

6-axis Force/Torque Sensors

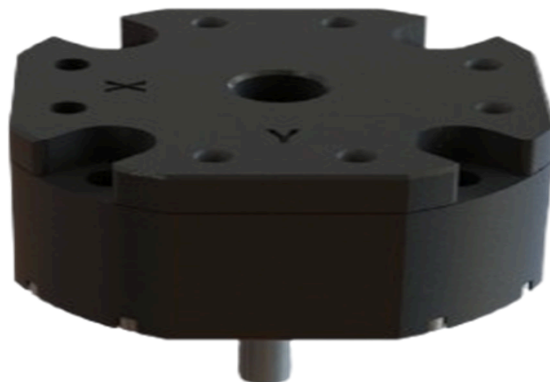
Smart 6-axis Force Torque Sensor, AFT150-D50-C

Rev. 2026.09.15

AIDIN
ROBOTICS

Force/Torque sensors user manual

This document is a user manual for AIDIN ROBOTICS
Force/Torque sensors



Foreword

Thank you for purchasing our Force Torque sensor. This manual contains the information necessary for the correct use of the AIDIN ROBOTICS AFT150-D50 sensor. Please carefully read this manual when using this software. Please note that the basic performance of the product will not be exhibited when the robot system is used out of conditions. This manual describes possible dangers and consequences using Force Torque sensor. Be sure to comply with safety precautions written in this manual to use safety and correctly.

Notice

This manual Do **NOT** allow copy, reproduction or share without authorization of AIDIN ROBOTICS. Please notify us any errors in this manual or the provided instructions.

Manufacturer

AIDIN ROBOTICS,

Safety Precautions

Installation of Force Torque sensor **MUST** be performed by qualified personnel in accordance with national and local codes. Please carefully read this manual when using this software.

Warning & Caution



This symbol indicates that a danger of possible serious injury or death exists if the associated instructions are not followed properly.



This symbol indicates that a danger of possible harm to people or physical damage to equipment and facilities exists if the associated instructions are not followed properly.

1. Product Overview

1.1 Smart 6-axis Force Torque Sensor, AFT150-D50

1.2 Flyer

- [📎 \[AIDIN ROBOTICS\] 6-axis FT Sensor AFT150-D50 leaflet_EN.pdf](#)

1.3 Key Features

- Smart 6-axis force/torque sensor
- All-in-one sensor (No additional amplifier)
- Digital output communications (CAN, CAN-FD)
- Easy installation and data collection
- Grippers, robot hands, collaborative robot, industrial robot

1.4 Specifications

Index	Unit	Value
Operating voltage	VDC	5
Max. safe excitation voltage	VDC	12
Nominal force range (F_XYZN)	N	150
Nominal torque range (M_XYZN)	Nm	7.5
Limit force (F_XYZL)	N	300
Limit torque (M_XYZL)	Nm	10

Index	Unit	Value
Dimensions	mm	D50 x H17.7
Weight	g	60
Temperature	°C	10-50
Sample rate	Hz	1000
Interfaces	1,000Kbps	CAN, CAN-FD
Wire	Length: 15 cm	For User: CAN_H / CAN_L / VCC / GND

1.5 Complex Loading Range

The sensor can operate up to its nominal capacity, while a single axis load. On Nominal capacity reading is incorrect and invalid. The nominal capacity of composite loading is The following diagram illustrates a complex loading scenario. The sensor cannot operate outside the normal operating area The following graph shows the range of payloads and tool lengths that can be used with AFT150-D50-C sensor for applications requiring high or moderate precision. A • The total percentage of the calibrated range used by Fxy and Tz axes is greater than 105%. Refer to the following Fxy, Tz equation.

$$\sqrt{F_x^2 + F_y^2} / F_{xy} \text{Range} + |T_z| / T_z \text{Range} > 105\%$$

• The total percentage of the calibrated range used by Fz and Txy axes is greater than 105%. Refer to the following Fz, Txy equation.

$$\sqrt{T_x^2 + T_y^2} / T_{xy} \text{Range} + |F_z| / F_z \text{Range} > 105\%$$

