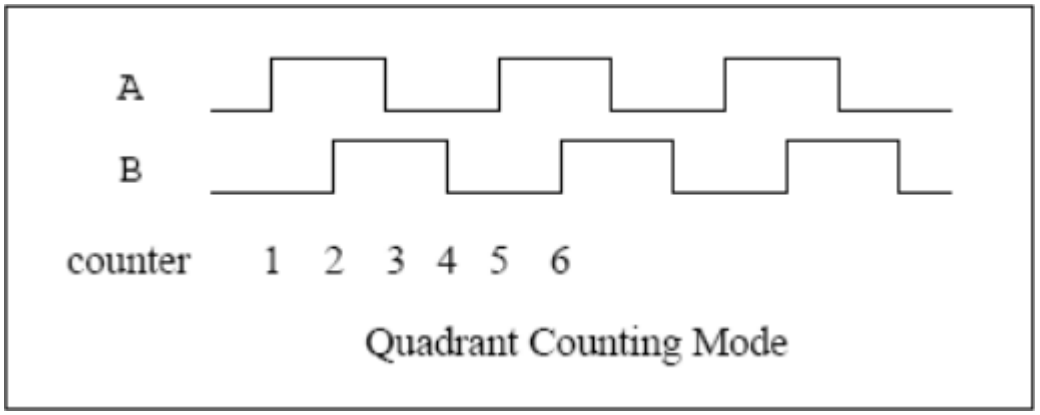
High Speed Mode will be more accuracy than Normal mode for measurement Frequency. We suggest using High Speed Mode if measurement frequency is more than 10k Hz.

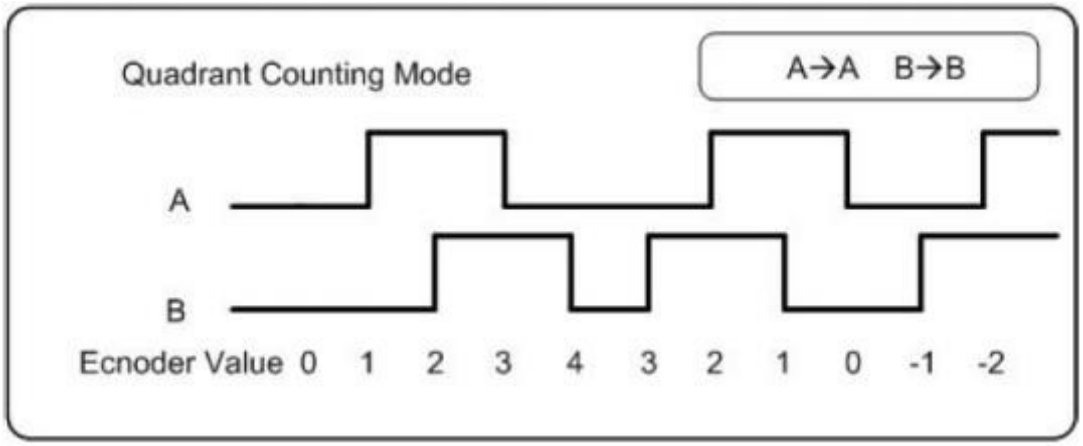
2.8.4 Mode 3：Up Counting



Counter_0 will be incremented by one for every falling edge of InA0.

2.8.5 Mode 4：Quadrant Counting(A/B Phase)





When InA0 is used as an A signal and InB0 is used as a B signal:

- Counter_0 will be increased by one when the InA0 phase leads by 90 degrees to InB0.
- Counter_0 will be decreased by one when the InA0 phase lags by 90 degrees to InB0.

3. DeviceNet Profile Area

This section describes the detailed functions for each object class that is implemented in the CAN-2084D 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-2084D
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	0x600
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-2084D
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
Strobe Produced	0x64	0x01	0x02
Strobe Consumed	0x64	0x01	0x01

Connection Instance ID	Description
0x01	References the Explicit Messaging Connection into the Server
0x02	References the Poll I/O Connection
0x03	References the 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	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 counter values for channels 0–7.

Contents of Each Assembly Object Instance

Instance ID	Description	Type	Method	Default Value
0x64	Counter values for channels 0–7	UINT32	Get	0x00

Parameter description of Assembly Object Instance

Instance ID	Data Range	Parameter Description
0x64	0x00000000 ~ 0xFFFFFFFF Per Counter Channel	Counter values for channels 0–7

Instance Attribute (Instance ID=0x64)

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-2084D module.

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	Access the counter input configuration of channel 0.
0x02	Access the counter input configuration of channel 1.
0x03	Access the counter input configuration of channel 2.
0x04	Access the counter input configuration of channel 3.
0x05	Access the counter input configuration of channel 4.
0x06	Access the counter input configuration of channel 5.
0x07	Access the counter input configuration of channel 6.
0x08	Access the counter input configuration of channel 7.

