▶ INTRODUCTION

RAMS900 triaxial accelerometer provides accurate acceleration for the control system through CANopen or analog interface. The selected capacitive MEMS sensor has the highest operating accuracy, especially at small amplitudes, where measurement errors due to cross-axis sensitivity are negligible. Therefore, the RAM900S is ideal for the precise acquisition of vibration, such as vibrations on wind turbine towers. Even if the accelerometer is mounted on a solid load-bearing machine part, measurement distortions caused by vibrations can still be excluded. The RAM900S uses a 4th-order low-pass filter to effectively suppress spurious frequencies, such as those caused by yaw brakes in wind turbines. Not only effectively compensates for offsets caused by installation errors or temperature drift, but also significantly improves accuracy. (Configurable: safety relay outputs can be integrated in any safety system and are activated when specified default limits are exceeded). Four factory-configured frequency bands with different filters can be fine-tuned according to the requirements of specific equipment types. For wind turbines, the RAM900S accelerometer is ideal for implementing the safety function "excessive shock and tower vibration protection". Long-term monitoring can prevent damage or even destruction of equipment. For the aggressive environment of category C5M, the shell design made of glass fiber reinforced plastic is the most ideal, and can ensure the long-term and safe operation of offshore engineering equipment. In terms of product reliability and stability, various measures such as industrial-grade MCU and three-proof PCB board are also used to improve the reliability of the product.

▶ MAIN FEATURES

★ Compact design

★ Temperature range –40 ... +85 °C

★ Compensates for temperature drift and installation errors

★ Analog and CANopen interfaces

★ 3DB bandwidth 10 Hz

★ IP67 protection level

▶ APPLICATION

★ Anti-sway control of gantry cranes

★ Rotation detection of mining bucket excavators

★ Vibration monitoring of large machines and equipment

★ Side sway monitoring of operating platforms

★ Nacelle torsion monitoring of wind turbines

▶ SPECIFICATIONS

RAM900S	Condition	Parameter	Unit
Range	CANopen	±2	g
	Current	±2	g
Axis	CANopen	X, Y, |X,Y|	Axis
	Current	|X|, |Y|	Axis
Bias calibration		<3	mg
Resolution	CANopen	1	mg
	Current	1	mg
Power-on repeatability		5	mg
Analog Sensitivity (±10%)		16	mA/g
Bias temperature coefficient		0.01	%/℃
Non-linearity		<0.5	%FS(max)
Long-term stability		<30	mg
Full temperature range working accuracy	-40 ~ 80℃	30	mg
Power-on start time		<2	s
Response time		0.002	s
Response frequency	5Hz		
3DB bandwidth	10Hz		
Output signal	CANopen, Analog(current 4~20mA)，Switch (relay switch)		
Supply voltage	Input 9~36VDC		
Supply current	<50mA @12VDC		
Relay switching threshold	default|X,Y|≥0.16g		
CANopen(Baudrate)	Factory default(500Kbps)		
CANopen(Node ID)	Factory default(0x01)		
Weight	≤250g(not include cable)		

Note: CAN bus terminal resistance can be added on request, or remark can be added to PCBA. "|X,Y|" as its level of vehicle body acceleration.

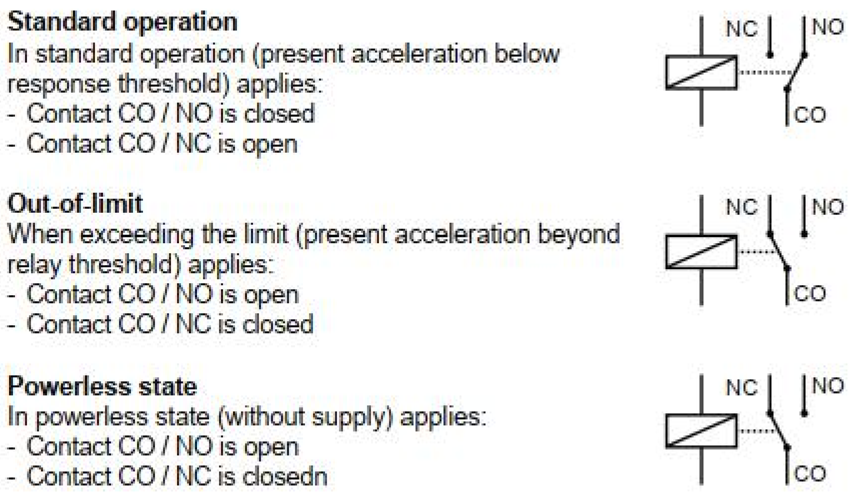
SWITCH OUTPUT CONTROL

Relay switching maximum voltage: 30VDC

Relay switching maximum current: 1.0A

Relay response threshold: set according to user requirements (default is | X , Y| = 0.16g).

Relay work: Under the correct circumstances, after the equipment is powered on, NO and CO contact closed (closed), NC and CO disconnected (open). When it detecte the acceleration of X, Y exceeds the set threshold, the relay NO is open with the CO, and the NC is closed with the CO.



Standard operation
In standard operation (present acceleration below response threshold) applies:
- Contact CO / NO is closed
- Contact CO / NC is open

Out-of-limit
When exceeding the limit (present acceleration beyond relay threshold) applies:
- Contact CO / NO is open
- Contact CO / NC is closed

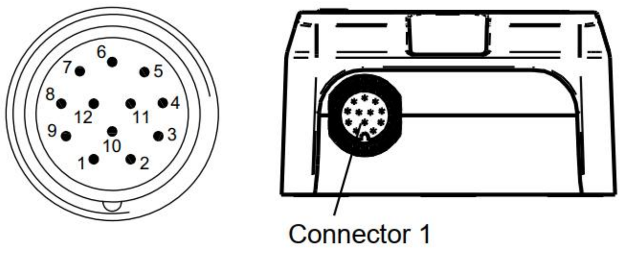
Powerless state
In powerless state (without supply) applies:
- Contact CO / NO is open
- Contact CO / NC is closedn

OUTPUT CONFIGURATION

Output	CANopen 1	CANopen 2	CANopen 3	CANopen 4	Analog1 current	Analog2 current
Axial	X	Y	|X,Y|	|X,Y|	X	Y
Range	±2g	±2g	±2g	±2g	±2g	±2g
Resolution	1.00mg	1.00mg	1.00mg	1.00mg	1.00mg	1.00mg
Filter type	Lowerpass	Lowerpass	Lowerpass	Lowerpass	Lowerpass	Lowerpass
3DB bandwidth	10Hz	10Hz	10Hz	10Hz	10Hz	10Hz
Relay ID	0	0	-	1		
Relay Action Threshold	-	-	-	See part no.	-	-
Relay trigger time	-	-	-	0s	-	-
Relay decay value	-	-	-	100%		
Relay decay time	-	-	-	1s		

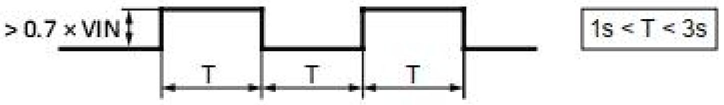
* can provide according to the different configuration, the default relay action thresholds for: 0.16 g (| XY | level of vehicle body acceleration).