2. Installation Guide

2.1 Basic Components

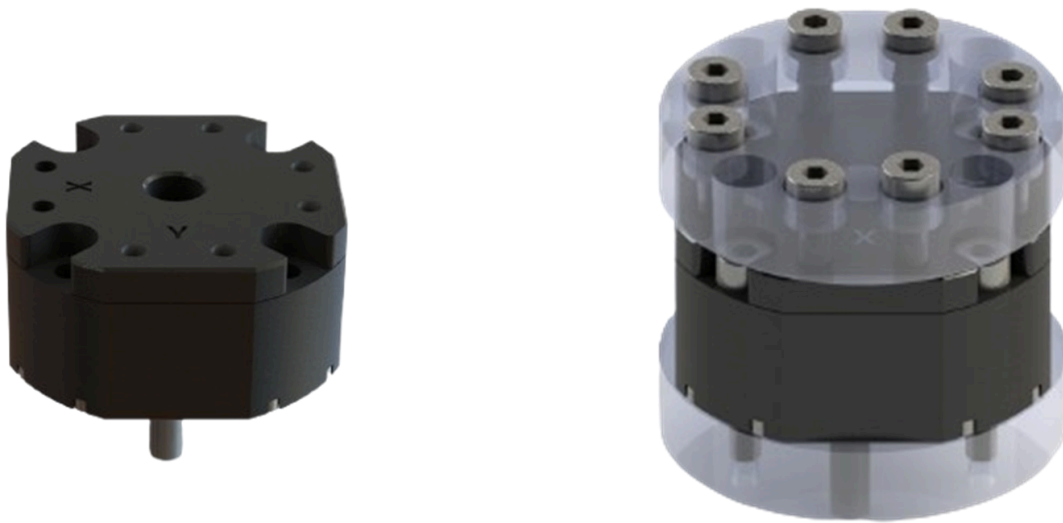
- AFT150-D50 x 1 EA

2.2 Mounting

⚠ M3 bolt 1.5Nm required for tightening (Need Locatiting)

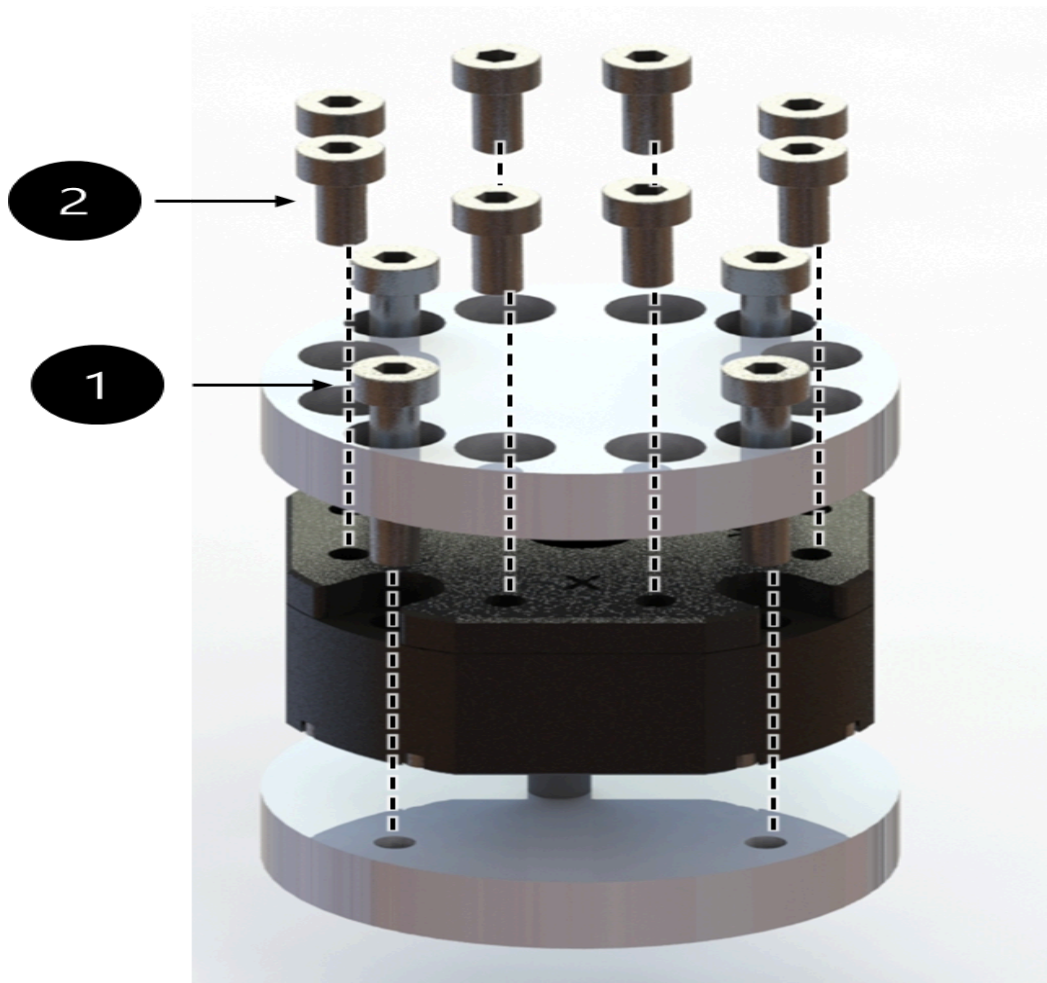
M4 bolt 3.6Nm required for tightening There are two types of bolts used.