Instance Attribute (Instance ID=0x01~0x08)

Attribute ID	Description	Method	Data Type	Default Value
0x01	Counter types for channels 0–7	unsigned char	Get/Set	0x03
0x02	Counter values for channels 0–7	UInt32	Get	0x0000 0000
0x03	Overflow counts for channels 0–7	UInt32	Get	0x0000 0000

0x04	Detection trigger modes for channels 0–7	unsigned char	Get/Set	0x00
0x05	Frequency modes for channels 0–7	unsigned char	Get/Set	0x00
0x06	Frequency mode timeout settings for channels 0–7	UInt16	Get/Set	0xBB8 (3000 ms)
0x07	Digital low-pass filter settings for channels 0–7	unsigned char	Get/Set	0x00
0x08	Low-pass filter time constants for channels 0–7	UInt16	Get/Set	0x01
0x09	Input signal types for channels 0–7	UInt16	Get/Set	0x01
0x0A	Clear counter values for channels 0–7	unsigned char	Set	0x00

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>0x00 ~ 0x04</td><td>
<table>
<tr><th>Value</th><th>Counter Mode</th><th>Counter Count</th></tr>
<tr><td>0x00</td><td>Dir/Pulse Counting Mode</td><td>4</td></tr>
<tr><td>0x01</td><td>Up/Down Counting Mode</td><td>4</td></tr>
<tr><td>0x02</td><td>Frequency Mode</td><td>8</td></tr>
<tr><td>0x03</td><td>Up Counting Mode (Default)</td><td>8</td></tr>
<tr><td>0x04</td><td>Quadrant Counting Mode</td><td>4</td></tr>
</table>
Notice:
When using Dir/Pulse, Up/Down, or A/B Phase modes, the channels are configured in pairs.
For example, if channel 0 is set to Up/Down Counting Mode, channels 0 and 1 will both operate in this mode, and the values read from the two channels will be</td></tr>
</table>

<table>
<tr><td></td><td></td><td>identical.</td></tr>
<tr><td>0x02</td><td>0x00000000 ~ 0xFFFFFFFF</td><td>Get the counter value of the corresponding channel</td></tr>
<tr><td>0x03</td><td>0x00000000 ~ 0xFFFFFFFF</td><td>Records the counter value when an overflow occurs on the channel.
When the counter (Data) exceeds the valid range or an overflow occurs, the related value will be recorded in this object.</td></tr>
<tr><td>0x04</td><td>0 ~ 1</td><td><table><tr><td>Value</td><td>Edge trigger type</td></tr><tr><td>0</td><td>Falling edge trigger（Default）</td></tr><tr><td>1</td><td>Rising edge trigger</td></tr></table></td></tr>
<tr><td>0x05</td><td>0 ~ 1</td><td>When the channel is in Frequency Mode, this setting determines whether the frequency measurement operates in Normal Mode or High-Speed Mode<table><tr><td>Value</td><td>Description</td></tr><tr><td>0</td><td>Normal mode uses one pulse period to calculate the frequency, providing a fast response but lower resolution.</td></tr><tr><td>1</td><td>High-speed mode averages 11 pulse periods (Sample 11 pulses / averaging) to calculate the frequency, providing higher accuracy. It is recommended when the input frequency is greater than 10 kHz.</td></tr></table></td></tr>
<tr><td>0x06</td><td>0x0000 ~ 0xFFFF</td><td>Sets the maximum waiting time (timeout) for frequency measurement. If no valid pulse is detected within this period, it is treated as a timeout.</td></tr>
</table>

<table>
<tr><td></td><td></td><td>Default：0x0BB8 (3000 ms)
Recommended setting: Avoid setting it too small, as the frequency measurement may be considered a timeout before it is completed.</td></tr>
<tr><td>0x07</td><td>0 ~ 1</td><td>Controls whether the digital low-pass filter of each channel is enabled to suppress high-frequency noise.
The CAN-2084D includes three independent second-order digital filters. Refer to Section 2.7.
Whether the signal is treated as “noise” also depends on the clock period configured in the low-pass filter period object (Attribute 0x08).
<table>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Off (Default)</td></tr>
<tr><td>1</td><td>On</td></tr>
</table></td></tr>
<tr><td>0x08</td><td>1 μs ~ 32767 μs
(Default: 1)</td><td>Sets the clock period of the low-pass filter to determine the minimum noise pulse width to be filtered.
It is used to suppress relay bounce, interference spikes, and similar conditions to prevent them from affecting counting and frequency measurement.
Note:
If the input high-level duration H is shorter than the filter clock period T (H < T), the pulse will be treated as noise and filtered out.
In practice, to avoid boundary misjudgment, it is recommended to design H > 2T to ensure that valid pulses intended for counting can</td></tr>
</table>

<table>
<tr><td></td><td></td><td>pass through.</td></tr>
<tr><td>0x09</td><td>0 ~ 1
(Default: 1)</td><td>Sets the input channel to use either Isolated Input or TTL Input mode, together with the physical jumpers JP1–JP8 (refer to Section 2.5).<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>TTL Input
(Non-isolated)</td></tr><tr><td>1</td><td>Isolated Input
(Isolated, default)</td></tr></table></td></tr>
<tr><td>0x0A</td><td>Any value</td><td>Clear the counter value of the corresponding channel.</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

3.7 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:

Set the counter mode of Channel 1 to Dir/Pulse Counting Mode.
(Class ID:0x64，Instance ID:0x01，Attribute ID:0x01)

Assume the station number of CAN-2084D is 0x01, and the master (ID: 0x00) has already fully established an Explicit connection with the device. By accessing the object's **Attribute ID (0x01)** to write the value, the user can set the counter mode of Channel 1 to **Dir/Pulse Counting Mode**.
A slave response of 0x90 indicates that the setting was successful.

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	01	01	--	--

Master → **Slave (CAN-2084D)**

CAN ID											RTR	LEN	8 個位元組 資料 (十六進制)							
		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-2084D)**

Example 2:

Read the counter value of Channel 1.
(Class ID: 0x64, Instance ID: 0x01, Attribute ID: 0x02)

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	02	--	--	--

Master → **Slave (CAN-2084D)**

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	00	00	00	00	--	--

Master ← **Slave (CAN-2084D)**

The slave responds with **0x00 0x00 0x00 0x00**, indicating that the current counter value of **Channel 1** is **0**.