WIRING

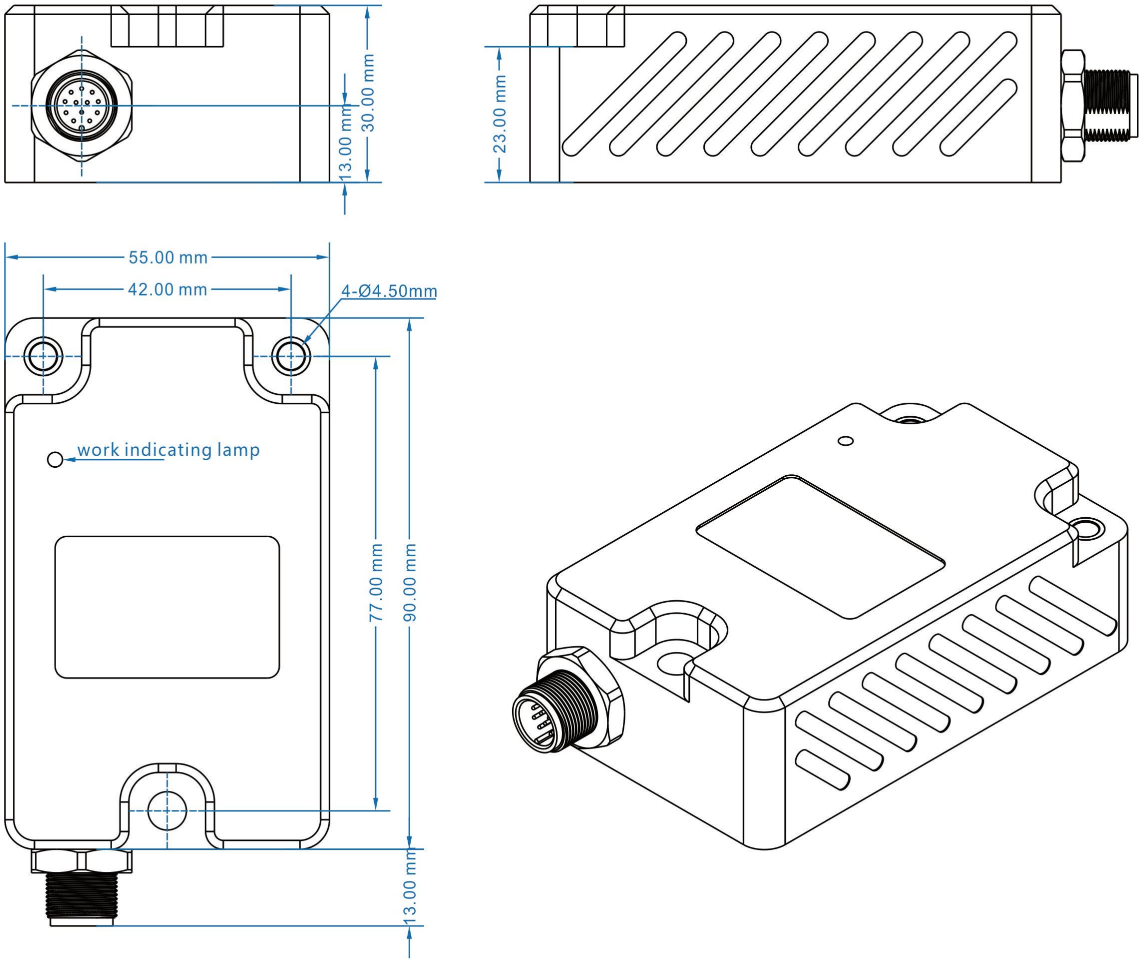


NO.	Definition	Color (if with cable)	Remark
1	GND	Brown	power ground
2	TEST INPUT*	Blue	test input
3	VIN	White	9-36VDC
4	AGND	Dark green	Analog ground
5	X1	pink	Analog channel X1
6	Y1	yellow	Analog channel Y1
7	RELAY_NO	black	Relay normally open
8	CAN_GND	grey	CAN signal ground
9	RELAY_CO	red	Relay common
10	RELAY_NC	Purple	Relay Normal close
11	CAN_LOW	orange	CAN_LOW signal
12	CAN_HIGH	light green	CAN_HIGH signal

test method of the TEST INPUT pin:
TEST INPUT---Mainly used for relay testing during maintenance. In order to ensure the normal function of the relay, the user can perform periodical (once a year) maintenance on purpose, and can apply two consecutive high-level signals to the TEST INPUT pin, and the relay will meet the requirements. Perform action switching.



▶ SIZE


30.00 mm
13.00 mm
23.00 mm
55.00 mm
42.00 mm
4-Ø4.50mm
work indicating lamp
77.00 mm
90.00 mm
13.00 mm

▶ CANopen communication protocol

1. Product communication protocol object dictionary (EDS)

Index (hex)	Sub Index	Name	Data/ Object	Access Type	Default (Hex)	Comment
1000	**-**	**Device type**	**Uint32**	**RO**	**0x0004019A**	
1001	**-**	**Error register**	**Uint8**	**RO**	**0**	
1002	**-**	**Manufacturer status register**	**Uint32**	**RO**	**0**	**ADC errors/timeouts,etc**
1003		**Predefined error field**	**Array**			**Contains list of recent errors**
	0	Number of errors	Uint8	RW	xx	
	xx	No. error	Uint32	RO		
1005	**-**	**COB-ID SYNC message**	**Uint32**	**RW**	**0x00000080**	
1008		**Device Name**	**String**	**RO**	**“Acceleration Sensor RAM900S”**	
1009	**-**	**Hardware version**	**String**	**RO**	**“1.00”**	
100A	**-**	**Software version**	**Sting**	**RO**	**“1.00”**	
1010		**Store Parameters**	**Array**			
	0	Largest supported sub-index	Uint8	RO	4	
	1	Save all parameters	Uint32	RW	1/0x65766173	“save”
	2	Save communication parameters	Uint32	RW	1/0x65766173	“save”
	3	Save application parameters	Uint32	RW	1/0x65766173	“save”
	4	Save manufacturer parameters	Uint32	RW	1/0x65766173	“save”
1011		**Restore default parameters**	**Array**			
	0	Largest supported sub-index	Uint8	RO	4	
	1	Restore all default parameters	Uint32	RW	1/0x64616F6C	“load”
	2	Restore default communication parameters	Uint32	RW	1/0x64616F6C	“load”
	3	Restore default application parameters	Uint32	RW	1/0x64616F6C	“load”
	4	Restore default manufacturer	Uint32	RW	1/0x64616F6C	“load”