- The first type is M4 bolt, which fixes the sensor to the bottom part, requires a length of at least 12.5 mm.
- The second type is M3 bolt, which is used for the top part that contacts the upper plate of the sensor, must not protrude more than 3 mm into the sensor. Please take caution when assembling.
- **Performance cannot be guaranteed and A/S not possible when disassembling internal/external bolts**



* Transparent parts are user part example

- Sensor tightening order



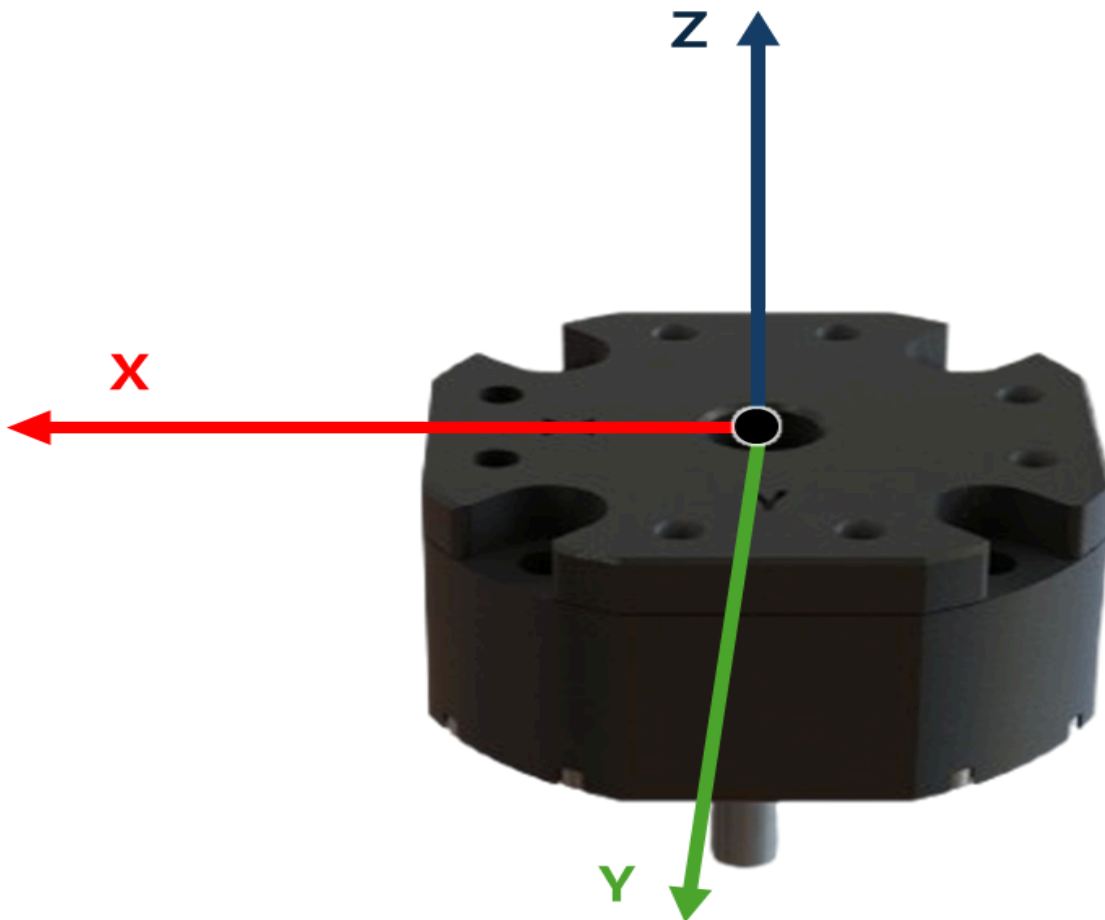
- CAUTION OF CABLE CUTTING AND EXCESS PULLING
- Secure the sensor line with the robot so that it does not pull as it moves
 - Do not fix it to the robot using cable ties, etc., or use cable ties to fix it in a bundle shape like other wires

- We recommend that you use Velcro to fix it with the robot, and please use Velcro to fix it when fixing it in a bundle form with other lines

