ATO

V2.3

Angle sensor

TL725D-CAN2.0A/B

TECHNICAL MANUAL

○ Revision date: 2026-7-8

Note: Product functions, parameters, appearance, etc. will be adjusted as the technology upgrades, please contact our pre-sales business to confirm when purchasing.

Disclaimer

This product is developed exclusively for commercial applications and is prohibited from being used for illegal purposes, such as military activities, research related to nuclear, chemical, or biological weapons, or any other actions that violate laws and regulations. It is strictly forbidden to engage in activities related to goods embargoed by the United Nations, the European Union, or the OSCE, as well as any other activities prohibited by the Export Administration Regulations.

Without the written consent of our company, this product must not be transferred to any third party. Purchasing this product signifies your acceptance of this disclaimer and your agreement to sign the related liability agreement. If these regulations are violated, all responsibilities will be borne by the purchaser, and our company will not be held liable.

● PRODUCT INTRODUCTION

TL725D is a professional angle sensor designed for navigation applications in agricultural machinery automatic lifting, with built-in industrial grade three-axis angular rate and three-axis linear velocity. This product can still output smooth posture data even under the high vibration characteristics of agricultural machinery. A dynamic attitude algorithm based on MEMS inertial measurement platform is developed to output real-time data on the dynamic inclination angle, horizontal azimuth angle, three-axis angular rate, three-axis linear velocity, and forward axial body linear velocity of an object . This product is specifically designed for automated agricultural machinery, precise operations, and efficiency, and is an essential component for the new generation of automated precision agriculture control.

● FEATURES

★ Horizontal relative heading and attitude angle output
★ Real-time angular rate output
★ Long life and strong stability
★ Forward axis linear velocity
★ Compact and lightweight design
★ CAN2.0A/B output optional
★ All solid state
★ Light weight
★ DC9~36V power supply

●APPLICATION

★ Automated agricultural machinery
★ Precision agriculture
★ Combination Harvesters
★ Farming machine




• SPECIFICATIONS

TL725D-C2/C3	PARAMETERS
Three-axis attitude	relative heading ±180°, Roll angle ±180°, Pitch angle ±90°
Bandwidth	>100Hz
Nonlinear	0.1% of FS
Startup time	250ms
Input voltage	+9~36V
Current	35mA(12V)
Working temp	-40 ~ +80℃
Storage temp	-40 ~ +85℃
Vibration	5g~10g
Impact	200g pk, 2ms, ½sine
Working life	10 years
Output frequency	1Hz~200Hz Settable
Output signal	CAN 2.0A / CAN2.0B(Optional)
MTBF	≥98000 Hours/times
Insulation resistance	≥100 MΩ
Shockproof	100g@11ms, 3 Axial Direction (Half Sinusoid)
Anti-vibration	10grms/10~1000Hz
Protection grade	IP67
Cable	Standard 1 m 4 pin shield wire
Weight	≤160g (including 1 meter standard cable)

• ORDERING

TL725 □ □

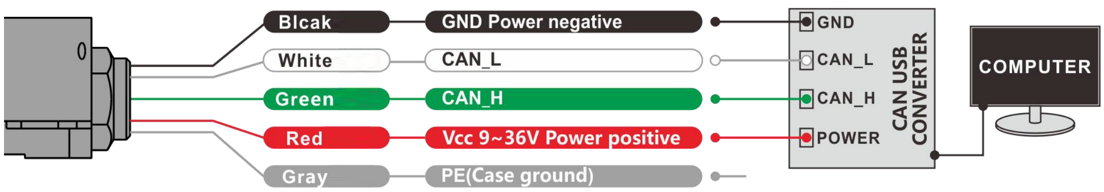
Output
- C2: CAN 2.0A
- C3: CAN 2.0B

Application
- AGM: Agricultural machinery
- V: Vehicle
- MTI: Mine Truck inclination