		parameters				
1014	**-**	**Emergency COB-ID**	**Uint32**	**RW**	**0x00000080+Node_ID**	
1016		**Consumer heartbeat time**	**Array**			
	0	Largest supported sub-index	Uint8	RO	1	
	1	Consumer heartbeat time	Uint32	RW	0	
1017	**-**	**Producer heartbeat time**	**Uint16**	**RW**	**0**	
1018		**Identity object**	**Record**			
	0	Largest supported sub-index	Uint8	RO	4	
	1	Vendor ID	Uint32	RO	0xEC	
	2	Product code	Uint32	RO	0x50(RAM900S)	
	3	Revision number	Uint32	RO	0x04040121	
	4	Serial number	Uint32	RO	0x4C70AB65	
1800		**1st transmit PDO communication parameter**	**Record**			
	0	Largest supported sub-index	Uint8	RO	5	
	1	COB-ID used by PDO	Uint32	RO	0x00000180+NodeID	
	2	Transmission type	Uint8	RW	0xFF	
	5	**Event time**	**Uint16**	**RW**	**20**	**Factory default 20mS**
1A00		**1st transmit PDO Mapped Object**	**RECORD**			
	0	Number of mapped objects	Uint8	RO	4	
	1	Transmit PDO object	UINT32	RO	0x30000110	
	2	Transmit PDO object	UINT32	RO	0x30000210	
	3	Transmit PDO object	UINT32	RO	0x30000310	
	4	Transmit PDO object	UINT32	RO	0x30000410	
1F80	**-**	**NMT startup**	**Uint32**	**RW**	**0 or 8**	
2100	**-**	**Baud rate**	**Uint8**	**RW**	**02**	**01:20K** **02:50K(Factory default)** **03:100K 04:125K** **05:250K 06:500K** **07:800K 08:1M**
2101	**-**	**NodeId**	**Uint8**	**RW**	**1**	**Range:0x01-0x7F**

						Factory default:0x01
3000		Filterband Outputs	RECORD			

2. Detailed specification of communication protocol objects

2.1Communication Profile Area(communication protocol area)

2.1.1OD Index (0x1000): Device Type(device type， UINT32， RO)

I. Query device type

SDO request message format

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

SDO Response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x00	0x10	0x00	0x9A	0x01	0x04	0x00

Note：Device Type = 0x0004019A.

2.1.2 OD Index (0x1001): Error Register(error register， UINT8， RO)

I. Query the content of the error register and save the latest error type.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x01	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x01	0x10	0x00	0x10	0x00	0x00	0x00

Note: Error Register = 0x10，A communication error occurred.

Bit	Error type
0	generic
1	current
2	voltage
3	temperature
4	communication
5	device profile specific
6	reserved (=0)
7	manufacturer specific

2.1.3 OD Index (0x1002): Manufacturer status Register

(Factory Defined Error Status Register, UINT32, RO)

I. Query the content of the error status register defined by the manufacturer.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x02	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x02	0x10	0x00	0x00	0x00	0x00	0x00

Note: Manufacturer status Register = 0x00000000.

2.1.4 OD Index (0x1003): Pre-defined error field(pre-defined error field, ARRAY)

This object represents errors generated on the CANopen sensor, programmed by the emergency object, and provides error history.

2.1.4.1 Sub-Index:0x00(Number of errors, RW)

A. Query the number of errors in the error field

If no errors exist, its sub-index 0x00 has the value 0x00.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x03	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x03	0x10	0x00	0x02	0x00	0x00	0x00

Note: Number of errors = 0x02, the number of errors is 2.

B. Clear the number of errors in the error field

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x2F	0x03	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x03	0x10	0x00	0x00	0x00	0x00	0x00

Note: Clear Number of errors.

2.1.4.2 Sub-Index:0x01(the 1st error,UINT32， RO)

A. Query the first error code of Sub-Index 0x01

If the content of sub-index 0x00 is not zero, it indicates the number of errors that have occurred in the sensor, where the content of sub-index 0x01 is the latest error code, 0x02 is the next most recent error code, and so on.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x03	0x10	0x01	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x03	0x10	0x01	0x00	0x81	0x00	0x03

Note: Error Code = 0x03008100。

The upper 16 bits of the Error Code are composed of additional information and the lower 16 bits of the error code. For error codes, please refer to the standard CANopen specification DS-302.

2.1.4.3 Sub-Index:0x02(the 2nd error,UINT32， RO)

A. Query Sub-Index 0x02 Second error code

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x03	0x10	0x02	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x03	0x10	0x02	0x20	0x81	0x06	0x04

Note: Error Code = 0x04068120。

The upper 16 bits of the Error Code are composed of additional information and the lower 16 bits of the error code. For error codes, please refer to the standard CANopen specification DS-302.

2.1.5 OD Index (0x1005): COB-ID SYNC message

(COB-ID synchronization message, UINT32, RW, default: 0x00000080)

This object is the configured synchronization (SYNC) object COB-ID. It defines whether a CANopen device generates synchronization or not. The object structure is defined as follows:

31 30 29 28 11 10 0

<table>
<tr><td rowspan="2">X</td><td rowspan="2">Gen.</td><td rowspan="2">frame</td><td>0000h</td><td>11 bits CAN-ID</td></tr>
<tr><td colspan="2">29 bits CAN-ID</td></tr>
</table>

Synchronized COB-ID structure

Bit (S)	value	Description
x	x	meaningless
Gen.	0 1	The CANopen device does not generate synchronization messages The CANopen device generates synchronization messages
Frame	0 1	11-bit CAN-ID is valid (CAN standard frame) 29-bit CAN-ID is valid (CAN extended frame)
29 bits CAN-ID	x	29-bit extended frame CAN-ID
11 bits CAN-ID	x	11-bit standard frame CAN-ID

Synchronous COB-ID description

A. Query COB-ID synchronization message

SDO request message format

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

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x05	0x10	0x00	0x80	0x00	0x00	0x00

Note: The synchronization COB-ID is: 0x00000080.