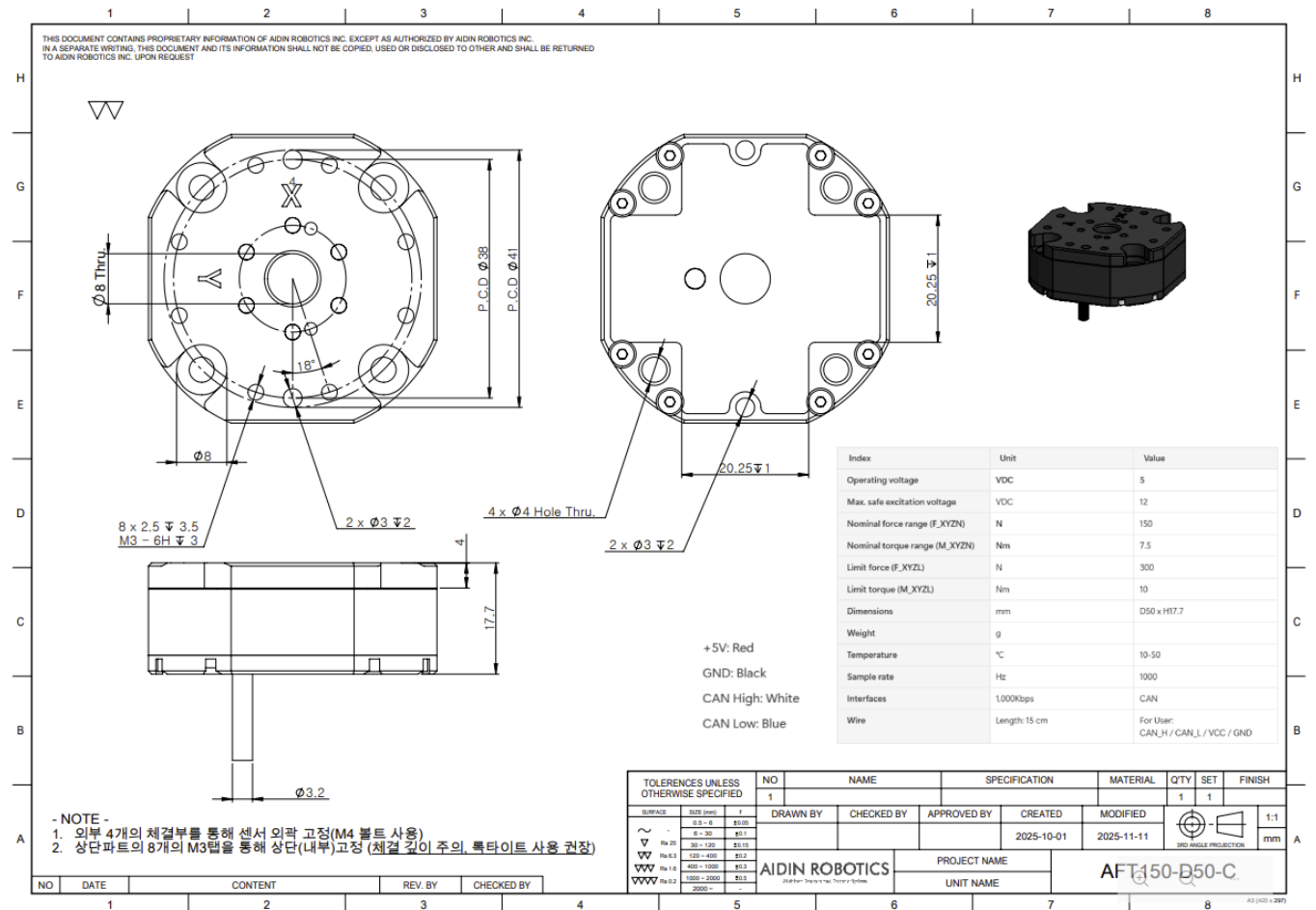
💡 Performance CANNOT be guaranteed if not followed to the tightening torque guide.

💡 If the sensor line is not fixed to the robot, the output signal may have a noise due to the pulled by robot

2.3 Axes and Drawings



Drawing File :2.3 Wiring



- AFT150-D50-C_배포최종본.PDF
- AFT150-D50-C_STEP 배포최종본.zip

2.4 Cable

- Length: 15cm



GHR-04V-S(JST社)

Pin No.	Feature	
1	+5V	RED
2	GND	BLACK
3	CAN H	WHITE
4	CAN L	BLUE

<Pin Map>

💡 To acquire CAN data, a 120 Ω termination resistor is required between CAN H and CAN L.