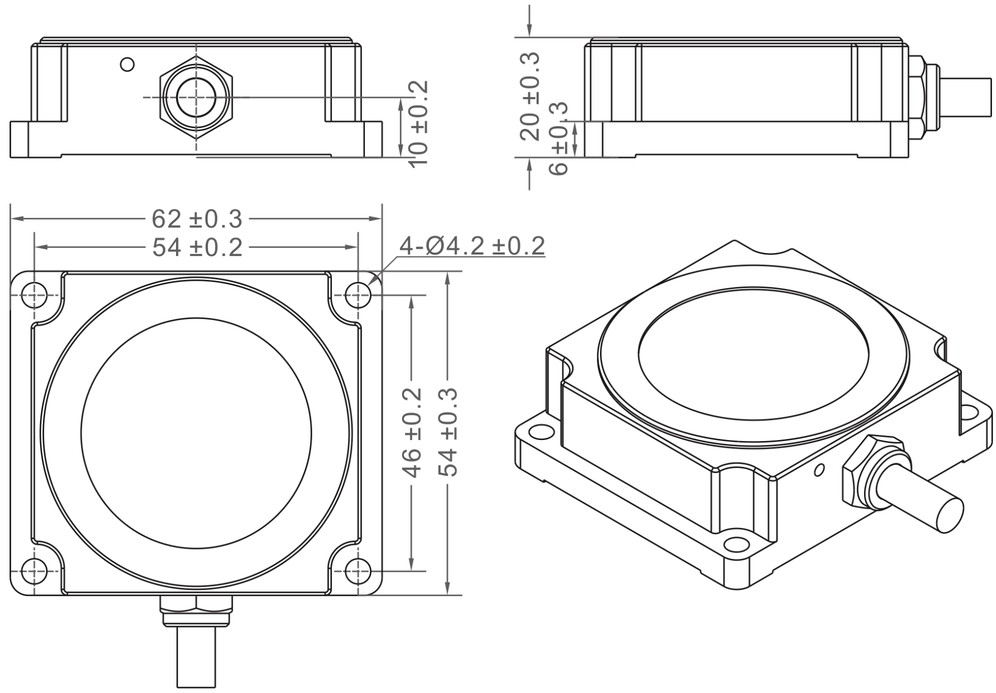
E.g: TL725D-C2-AGM : CAN2.0A communication protocol / Agricultural machinery application.

• ELECTRICAL CONNECTION

Color	BLACK	WHITE	GREEN	RED	GRAY
function	GND	CAN_L	CAN_H	Vcc 9~36V	PE



● SIZE



Shell size: L62×W54×H20mm

Installation size: L54×W46×H4mm

Suggest using screws: 4 M4 screws

● NOTE

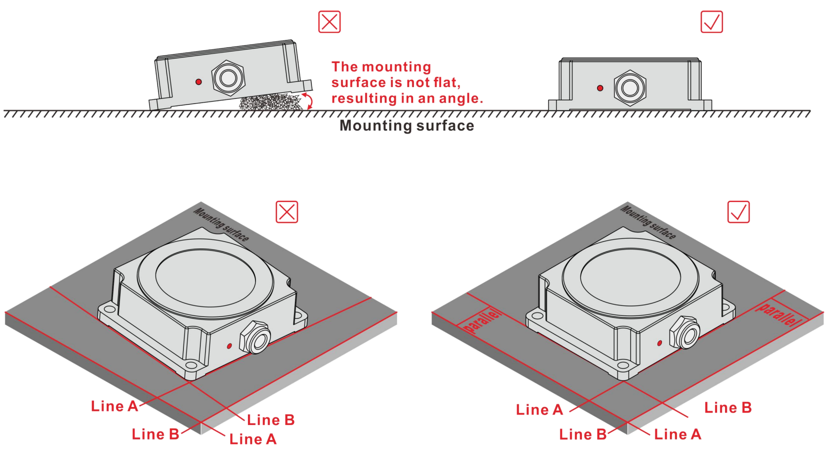
1. During installation, the installation surface of the angle meter should be kept parallel to the measured target surface, and the influence of dynamics and linear velocity on the angle meter should be reduced. Incorrect installation can lead to measurement errors, especially pay attention to one "surface" and two "lines":

1) The fixing between the installation surface and the measured surface must be tight, flat, and stable. If the installation surface is uneven, it can easily cause errors in the angle measurement of the angle meter.