B. Set COB-ID synchronization message

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x23	0x05	0x10	0x00	0x81	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x05	0x10	0x00	0x00	0x00	0x00	0x00

Note: Set Sync COB-ID = 0x00000081.

2.1.6 OD Index (0x1008): Device name

A. (Device name, CONST STRING, RO, fixed as: "Acceleration Sensor RAM900S")

B. A. Query Device name.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x08	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format(Response reply is block download, please refer to CANopen specification)。

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x41	0x08	0x10	0x00	0x1B	0x00	0x00	0x00

Note: The fifth byte is the string length.

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x60	0x08	0x10	0x00	0x00	0x00	0x00	0x00
0x580+ Node_ID	0x00	0x41	0x63	0x63	0x65	0x6C	0x65	0x72
		‘A’	‘c’	‘c’	‘e’	‘l’	‘e’	‘r’
0x600+ Node_ID	0x70	0x08	0x10	0x00	0x00	0x00	0x00	0x00
0x580+ Node_ID	0x10	0x61	0x74	0x69	0x6F	0x6E	0x20	0x53
		‘a’	‘t’	‘i’	‘o’	‘n’	‘ ‘	‘S’

0x600+ Node_ID	0x60	0x08	0x10	0x00	0x00	0x00	0x00	0x00
0x580+ Node_ID	0x00	0x65	0x6E	0x73	0x6F	0x72	0x20	0x52
		‘e’	‘n’	‘s’	‘o’	‘r’	‘ ‘	‘R’
0x600+ Node_ID	0x70	0x08	0x10	0x00	0x00	0x00	0x00	0x00
0x580+ Node_ID	0x13	0x41	0x4D	0x39	0x30	0x30	0x53	0x00
		‘A’	‘M’	‘9’	‘0’	‘0’	‘S’	

2.1.7 OD Index (0x1009): Hardware version

(hardware version, CONST STRING, RO, fixed to: "1.00")

A. Query the Hardware version.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x09	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format(Response reply is block download, please refer to CANopen specification).

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x42	0x09	0x10	0x00	0x31	0x2E	0x30	0x30
					‘1’	‘.’	‘0’	‘0’

Note: hardware version = “1.00”.

2.1.8 OD Index (0x100A): Software version

(software version, CONST STRING, RO, fixed as: "1.00")

A. Query the Hardware version.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x0A	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format (Response reply is block download, please refer to CANopen specification).

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x42	0x0A	0x10	0x00	0x31	0x2E	0x30	0x30
					‘1’	‘.’	‘0’	‘0’

Note: software version = “1.00”.

2.1.9 OD Index (0x1010): Store parameters (save parameters, ARRAY)

Expresses the ability to store relevant information provided by CANopen through read access.

Subindex 0x00 contains the subindex number.

Sub-index 0x01 refers to all parameters that can be stored on the CANopen device.

Sub-index 0x02 refers to communication-related parameters (indexes from 1000h to 1FFFh).

Sub-index 0x03 refers to application-related parameters (indexes from 6000h to 9FFFh).

Sub-index 0x04 refers to an individual parameter optional by the manufacturer.

2.1.9.1 Sub-Index:0x00(Number of sub-index, RO)

A. Query the number of sub-indexes

Its sub-index 0x00 has the value 0x04.

SDO request message format

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

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x10	0x10	0x00	0x04	0x00	0x00	0x00

Note: Number of Sub-Index = 0x04，The number of sub-indexes is 4.

2.1.9.2 Sub-Index:0x01(Save all parameters, UINT32， RW)

A. Query Sub-Index 0x01 to save all parameters

Query the content of the sub-index 0x01, mainly to read the access structure, to know whether the device automatically saves all parameters, or executes the save action according to the command.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x10	0x10	0x01	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x10	0x10	0x01	0x01	0x00	0x00	0x00

Note: Access structure content = 0x00000001, the CANopen device executes the action of saving parameters according to the command.

31	2 1	0
(00 0000 0000 0000 0000 0000 0000 0000) reserved	auto	cmd

Bit	Value	
auto	0 1	CANopen device does not save parameters automatically CANopen device automatically saves parameters
cmd	0 1	CANopen device does not accept save parameter command The CANopen device executes the action of saving parameters according to the command

access structure

B. Write Sub-Index 0x01 to save all parameters.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x23	0x10	0x10	0x01	0x73	0x61	0x76	0x65

Note: write “save”。

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x10	0x10	0x01	0x00	0x00	0x00	0x00

Note: All parameters set after recovery will be saved after power-off, and the new setting parameters will be valid after power-on or node reset.

2.1.9.3 Sub-Index: 0x02 (save communication parameters, UINT32, RW)

A. Query Sub-Index 0x02 to save communication parameters

Query the content of the sub-index 0x02, mainly to read the access structure, to know whether the device automatically saves the communication parameters, or executes the saving action according to the command.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x10	0x10	0x02	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x10	0x10	0x02	0x01	0x00	0x00	0x00

Note: Access structure content = 0x00000001, the CANopen device executes the action of saving parameters according to the command.

31	2	1 0
(00 0000 0000 0000 0000 0000 0000 0000) reserved	auto	cmd

Bit	Value	
auto	0 1	CANopen device does not save parameters automatically CANopen device automatically saves parameters
cmd	0 1	CANopen device does not accept save parameter command The CANopen device executes the action of saving parameters according to the command

Access structure

B. Write Sub-Index 0x02 to save communication parameters.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x23	0x10	0x10	0x02	0x73	0x61	0x76	0x65

Note: write “save”。

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x10	0x10	0x02	0x00	0x00	0x00	0x00

Note: The communication parameters set after the reply will be saved after power-off, and the new communication parameters will be valid after power-on or node reset.

2.1.9.4 Sub-Index:0x03(Save application parameters,UINT32， RW)

A. Query Sub-Index 0x03 to save application parameters

Query the content of the sub-index 0x03, mainly to read the access structure, to know whether the device automatically saves the application parameters, or executes the save action according to the command.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x10	0x10	0x03	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x10	0x10	0x03	0x01	0x00	0x00	0x00

Note: Access structure content = 0x00000001, the CANopen device executes the action of saving parameters according to the command.

31 2	1	0
(00 0000 0000 0000 0000 0000 0000 0000) reserved	auto	cmd

Bit	Value	
auto	0 1	CANopen device does not save parameters automatically CANopen device automatically saves parameters
cmd	0 1	CANopen device does not accept save parameter command CANopen device executes the action of saving parameters according to the command

Access structure

B. Write Sub-Index 0x03 to save application parameters.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x23	0x10	0x10	0x03	0x73	0x61	0x76	0x65

Note: write“save”。

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x10	0x10	0x03	0x00	0x00	0x00	0x00

Note: The application parameters set after recovery will be saved after power-off, and the new application parameters will be valid after re-powering or resetting the node.