3. Communication

3.1 Default CAN Setting

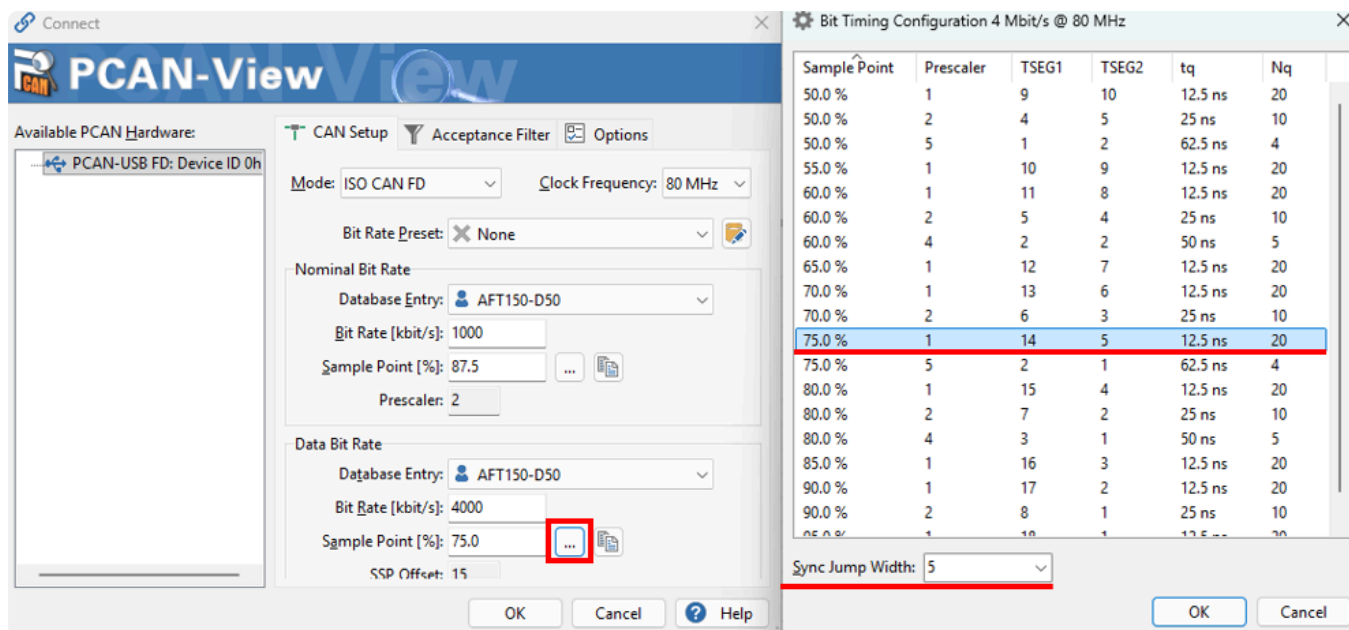
- The output rate can be changed from 100 Hz to 1000 Hz
- Please refer to the “3.2 User Commands ” below for how to change it
- CAN-2.0
 - Nominal bitrate : 1Mbps
 - Data bitrate : 1Mbps
 - RX ID MASK : 0x220~0x22F, 0x320~0x32F
- CAN-FD
 - Nominal bitrate : 1Mbps
 - Data bitrate : 4Mbps (BRS mode)
 - RX ID MASK : 0x220~0x22F, 0x320~0x32F
- CAN-FD parameter setting example using **PCAN device** (refer 3. Additional information)

The screenshot shows the PCAN-View software interface. The main window is titled 'PCAN-View' and has tabs for 'CAN Setup', 'Acceptance Filter', and 'Options'. The 'CAN Setup' tab is active, showing settings for 'Mode: ISO CAN FD', 'Clock Frequency: 80 MHz', 'Bit Rate Preset: None', 'Nominal Bit Rate' (Database Entry: AFT150-D50, Bit Rate [kbit/s]: 1000, Sample Point [%]: 87.5, Prescaler: 2), and 'Data Bit Rate' (Database Entry: AFT150-D50, Bit Rate [kbit/s]: 4000, Sample Point [%]: 75.0, SSP Offset: 15). A red box highlights the 'Sample Point [%]' field in the Nominal Bit Rate section.

Overlaid on the right is the 'Bit Timing Configuration 1 Mbit/s @ 80 MHz' window. It contains a table with the following data:

Sample Point	Prescaler	TSEG1	TSEG2	tq	Nq
78.8 %	1	62	17	12.5 ns	80
80.0 %	1	63	16	12.5 ns	80
80.0 %	2	31	8	25 ns	40
80.0 %	4	15	4	50 ns	20
80.0 %	8	7	2	100 ns	10
80.0 %	16	3	1	200 ns	5
81.3 %	1	64	15	12.5 ns	80
81.3 %	5	12	3	62.5 ns	16
82.5 %	1	65	14	12.5 ns	80
82.5 %	2	32	7	25 ns	40
83.8 %	1	66	13	12.5 ns	80
85.0 %	1	67	12	12.5 ns	80
85.0 %	2	33	6	25 ns	40
85.0 %	4	16	3	50 ns	20
86.3 %	1	68	11	12.5 ns	80
87.5 %	1	69	10	12.5 ns	80
87.5 %	2	34	5	25 ns	40
87.5 %	5	13	2	62.5 ns	16
87.5 %	10	6	1	125 ns	8

The row for 87.5 % with Prescaler 2 is highlighted in blue. Below the table, the 'Sync Jump Width' is set to 4. The 'OK' and 'Cancel' buttons are at the bottom right.



3.2 User Commands

- RX CAN IDs : 0x220 (CAN 2.0) , 0x320 (CAN FD) TX CAN IDs : The pre-configured TX CAN IDs are 0x230 (CAN 2.0) and 0x330 (CAN FD). Accordingly, the 'current TX CAN ID' in the table below is represented as [0x30, 0x02] or [0x30, 0x03] (LSB first).
- Data Transmission Procedure The sensor does not automatically output data upon power-up. To receive data, the host must send the **CAN MODE** configuration command followed by the **TRANSMIT DATA** request command.
Example: **CAN MODE:** ID: 0x220, Data: 0x30 0x02 0x04 0x01 **TRANSMIT DATA:** ID: 0x220, Data: 0x30 0x02 0x03 0x02