2) The axis of the angle meter must be parallel to the measured axis, and there must be no angle between the two axes.

2. Do not shake the product violently during use to avoid severe vibrations. Keep away from the vibration source (if unavoidable, install shock absorbers) to avoid affecting the measurement accuracy of the product;

3. During use, try to avoid movements with angular rate greater than 300 °/s, such as rapid linear velocity, sudden stops, and sharp turns, to avoid affecting the measurement accuracy of the product.



● COMMUNICATION FRAME FORMAT

CAN2.0 protocol supports 2.0A (11-bit ID) or 2.0B (29-bit ID).

1) Modify the node number, (node range:0x01-0x7F), the default node number is 0x05.

Request message format.

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x10	0x10	0x00	Node_ID	0x00	0x00	0x00

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x10	0x10	0x00	Node_ID	0x00	0x00	0x00

Note: If the controller sends CAN-ID=0x600+0x05, send data: 40 10 10 00 10 00 00 00 The sensor returns CAN-ID=0x580+0x05, return data: 40 10 10 00 10 00 00 00 after power on again, the frame ID received is 0x590 (0x580+0x10), indicating that the frame ID has been modified successfully.

2)Set CAN baud rate

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x20	0x10	0x00	Baud	0x00	0x00	0x00

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x20	0x10	0x00	Baud	0x00	0x00	0x00

Note: The fifth byte (Baud) is 0x00, 0x01, 0x02, 0x03, 0x04. Among them, 0x00 means setting the baud rate to 1000K bps, 0x01 means to set the baud rate to 500K bps, 0x02 means to set the baud rate to 250K bps, 0x03 stands for setting the baud rate 125K bps, 0x04 means to set the baud rate to 100K bps, The default baud rate is 125K bps. After sending this command and receiving the returned data, the sensor needs to be powered on again to valid the baud rate modification.

3)Set the automatic output cycle time (factory default output cycle is 100mS)

A. Set automatic output cycle

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x22	0x00	0x22	0x00	T_L	T_H	0x00	0x00

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x22	0x00	0x22	0x00	T_L	T_H	0x00	0x00

The fifth and sixth bytes represent time, the fifth byte is the low byte, and the sixth byte is the high byte. The time range is 5ms~1000ms;

For example: 10ms TIME_L=0x0A, TIME_H=0x00,
100ms TIME_L=0x64, TIME_H=0x00,
1000ms TIME_L=0xE8, TIME_H=0x03,

The adjustable range is 5ms~1000ms, and the factory default value is 100ms.

B. Query automatic output cycle

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x22	0x00	0x22	0x01	0x00	0x00	0x00	0x00

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x22	0x00	0x22	0x01	T_L	T_H	0x00	0x00

The fifth and sixth bytes represent time, with the fifth byte being the low byte and the sixth byte being the high byte, within the time range 5ms~1000ms;

for example:10ms TIME_L=0x0A TIME_H=0x00

100ms TIME_L=0x64 TIME_H=0x00

1000ms TIME_L=0xE8 TIME_H=0x03

Scope can be set 5ms~1000ms, The factory default value is 100ms.

4)Set the output frame type (Factory default: Angle frame+linear velocity frame+angular rate frame)

The output data frames consist of angle frames, linear velocity frames, angular rate frames, and standard frames. Open the corresponding frame output, and the sensor will output these frames according to the set cycle. If the output cycle is 100ms and the angle frame+linear velocity frame+gyro angular rate frame is turned on, then every 100ms, the angle frame+linear velocity frame+gyro angular rate frame will be continuously output (there will be a small time interval between the angle frame, linear velocity frame, and gyro angular rate frame).

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x30	0x10	0x00	MASK	0x00	0x00	0x00

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x30	0x10	0x00	MASK	0x00	0x00	0x00

The lower 4 bits of the fifth byte Mask are valid. Mask (binary: 0B0000dcba).

a: Representing angle frames (roll angle, pitch angle, azimuth angle), 1: turn on output, 0: turn off output.

b: Represents linear velocity frame (three-axis linear velocity), 1: turns on output, 0: turns off output.

c: Representing the angular rate frame (three-axis angular rate), 1: turn on output, 0: turn off output.

d: Represents standard frame (Z-axis angular rate, forward linear velocity, azimuth), 1: turns on output, 0: turns off output.