2.1.9.5 Sub-Index:0x04(Save manufacturer-defined parameters,UINT32， RW)

A. Query Sub-Index 0x04 to save the parameters defined by the manufacturer

Query the content of the sub-index 0x04, mainly to read the access structure, to know whether the device automatically saves the parameters defined by the manufacturer, or executes the save action according to the command.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x10	0x10	0x04	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x10	0x10	0x04	0x01	0x00	0x00	0x00

Note: Access structure content = 0x00000001, the CANopen device executes the action of saving parameters according to the command.

31 2	1	0
(00 0000 0000 0000 0000 0000 0000 0000) reserved	auto	cmd

Bit	Value	
auto	0 1	CANopen device does not save parameters automatically CANopen device automatically saves parameters
cmd	0 1	CANopen device does not accept save parameter command CANopen device executes the action of saving parameters according to the command

Access structure

B. Write Sub-Index 0x04 to save the parameters defined by the manufacturer。

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x23	0x10	0x10	0x04	0x73	0x61	0x76	0x65

Note: write “save”。

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x10	0x10	0x04	0x00	0x00	0x00	0x00

Note: The manufacturer-defined parameters set after recovery will be saved after power-off, and the new manufacturer-defined parameters will be valid after re-powering or resetting the node.

2.1.10 OD Index (0x1011): Restore parameters(restore default parameters， ARRAY)

With read access, Express CANopen provides the ability to save relevant information about the recovery.

Subindex 0x00 contains the subindex number.

Sub-index 0x01 refers to restore all parameters.

The sub-index 0x02 refers to the parameters related to resume communication (index from 1000h~1FFFh).

Sub-index 0x03 refers to recovery application-related parameters (indexes from 6000h to 9FFFh).

Sub-index 0x04 refers to the recovery of manufacturer-optionally defined parameters.

2.1.10.1 Sub-Index:0x00(Number of sub-index, RO)

A. Query the number of sub-indexes

Sub-index 0x00 has a value of 0x04.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x11	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x11	0x10	0x00	0x04	0x00	0x00	0x00

Note: Number of Sub-Index = 0x04，The number of sub-indexes is 4.

2.1.10.2 Sub-Index:0x01(Restore all parameters to default,UINT32， RW)

A. Query Sub-Index 0x01 Restore all parameters to default

Query the content of the sub-index 0x01, mainly read the Access structure to know whether the device has the function of restoring all parameters.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x11	0x10	0x01	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x11	0x10	0x01	0x01	0x00	0x00	0x00

Note: Access structure content=0x00000001, CANopen supports recovery parameter action.

31 2	1	0
(00 0000 0000 0000 0000 0000 0000 0000) reserved	0	cmd

Bit	Value	
cmd	0	CANopen devices do not support restoring default parameters
	1	CANopen devices support restoring default parameters

Access structure

B. write Sub-Index 0x01 Restore all parameters to default。

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x23	0x11	0x10	0x01	0x6C	0x6F	0x61	0x64

Note: write“load”。

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x11	0x10	0x01	0x00	0x00	0x00	0x00

Note: After replying, all parameters will be restored to default values.

2.1.10.3 Sub-Index:0x02(restore default communication parameters,UINT32， RW)

A. Query Sub-Index 0x02 restore default communication parameters

Query the content of the sub-index 0x02, mainly read the Access structure to know whether the device has the function of restoring communication parameters.。

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x11	0x10	0x02	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x11	0x10	0x02	0x01	0x00	0x00	0x00

Note: Access structure Content=0x00000001, CANopen supports parameter recovery action.

31	2 1	0
(00 0000 0000 0000 0000 0000 0000 0000) reserved	0	cmd

Bit	Value	
cmd	0	CANopen devices do not support restoring default parameters
	1	CANopen devices support restoring default parameters

Access structure

B. write Sub-Index 0x02 restore default communication parameters。

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x23	0x11	0x10	0x02	0x6C	0x6F	0x61	0x64

Note: write“load”。

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x11	0x10	0x02	0x00	0x00	0x00	0x00

Note: After the reply, the communication parameters will be restored to the default values.

2.1.10.4 Sub-Index:0x03(Restore default application parameters,UINT32， RW)

A. Query Sub-Index 0x03 to restore application parameters

Query the contents of the sub-index 0x03, mainly read the Access structure to know whether the device has the function of restoring application parameters.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x11	0x10	0x03	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x11	0x10	0x03	0x01	0x00	0x00	0x00

Note: Access structure Content=0x00000001, CANopen supports parameter recovery action.

31	2 1	0
(00 0000 0000 0000 0000 0000 0000 0000) reserved	0	cmd

Bit	Value	
cmd	0	CANopen devices do not support restoring default parameters
	1	CANopen devices support restoring default parameters

Access structure

B. Write Sub-Index 0x03 to restore the default application parameters。

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x23	0x11	0x10	0x03	0x6C	0x6F	0x61	0x64

Note: write“save”。

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x11	0x10	0x03	0x00	0x00	0x00	0x00

Note: After replying, the application parameter will restore the default value.

2.1.10.5 Sub-Index:0x04(Restores default parameters defined by the manufacturer，UINT32， RW)

A. Query Sub-Index 0x04 to restore the default parameters defined by the manufacturer

Query the content of the sub-index 0x04, mainly read the Access structure to know whether the device has the function of restoring the default parameters defined by the manufacturer.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x11	0x10	0x04	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x11	0x10	0x04	0x01	0x00	0x00	0x00

Note: Access structure content = 0x00000001, CANopen supports recovery parameter action.

31	2 1	0
(00 0000 0000 0000 0000 0000 0000 0000) reserved	0	cmd

Bit	Value	
cmd	0	CANopen devices do not support restoring default parameters
	1	CANopen devices support restoring default parameters

Access structure

B. Writing Sub-Index 0x04 restores the default parameters defined by the manufacturer.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x23	0x10	0x10	0x04	0x6C	0x6F	0x61	0x64

Note: write“save”。

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x10	0x10	0x04	0x00	0x00	0x00	0x00

Note: After replying, the parameters defined by the manufacturer will return to the default values.