COMMAND	RX CAN ID	Data[0]	Data[1]	Data[2]	Data[3] See table below	Data[4]	Data[5]
SENSOR TX CAN ID SET	0x220	current	current	0x01	0x01	TX CAN 2.0	TX CAN 2.0
	0x320	TX CAN ID (LSB)	TX CAN ID (MSB)		0x02	TX CAN FD ID (LSB)	TX CAN FD ID (MSB)
BIAS				0x02			
TRANSMIT DATA				0x03	0x01		
					0x02		
					0x03		
					0x04		
					0x05		
					0x06		
CAN MODE				0x04	0x01		
					0x02		
					0x03		

COMMAND	RX CAN ID	Data[0]	Data[1]	Data[2]	Data[3] See table below	Data[4]	Data[5]
SAMPLE RATE SET				0x05	0x01 0x02 0x03 0x04		
FD PARAMETERSSET				0x06	0x01 0x02 0x03 0x04		
SERIAL NUMBER CONFIRM		0xFF	0xFE	0xFB			
TX CAN ID CONFIRM		0xFF	0xFE	0xFC	0x01 0x02		
FACTORY RESET		0xFF	0xFE	0xFD			

COMMAND	Data[3] DESCRIPTION
SENSOR TX CAN ID SET	0x01 : TX CAN 2.0 ID SET ex) ID(LSB) = 0x23, ID(MSB) = 0x01, Resulting ID 0x123 0x02 : TX CAN FD ID SET ex) ID(LSB) = 0x56, ID(MSB) = 0x04, Resulting ID 0x456
BIAS	Bias
TRANSMIT DATA	0x01 : INT without temperature compensation 0x02 : INT with temperature compensation 0x03 : INT Combined without temperature compensation (Unsupported in CAN 2.0) 0x04 : INT Combined with temperature compensation (Unsupported in CAN 2.0) 0x05 : Float Combined without temperature compensation (Unsupported in CAN 2.0) 0x06 : Float Combined with temperature compensation (Unsupported in CAN 2.0)
CAN MODE	0x01 : CAN 2.0 mode 0x02 : CAN FD mode BRS OFF (Effective from 2026.06.19) 0x03 : CAN FD mode BRS ON (Effective from 2026.06.19)
SAMPLE RATE SET	0x01 : RATE 100Hz (Default) 0x02 : RATE 250Hz 0x03 : RATE 500Hz 0x04 : RATE 1000Hz

COMMAND	Data[3] DESCRIPTION
FD PARAMETERSET	0x01 : FDSET1 0x02 : FDSET2 0x03 : FDSET3 (Default) 0x04 : FDSET4
SERIAL NUMBER CONFIRM	(To Be Determined)
TX CAN ID CONFIRM	0x01 : TX CAN 2.0 ID : data field not used (DLC = 0) 0x02 : TX CAN FD ID : data field not used (DLC = 0)
FACTORY RESET	FDSET3(Default) RATE 100Hz (Default) Zero Bias (Default) current TX CAN2.0 ID : 0x230 (Default), current TX CANFD ID : 330 (Default)

3.3 Sensor Data Output

 **To stabilize the sensor signal, it is recommended to have a running time of about 30 minutes.**

Please leave the sensor data on for at least 10 minutes before using it The first 10 minutes of sensor output data can cause data to flow.

Force INT Transmitting mode (data[2] : 0x03, data[3] : 1 or 2)

INDEX	TX CAN ID	DLC	data[0]	data[1]	data[2]	data[3]	data[4]	data[5]
FINT	2.0 ID 0x230 FD ID 0x330	6	Fx(LSB)	Fx(MSB)	Fy(LSB)	Fy(MSB)	Fz(LSB)	Fz(MSB)

- $F_x \text{ Output} = F_x(\text{MSB}) * 256 + F_x(\text{LSB})$
- $F_y \text{ Output} = F_y(\text{MSB}) * 256 + F_y(\text{LSB})$
- $F_z \text{ Output} = F_z(\text{MSB}) * 256 + F_z(\text{LSB})$
- $\text{Force}[N] = \text{Force Output} / 100$