Host sends: 40 30 10 00 07 00 00 00, will turn on angle frame, linear velocity frame, and angular rate angular velocity frame output, and turn off standard frame output.

Host sends: 40 30 10 00 05 00 00 00, will open angle frame and angular rate frame output, close linear velocity frame and standard frame output.

5)Clear relative heading

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x10	0x10	0x00	0x10	0x10	0x10	0x10

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x10	0x10	0x00	0x10	0x10	0x10	0x10

Clear the current relative heading angle to zero.

6) Software Reset

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x10	0x10	0x00	0x88	0x88	0x88	0x88

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x10	0x10	0x00	0x88	0x88	0x88	0x88

After the angular rate sends a response message, perform software reset.

7) Range setting of linear velocity

A. Set the range of the linear velocity

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x31	0x0B	0x00	Byte5	0x00	0x00	0x00

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x31	0x0B	0x00	Byte5	0x00	0x00	0x00

Byte5 represents the range of the linear velocity, 0x00 represents 2g; 0x01 represents 4g; 0x02 represents 8g; 0x03 represents 16g.

Attention: After setting the range of the linear velocity, it is necessary to power it on again or reset the software to be effective.

B. Read the current range of the linear velocity

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x31	0x0B	0x01	0x00	0x00	0x00	0x00

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x31	0x0B	0x01	Byte5	0x00	0x00	0x00

Byte5 represents the current range of the linear velocity, 0x00 represents 2g; 0x01 represents 4g; 0x02 represents 8g; 0x03 represents 16g.

8) angular rate range setting

A. Set the angular rate range

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x31	0x0C	0x00	Byte5	0x00	0x00	0x00

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x31	0x0C	0x00	Byte5	0x00	0x00	0x00

Byte5 represents the range of the angular rate, 0x00 represents 125 degrees per second; 0x01 represents 250 degrees per second; 0x02 represents 500 degrees per second; 0x03 represents 1000 degrees per second; 0x04 represents 2000 degrees per second.

Note: As the maximum output of the angular rate can only display+/-327.65 degrees/second, when the selected range is 50010002000 degrees/second, the output value of+/-327.65 degrees/second cannot truly reflect the actual angular rate value.

Attention: After setting the angular rate range, it is necessary to power it on again or reset the software to be effective.

B. Read the current angular rate range

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x31	0x0C	0x01	0x00	0x00	0x00	0x00

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x31	0x0C	0x01	Byte5	0x00	0x00	0x00

Byte5 represents the current range of the angular rate, 0x00 represents 125 degrees per second; 0x01 represents 250 degrees per second; 0x02 represents 500 degrees per second; 0x03 represents 1000 degrees per second; 0x04 represents 2000 degrees per second.

9) Sensor response frequency bandwidth setting

A. Set response frequency bandwidth

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x31	0x0F	0x00	Byte5	0x00	0x00	0x00

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x31	0x0F	0x00	Byte5	0x00	0x00	0x00

Byte5 represents the response frequency 3dB bandwidth of the sensor, 0x00 represents 25Hz; 0x01 represents 50Hz; 0x02 represents 100Hz; 0x03 represents 200Hz.

Attention: After setting the response frequency to 3dB bandwidth, it is necessary to re power on or reset the software to be effective.

B. Read the current response frequency bandwidth

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x31	0x0F	0x01	0x00	0x00	0x00	0x00

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x31	0x0F	0x01	Byte5	0x00	0x00	0x00

Byte5 represents the current sensor response frequency 3dB bandwidth, 0x00 represents 25Hz; 0x01 represents 50Hz; 0x02 represents 100Hz; 0x03 represents 200Hz.