2.1.11 OD Index (0x1014): Emergency COB-ID(Emergency COB-ID, UINT32, RW, default: 0x00000080+NodeID)

The object structure is defined as follows:

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28 … 11</td><td>10 … 0</td></tr>
<tr><td rowspan="2">valid</td><td rowspan="2">0</td><td rowspan="2">frame</td><td>0000h</td><td>11 bits CAN-ID</td></tr>
<tr><td colspan="2">29 bits CAN-ID</td></tr>
</table>

Synchronized COB-ID structure

Bit （S）	Value	Description
valid	0	EMERGENCY ID exists/valid
	1	EMERGENCY ID does not exist/invalid
30	0	reserved (always 0)
Frame	0	11-bit CAN-ID is valid (CAN standard frame)
	1	29-bit CAN-ID is valid (CAN extended frame)
29 bits CAN-ID	x	29-bit extended frame CAN-ID
11 bits CAN-ID	x	11-bit standard frame CAN-ID

Emergency COB-ID Description

A. Query Emergency-ID

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x14	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x14	0x10	0x00	0x81	0x00	0x00	0x00

Note: sync Emergency-ID as： 0x00000081+NodeID。

B. Set Emergency-ID

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x23	0x14	0x10	0x00	0x83	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x14	0x10	0x00	0x00	0x00	0x00	0x00

Note: set sync Emergency-ID = 0x00000083。

2.1.12 OD Index (0x1016): Consumer heartbeat time(Consumer heartbeat timed out， ARRAY)

Sub-index 0x00: supports the maximum number of sub-indexes, UINT8, fixed to 0x01.

Sub-index 0x01: consumer heartbeat timeout, UINT32, RW.

The values are defined as follows:

31 24	23 16	15 0
Reserve（00h)	Node-ID	heartbeat timeout

- Bits 0 to 15: User heartbeat time (ms).

- Bits 16 to 23: The node ID of the producer whose heartbeat is to be monitored.

If the heartbeat time is 0 or the Node-ID is 0 or greater than 127, the corresponding object entry is invalid.

2.1.12.1 Query the number of subindex entries

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x16	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x16	0x10	0x00	0x01	0x00	0x00	0x00

Note: The subindex entries are:0x01。

2.1.12.2 Sub-index 0x01 consumer heartbeat timed out

A. Query consumer heartbeat timeout

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x16	0x10	0x01	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x16	0x10	0x01	0x00	0x00	0x00	0x00

Note: Subindex entry 0x01 is 0x00000000, invalid.

B. Set 0x01 consumer heartbeat timeout

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+Node_ID	0x23	0x16	0x10	0x01	0x64	0x00	0x7F	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	0x60	0x16	0x10	0x01	0x00	0x00	0x00	0x00

Note: The setting content is 0x007F0064, the monitoring ID is 127, and the heartbeat timeout is 100mS.

2.1.13 OD Index (0x1017): Producer heartbeat time(Producer heartbeat timeout， UINT16)

A. Query producer heartbeat timeout

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+Node_ID	0x40	0x17	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	0x4B	0x17	0x10	0x00	0x00	0x00	0x00	0x00

Note: The producer heartbeat timeout is: 0x00, if it is 0, the producer heartbeat timeout is invalid.

B. Set the producer heartbeat timeout

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+Node_ID	0x2B	0x17	0x10	0x00	0x64	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+Node_ID	0x60	0x17	0x10	0x00	0x00	0x00	0x00	0x00

Note: The setting content is 0x0064, and the producer heartbeat timeout is 100mS. After the setting takes effect, the sensor generates heartbeat messages with a timing period of 100mS.

2.1.14 OD Index (0x1018): Identity object(object identity， RECORD)

Through read access, express the identification information provided by CANopen.

Sub-index 0x00 contains the largest number of sub-indexes.

Sub-index 0x01 refers to VendorID, UINT32.

Sub-index 0x02 refers to Product code, UINT32.

Sub-index 0x03 refers to the Revision number, UINT32.

Sub-index 0x04 refers to Serial number, UINT32.

2.1.14.1 Sub-Index:0x00(Number of sub-index, RO)

A. Query the number of sub-indexes

Its sub-index 0x00 has the value 0x04.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x18	0x10	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x18	0x10	0x00	0x04	0x00	0x00	0x00

Note: Number of Sub-Index = 0x04，The number of sub-indexes is 4.

2.1.14.2 Sub-Index:0x01 Vendor ID (supplier code,UINT32， RO)

A. Query Sub-Index 0x01 Vendor ID

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x18	0x10	0x01	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x18	0x10	0x01	0xEC	0x00	0x00	0x00

Note: Vendor ID =0x000000EC。

2.1.14.3 Sub-Index:0x02 Product code (Product Code,UINT32， R0)

A. Query Sub-Index 0x02 Product code

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x18	0x10	0x02	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x18	0x10	0x02	0x50	0x00	0x00	0x00

Note: Product code = 0x00000050。

2.1.14.4 Sub-Index:0x03(Revision number,UINT32， RO)

A. Query Sub-Index 0x03 Revision number

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x18	0x10	0x03	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x18	0x10	0x03	0x01	0x02	0x03	0x04

2.1.14.5 Sub-Index:0x04(Serial number,UINT32， RO)

A. Query Sub-Index 0x04 Serial number

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x18	0x10	0x04	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x18	0x10	0x04	0x01	0x01	0x01	0x01

Note: Serial number =0x01010101。

2.1.15 OD Index(0x1800): 1st transmit PDO communication parameter (Send PDO communication parameters, RECORD)

Through read access, express the identification information provided by CANopen.

Subindex 0x00 contains the largest number of subindexes, UINT8.

Sub-index 0x01 refers to PDO COB-ID, UINT32.

Sub-index 0x02 refers to the Transmission type, UINT8.

Sub-index 0x05 refers to Event time, UINT16.

2.1.15.1 Sub-Index:0x00(Number of sub-index, RO)

A. Query the number of sub-indexes

Its sub-index 0x00 has the value 0x05.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x18	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x00	0x18	0x00	0x05	0x00	0x00	0x00

Note: Number of Sub-Index = 0x05，The number of sub-indexes is 5。

2.1.15.2 Sub-Index:0x01 PDO COB-ID (PDO COB-ID,UINT32， RO)

A. Query Sub-Index 0x01 PDO COB-ID

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x18	0x01	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x00	0x18	0x01	0x81	0x01	0x00	0x40

Note: PDO COB-ID =0x40000181（TPDO ID = 0x180 + 0x01, not support RTR）。

TPDO's COB-ID structure

31	30	29	28 11	10 0
valid	RTR	frame	0 0000h	11-bit CAN-ID
			29-bit CAN-ID	

COB-ID description of TPDO

Bit （S）	Value	Description
Valid	0	PDO exists/valid
	1	PDO does not exist/invalid
RTR	0	This PDO allows RTR
	1	This PDO does not allow RTR
frame	0	11-bit CAN-ID is valid (standard frame)
	1	29-bit CAN-ID is valid (extended frame)
29 bits CAN-ID	x	29-bit CAN-ID is valid (extended frame)
11 bits CAN-ID	x	11-bit CAN-ID is valid (standard frame)