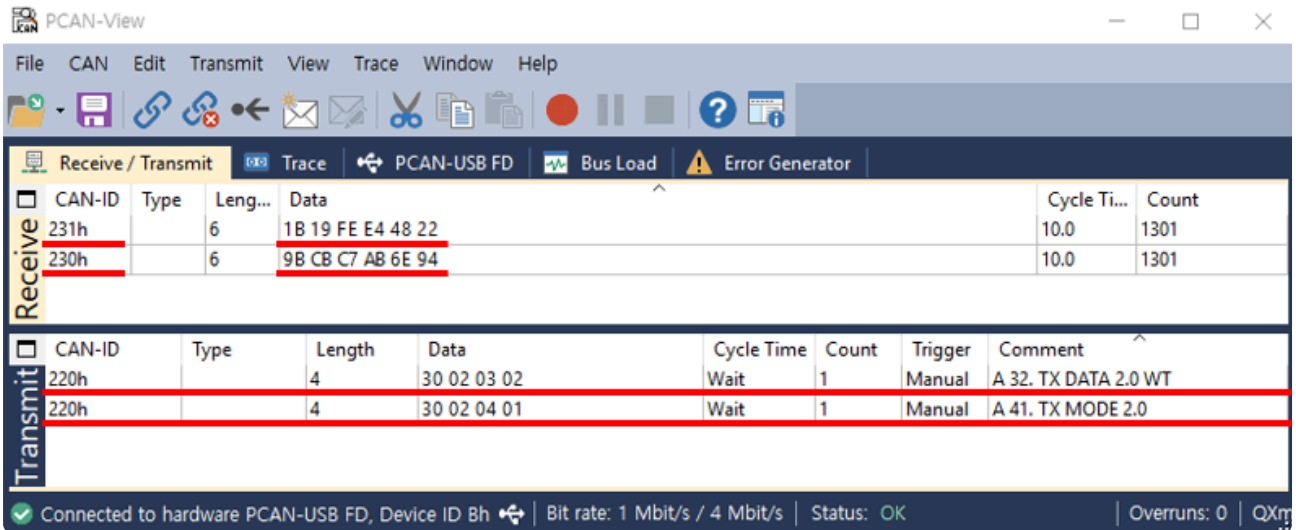
 **The final calculated Force and Torque values need to be cast into 16-bit integers.**

Torque INT Transmitting mode (data[2] : 0x03, data[3] : 1 or 2)

INDEX	TX CAN ID	DLC	data[0]	data[1]	data[2]	data[3]	data[4]	data[5]
TINT	2.0 ID 0x231 FD ID 0x331	6	Tx(LSB)	Tx(MSB)	Ty(LSB)	Ty(MSB)	Tz(LSB)	Tz(MSB)

- Tx Output = Tx(MSB)*256 + Tx(LSB)
- Ty Output = Ty(MSB)*256 + Ty(LSB)
- Tz Output = Tz(MSB)*256 + Tz(LSB)
- Torque[Nm] = Torque Output/1000

 **The final calculated Force and Torque values need to be cast into 16-bit integers.**



Combined INT Transmitting mode (data[2] : 0x03, data[3] : 3 or 4)

INDEX	TX CAN ID	DLC	data[0]	data[1]	data[2]	data[3]	data[4]	data[5]
CINT	FD ID 0x332	12	Fx(LSB)	Fx(MSB)	Fy(LSB)	Fy(MSB)	Fz(LSB)	Fz(MSB)

INDEX	data[6]	data[7]	data[8]	data[9]	data[10]	data[11]
CINT	Tx(LSB)	Tx(MSB)	Ty(LSB)	Ty(MSB)	Tz(LSB)	Tz(MSB)

- Fx Output = Fx(MSB)*256 + Fx(LSB)
- Fy Output = Fy(MSB)*256 + Fy(LSB)
- Fz Output = Fz(MSB)*256 + Fz(LSB)
- Tx Output = Tx(MSB)*256 + Tx(LSB)
- Ty Output = Ty(MSB)*256 + Ty(LSB)
- Tz Output = Tz(MSB)*256 + Tz(LSB)
- Force[N] = Force Output/100
- Torque[Nm] = Torque Output/1000

 **The final calculated Force and Torque values need to be cast into 16-bit integers**

The top screenshot shows a received message (Rx) with CAN-ID 332h, Type FD, Length 12, and Data 01 E1 A4 A3 80 03 6E 3E E7 E8 4C FE. The bottom screenshot shows transmitted messages (Tx) with CAN-ID 320h, Type FD, and various data fields. Both screenshots show a status bar at the bottom indicating 'Connected to hardware PCAN-USB FD, Device ID 0h' and 'Bit rate: 1 Mbit/s / 4 Mbit/s'.

Combined Float Transmitting mode (data[2] : 0x03, data[3] : 5 or 6)

INDEX	TX CAN ID	DLC	data[0~3]	data[4~7]	data[8~11]	data[12~15]	data[16~19]	data[20~23]
-------	-----------------	-----	-----------	-----------	------------	-------------	-------------	-------------

CFLOAT	FD ID 0x332	24	Fx	Fy	Fz	Tx	Ty	Tz
--------	----------------	----	----	----	----	----	----	----

```
uint32_t Fx_Raw = (Fx(MSB) << 24) | (Fx(3rd) << 16) | (Fx(2nd) << 8) | (Fx(LSB));
float Fx_Output = *(float *)&Fx_Raw;
uint32_t Fy_Raw = (Fy(MSB) << 24) | (Fy(3rd) << 16) | (Fy(2nd) << 8) | (Fy(LSB));
float Fy_Output = *(float *)&Fy_Raw;
uint32_t Fz_Raw = (Fz(MSB) << 24) | (Fz(3rd) << 16) | (Fz(2nd) << 8) | (Fz(LSB));
float Fz_Output = *(float *)&Fz_Raw;
uint32_t Tx_Raw = (Tx(MSB) << 24) | (Tx(3rd) << 16) | (Tx(2nd) << 8) | (Tx(LSB));
float Tx_Output = *(float *)&Tx_Raw;
uint32_t Ty_Raw = (Ty(MSB) << 24) | (Ty(3rd) << 16) | (Ty(2nd) << 8) | (Ty(LSB));
float Ty_Output = *(float *)&Ty_Raw;
uint32_t Tz_Raw = (Tz(MSB) << 24) | (Tz(3rd) << 16) | (Tz(2nd) << 8) | (Tz(LSB));
float Tz_Output = *(float *)&Tz_Raw;
```

- Force[N] = Force Output
- Torque[Nm] = Torque Output

💡 The final calculated Force and Torque values need to be cast into float.