10) Frame format setting

A. Set frame format

Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x31	0x11	0x00	Byte5	0x00	0x00	0x00

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x31	0x11	0x00	Byte5	0x00	0x00	0x00

Byte5 represents frame format, 0x00 represents standard frame (ID is 11 bits); 0x01 represents an extended frame (with an ID of 29 bits).
Attention: After setting the frame format, it is necessary to power on again or reset the software to be effective.
B. Read frame format
Request message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+0x05	0x40	0x31	0x11	0x01	0x00	0x00	0x00	0x00

Response message format

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+0x05	0x40	0x31	0x11	0x01	Byte5	0x00	0x00	0x00

Byte5 represents frame format, 0x00 represents standard frame (ID is 11 bits); 0x01 represents an extended frame (with an ID of 29 bits).
11) Data parsing
The PDO ID can be set to 0x180, 0x280, 0x380, 0x480, 0x580 at the factory, with the default being 0x580.
① There are four types of data frames: angle frame, linear velocity frame, angular rate frame, and standard frame.

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	Data0	Data1	Data2	Data3	Data4	Data5	Data6	flag

Data0-Data6: Represents data, which is determined by the corresponding flag bits to be angles, linear velocitys, and gyro frames.
Flag (ddddcccaa): unsigned single byte, 8-bit representing data frame type and installation measurement method:
Aa: represents the data type of the frame
00: represents the angle frame (roll angle ± 180 °, pitch angle ± 90 °, azimuth ± 180 °);
01: represents linear velocity (± 16.000g);
10: Representing angular rate (± 327.65 °/S);
11: Representing the standard frame (Z-axis angular velocity+Y-axis forward linear velocity+Z-axis azimuth);
Cc: Reserved, set to 00.
Dddd: Installation measurement direction
0000: Horizontal installation measurement direction;
0001: Vertical upward installation measurement direction;
0010: Install measurement direction laterally to the left;
0011: Install measurement direction laterally to the right;
0100: Horizontal reverse installation measurement direction;
The following is a frame analysis of data types with different horizontal measurement methods:

A: Angle data frame

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	XL	XH	YL	YH	ZL	ZH	Temp	0x00

There are eight bytes of data after CAN-ID, the first two bytes are XL, XH is the inclination of X axis (ROLL angle), the third and fourth bytes are YL, and YH is Y axis (PITCH pitch angle) The angle of inclination, the fifth and sixth bytes are ZL, and ZH is the inclination of the Z axis (YAW

relative heading); the angle size is int16_t, the low byte is first, the high byte is after, and finally divided by 100 to get the angle floating point number. The seventh byte is the temperature value, which is a signed single-byte integer.
Angle conversion example: 26 15 DA EA 28 23 19 00
Flag= 0x00, which means horizontal installation measurement mode, the data is angle.
The angle data size of the X-axis roll angle is represented by a 16-bit signed binary number, the upper 8 bits are XH, and the lower 8 bits are XL.
Use the 16-bit signed binary number to convert to a decimal number, and then divide by 100, the result is the angle.
For example: XL=0x26, XH=0x15, the angle is 54.14°
XH XL
0x15 0x26
0x1526 (5414)
The final result is: 5414/100=54.14°
YH YL
0xEA 0xDA 0xEADA(-5414)
The final result is: -5414/100=-54.14°
ZH ZL
0x23 0x28 0x2328(9000)
The final result is: 9000/100=90.00°
Temp 0x19(25) = 25°C.

B: linear velocity data frame

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	XL	XH	YL	YH	ZL	ZH	0x00	0x01

There are eight bytes of data after CAN-ID, the first two bytes are XL, XH is the linear velocity of X axis, the third and fourth bytes are YL, YH is the linear velocity of Y axis, and the fifth and sixth are two bytes. ZL, ZH are the magnitude of the linear velocity of the Z axis; the magnitude of the linear velocity is int16_t, the low byte comes first, and the high byte comes after. Finally, divide by 1000 to get the linear velocity floating point number. The seventh byte is reserved.
Examples of linear velocity conversion:
26 15 DA EA 28 23 19 01
Flag= 0x01, which means horizontal installation measurement mode, the data is linear velocity.
The X-axis linear velocity data size is represented by a 16-bit signed binary number, the upper 8 bits are XH, and the lower 8 bits are XL.
Use the 16-bit signed binary number to convert to a decimal number, and then divide by 1000, the result is the linear velocity.
For example, XL=0x26, XH=0x15, the linear velocity is 5.414g
XH XL
0x15 0x26 0x1526 (5414)
The final result is: 5414/1000=5.414g
YH YL
0xEA 0xDA 0xEADA(-5414)
The final result is: -5414/1000=-5.414g