2.1.15.3 Sub-Index:0x02 Transmission type (transfer type,UINT8， RW)

A. Query Sub-Index 0x02 Transmission type

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x18	0x02	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x00	0x18	0x02	0xFF	0x00	0x00	0x00

Note: Transmission type = 0xFF。

Transmission Type	Condition to trigger PDO (B = both needed, O = one or both)			PDO Transmission
	SYNC	RTR	EVENT	
0	B	-	B	Sync, acyclic
1-240	O	-	-	Sync, acyclic
241-251	-	-	-	reserved
252	B	B	-	Sync, after RTR
253	-	O	-	Async, after RTR
254	-	O	O	Async, manufacturer specific event
255	-	O	O	Async, device profile specific event

B. set Sub-Index 0x02 Transmission type

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x2F	0x00	0x18	0x02	0x01	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x00	0x18	0x02	0x00	0x00	0x00	0x00

Note: set Transmission type = 0x01，In the same way, the count is 1.

2.1.15.4 Sub-Index:0x05(Event Time,UINT16，unit：mS， RW)

A. query Sub-Index 0x05 Event Time

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x18	0x05	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4B	0x00	0x18	0x05	0x64	0x00	0x00	0x00

Note: Event Time =0x0064 = 100mS.

B. set Sub-Index 0x05 Event Time

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x2B	0x00	0x18	0x05	0xE8	0x03	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x00	0x18	0x05	0x00	0x00	0x00	0x00

Note: Event Time =0x03E8 = 1000mS。

2.1.16 OD Index (0x1A00): 1st transmit PDO communication Mapping (PDO Communication parameter mapping， RECORD)

Express CANopen PDO communication parameter mapping with read access.

Sub-index 0x00 contains the largest number of sub-indexes, Number of mapped objects, UINT8.

Sub-index 0x01 refers to Transmit PDO object1, UINT32.

Sub-index 0x02 refers to Transmit PDO object2, UINT32.

Sub-index 0x03 refers to Transmit PDO object3, UINT32.

Sub-index 0x04 refers to Transmit PDO object4, UINT32.

OD object dictionary index 0X1A00 mapping parameters

32-bit range Transmit PDO mapping.

object 0X1A00 the first Transmis PDO		
Sub-index	Value	Meaning
0	4	4 objects mapped to PDO
1	0X30000110	Object 0x3000, subindex 01, consists of 16 bits
2	0X30000210	Object 0x3000, subindex 02, consists of 16 bits
3	0X30000310	Object 0x3000,subindex 03,consists of 16 bits
4	0X30000410	Object 0x3000,subindex 04,consists of 16 bits

2.1.16.1 Sub-Index:0x00(Number of sub-index, RO)

A. Query the number of sub-indexes

Its sub-index 0x00 has the value 0x4.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x1A	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x00	0x1A	0x00	0x04	0x00	0x00	0x00

Note: Number of mapped objects = 0x04，The number of sub-indexes is 4.

2.1.16.2 Sub-Index:0x01 Transmit PDO object1 (UINT32， RO)

A. query Sub-Index 0x01 Transmit PDO object1

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x1A	0x01	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x00	0x1A	0x01	0x10	0x01	0x00	0x30

Note: Transmit PDO object1 = 0x30000110。

2.1.16.3 Sub-Index:0x01 Transmit PDO object2 (UINT32， RO)

A. query Sub-Index 0x01 Transmit PDO object2

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x1A	0x02	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x00	0x1A	0x02	0x10	0x02	0x00	0x30

Note: Transmit PDO object1 = 0x30000210。

2.1.16.4 Sub-Index:0x01 Transmit PDO object3 (UINT32， RO)

A. query Sub-Index 0x01 Transmit PDO object3

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x1A	0x03	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x00	0x1A	0x03	0x10	0x03	0x00	0x30

Note:Transmit PDO object1 = 0x30000310。

2.1.16.5 Sub-Index:0x01 Transmit PDO object4 (UINT32， RO)

A. query Sub-Index 0x01 Transmit PDO object4

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x1A	0x04	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x00	0x1A	0x04	0x10	0x04	0x00	0x30

Note:Transmit PDO object1 = 0x30000410。

2.1.17 OD Index (0x1F80): NMT startup (NMT start， UINT32)

NMT StartUP The setting determines whether the device is in the operation mode directly after power-on or in the pre-operation mode.

2.1.17.1 read NMT StartUP object

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x80	0x1F	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x80	0x1F	0x00	0x00	0x00	0x00	0x00

Note: The values are interpreted as follows：

0x00000000: The state of the NMT state machine after initialization is Pre-Operational.

0x00000008: The state of the NMT state machine after initialization is Operational.

2.1.17.2 Set up the NMT StartUP object

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x23	0x80	0x1F	0x00	0x08	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x80	0x1F	0x00	0x00	0x00	0x00	0x00

Note: NMT StartUp The object is set to: 0x00000008.

2.2 Manufacturer Specific Profile Area(Manufacturer specified protocol area)

2.2.1 OD Index (0x2100) baud rate (UINT8， RW)

The factory default baud rate is 50Kbps.

2.2.1.1 Modify the baud rate

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x2F	0x00	0x21	0x00	Baud	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x00	0x21	0x00	0x00	0x00	0x00	0x00

Note: The fifth byte (Baud) is 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08.

Send this command and receive the returned data. To make the changed baud rate effective, you must save the settings, and then power on or reset the node, and the new baud rate will be valid.

Baud Corresponding baud rate

Content (HEX)	Baud rate	Content (HEX)	Baud rate
01	20K	06	500k
02	50K	07	800k
03	100k	08	1M
04	125k		
05	250k		

2.2.1.2 Query the current baud rate

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x21	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x00	0x21	0x00	0x02	0x00	0x00	0x00

Note: The fifth byte (Baud) is 0x06, and the current baud rate is 50Kbps.

2.2.2 OD Index (0x2101) node address (UINT8， RW)

2.2.2.1 Modify node number

Node_ID=0x01～0x7F，The default node number (Node_ID) is 0x01

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x2F	0x01	0x21	0x00	New Node_ID	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x01	0x21	0x00	0x00	0x00	0x00	0x00

2.2.2.2 query node number

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x01	0x21	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x01	0x21	0x00	0x01	0x00	0x00	0x00

Note: The return node is 0x01.