PCAN-View

File CAN Edit Transmit View Trace Window Help

Receive / Transmit Trace PCAN-USB FD Bus Load Error Generator

CAN-ID	Type	Length	Data	Cycle Time	Count
332h	FD	24	6A 4B 9E C2 4A 32 C7 47 FA 18 D5 CA A9 15 07 44 9F A8 BC C0 A0 31 E5 BE	10.0	679

Rx

CAN-ID	Type	Length	Data	Cycle Time	Count	Trigger	Comment
320h	FD BRS	4	30 03 03 04	Wait	5	Manual	B 34. TX DATA FD INT WT
320h	FD BRS	4	30 03 03 06	Wait	6	Manual	B 36. TX DATA FD FLO WT
320h	FD BRS	4	30 03 04 02	Wait	6	Manual	B 42. TX MODE FD BRS OFF
320h	FD BRS	4	30 03 04 03	Wait	8	Manual	B 43. TX MODE FD BRS ON

Transmit

Connected to hardware PCAN-USB FD, Device ID 0h Bit rate: 1 Mbit/s / 4 Mbit/s Status: OK Overruns: 0 QXmtF

PCAN-View

File CAN Edit Transmit View Trace Window Help

Receive / Transmit Trace PCAN-USB FD Bus Load Error Generator

CAN-ID	Type	Length	Data	Cycle Time	Count
332h	FD BRS	24	C6 AE 9D C2 F0 15 C7 47 CA B9 D4 CA 51 0E 07 44 0E 61 BC C0 A8 79 E4 BE	10.0	3016

Rx

CAN-ID	Type	Length	Data	Cycle Time	Count	Trigger	Comment
320h	FD BRS	4	30 03 03 04	Wait	5	Manual	B 34. TX DATA FD INT WT
320h	FD BRS	4	30 03 03 06	Wait	5	Manual	B 36. TX DATA FD FLO WT
320h	FD BRS	4	30 03 04 02	Wait	5	Manual	B 42. TX MODE FD BRS OFF
320h	FD BRS	4	30 03 04 03	Wait	8	Manual	B 43. TX MODE FD BRS ON

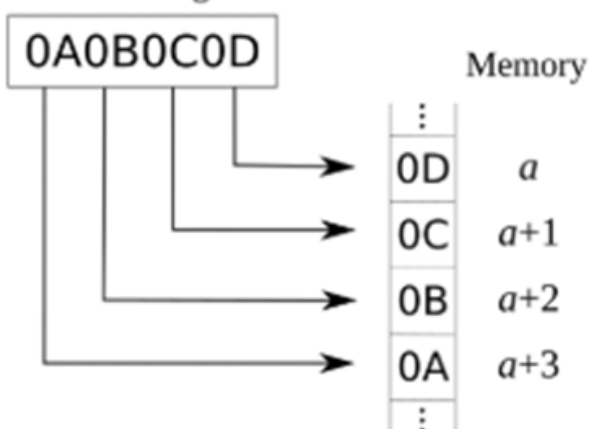
Transmit

Connected to hardware PCAN-USB FD, Device ID 0h Bit rate: 1 Mbit/s / 4 Mbit/s Status: OK Overruns: 0 QXmtF

💡 The calculation of the data must follow the little-endian format.

Little-endian

32-bit integer



4. Additional information

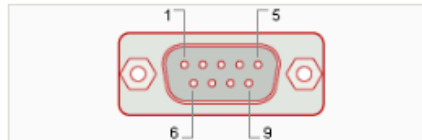
4.1 USB-to-CAN interfaces

PCAN-USB FD Device (USB to CAN FD)

- You may use a different CAN-enabled board, but use peak-system converter to use the sample program



Pin assignment D-Sub



- 1 not connected / optional +5 V
- 2 CAN-L
- 3 GND
- 6 GND
- 7 CAN-H

- Device Item number IPEH-004022
 - <https://www.peak-system.com/PCAN-USB-FD.365.0.html?&L=1>
- PCAN-View (Windows Software for Displaying)
 - <https://www.peak-system.com/PCAN-View.242.0.html?&L=1>

FAQ

(내용 준비 중)

Revision History

Date	Description
2026.09.15	Initial release (Current version)



Can't find what you're looking for?

Try our FAQs first. Still need help? Contact us and our team will get back to you.

[Browse FAQs](#)

[Contact Sales →](#)