ZH ZL
0x23 0x28 0x2328(9000)
The final result is: 9000/1000=9.000g

C: gyro data frame

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	XL	XH	YL	YH	ZL	ZH	0x00	0x02

There are eight bytes of data after CAN-ID, the first two bytes are XL, XH is the size of X-axis angular rate, the third and fourth bytes are YL, YH is the size of Y-axis angular rate, and the fifth and sixth bytes are two bytes. ZL, ZH are the size of the Z-axis gyro; the gyro size is int16_t, the low byte is first, the high byte is after, and finally divided by 100 to get the gyro floating point number. The seventh byte is reserved.

Examples of gyro conversion:
26 15 DA EA 28 23 19 02
Flag= 0x02, which means horizontal installation measurement mode, the data is gyro.
The X-axis gyro data size is represented by a 16-bit signed binary number, the upper 8 bits are XH, and the lower 8 bits are XL.
Use the 16-bit signed binary number to convert to a decimal number, and then divide by 100, the result is the size of the top.
For example, XL=0x26, XH=0x15, the top is 54.14°/S
XH XL
0x15 0x26 0x1526 (5414)
The final result is: 5414/100=54.14°/S
YH YL
0xEA 0xDA 0xEADA(-5414)
The final result is: -5414/100=-54.14°/S
ZH ZL
0x23 0x28 0x2328(9000)
The final result is: 9000/100=90.00°/S

D: Standard data frame

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	ZrateL	ZrateH	YaccL	YaccH	ZangL	ZangH	0x00	0x03

After CAN-ID, there are eight bytes of data. The first two bytes, ZrateL and ZrateH, represent the Z-axis angular rate angular velocity size. The third and fourth bytes, YaccL and YaccH, represent the Y-axis body linear velocity. The fifth and sixth bytes, ZangL and ZangH, represent the Z-axis azimuth size; All data items are int16_t, with low bytes at the beginning and high bytes at the end. Zrate and Zang are divided by 100 to obtain the Z-axis angular rate angular velocity and Z-axis azimuth. Divide Yacc by 1000 to obtain the Y-axis body linear velocity, with the 7th byte being reserved byte 0x00 and the 8th byte being 0x03.

Example of standard frame conversion:
26 15 DA EA 28 23 00 03
Flag= 0x03, The data is represented as gyro angular rate frames.
ZrateL=0x26, ZrateH=0x15, The angular rate has a speed of 54.14 °/S
ZrateH ZrateL
0x15 0x26

0x1526(5414)

The final result is 5414/100=54.14 °/S

YaccH YaccL

0xEA 0xDA

0xEADA(-5414)

The final result is: -5414/1000=-5.414g

ZangH ZangL

0x23 0x28

0x2328(9000)

The final result is: 9000/100=90.00°

② There are three ways of data cycle output: single frame angle output (output one frame angle frame according to cycle, the default output mode), double frame linear velocity output (double frame, double frame output according to cycle), three frame angle gyro plus table Output (three frames, output three frames continuously according to the cycle). Can be set at the factory.

A. Single frame of angle output (output one frame of angle frame in cycles, the default output mode), the message format is as follows:

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	XL	XH	YL	YH	ZL	ZH	Temp	0x00

B. Double frame linear velocity output (double frame, double frame output according to cycle), the message format is as follows:

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	XL	XH	YL	YH	ZL	ZH	Temp	0x00

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	XL	XH	YL	YH	ZL	ZH	0x00	0x01

C. Three-frame angle gyro linear velocity output (three frames, three frames are output continuously according to the cycle), the message format is as follows:

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	XL	XH	YL	YH	ZL	ZH	Temp	0x00

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	XL	XH	YL	YH	ZL	ZH	0x00	0x01

CAN-ID	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	XL	XH	YL	YH	ZL	ZH	0x00	0x02