2.2.3 OD Index (0x3000): Filterband Outputs (RECORD)

Expresses the ability to store relevant information provided by CANopen through read access.

Sub-index 0x00 Contains the most subindexes Number of Filterbands，UINT8。

Sub-index 0x01 refer to Filterband 1，INT16。

Sub-index 0x02 refer to Filterband 2，INT16。

Sub-index 0x03 refer to Filterband 3，INT16。

Sub-index 0x04 refer to Filterband 4，INT16。

2.2.3.1 Sub-Index:0x00(Number of sub-index, RO)

A. Query the number of sub-indexes

Its sub-index 0x00 has the value 0x04.

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x30	0x00	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4F	0x00	0x30	0x00	0x04	0x00	0x00	0x00

Note:Number of Filterbands = 0x04，The number of sub-indexes is 4.

2.2.3.2 Sub-Index:0x01 Filterband 1 (INT16， RO)

A. Query Sub-Index 0x01 Filterband 1

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x30	0x01	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4B	0x00	0x30	0x01	0x10	0x01	0x00	0x00

Note:Filterband 1 = 0x0101 = 257/1000 = 0.257*1g = 0.257g (resolution=1000, 1g)。

2.2.3.3 Sub-Index:0x02 Filterband 2 (INT16， RO)

A. Query Sub-Index 0x02 Filterband 2

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x30	0x02	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4B	0x00	0x30	0x02	0x10	0xFF	0x00	0x00

Note:Filterband 2 = 0xFF10 = -240/1000 = -0.24*0.1 = -0.024g (resolution=100, 0.1g)。

2.2.3.4 Sub-Index:0x03 Filterband 3 (INT16， RO)

A.Query Sub-Index 0x03 Filterband 3

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x30	0x03	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4B	0x00	0x30	0x03	0x10	0x7F	0x00	0x00

Note:Filterband 1 = 0x7F10 = 32528/1000 = 32.528 *0.01g = 0.32528g (resolution=10, 0.01g)。

2.2.3.5 Sub-Index:0x04 Filterband 4 (INT16， RO)

A. Query Sub-Index 0x04 Filterband 4

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x30	0x04	0x00	0x00	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4B	0x00	0x30	0x04	0x10	0x8F	0x00	0x00

Note:Filterband 1 = 0x8F10 = -28912/1000 = -28.912*0.001g = -0.028912g (resolution=1, 0.001g)。

2.3 Standardised Device Profile Area(Standardized protocol area)

2.3.1 OD Index(0x6000) Resolution resolution（UINT16， RW）

Value	Definition
1d	0.001g
10d	0.01g
100d	0.1g
1000d	1.0g
Other	Reserved

2.3.1.1 set resolution

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x2B	0x00	0x60	0x00	0xE8	0x03	0x00	0x00

SDO response message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x60	0x00	0x60	0x00	0x00	0x00	0x00	0x00

Note: The resolution is UINT16, double byte, respectively 1, 10, 100, 1000.

2.3.1.2 query resolution

SDO request message format

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x60	0x00	0x00	0x00	0x00	0x00

SDO response message format

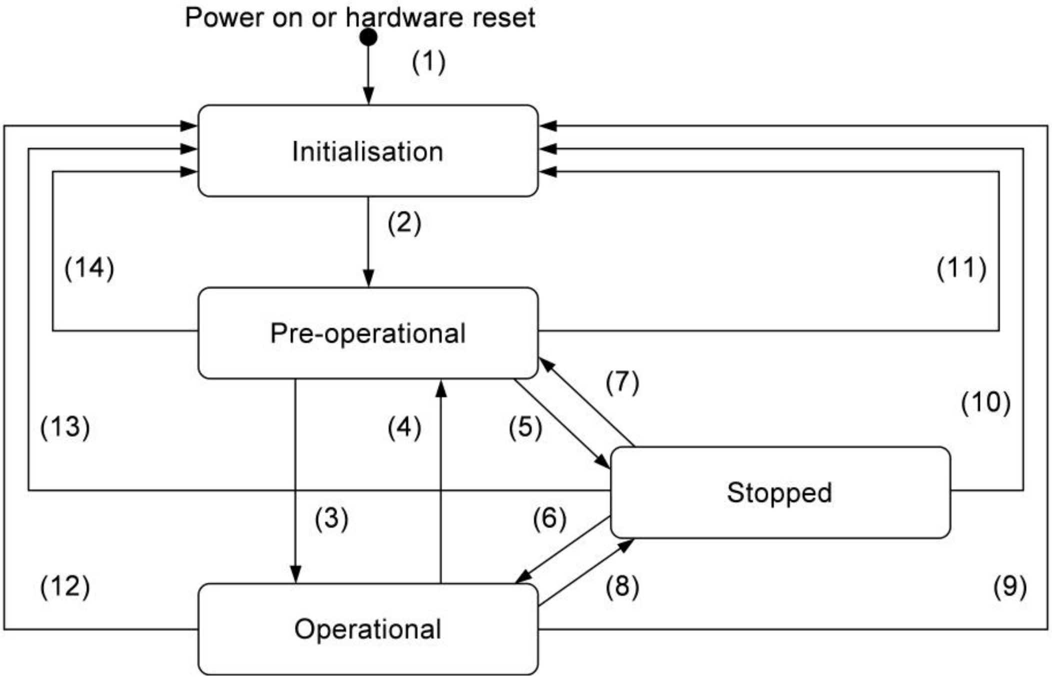
CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4B	0x00	0x60	0x00	0xE8	0x03	0x00	0x00

Note: The current resolution is 0x03E8 = 1000.

3. Network initialization and system Boot-Up

3.1 NMT state machine

After initialization, the CANopen device directly enters the configuration state (pre-operation mode). In this state, parameters can be configured for the CANopen device, and then the CANopen device can directly enter the running state. The figure below depicts the NMT state diagram for a CANopen device.



(1)	Power-on automatically enters the NMT initialization state
(2)	NMT initialization execution is completed-automatically enter the configuration state
(3)	Start remote nodes by NMT service or be controlled locally
(4) ， (7)	Enter configuration state by NMT service
(5) ， (8)	Stop remote node by NMT service
(6)	Start remote nodes by NMT service
(9) ， (10) ， (11)	Remote node reset by NMT service
(12) ， (13) ， (14)	Communication of remote nodes reset by NMT service

3.2 NMT state

3.2.1 NMT initialization state

CANopen enters the NMT initialization state after power-on or hardware reset, which mainly completes the reset of the CANopen device, mainly including reset application and reset communication, specifically including the parameters in the manufacturer protocol area and standard equipment protocol area being assigned power-on values, and the parameters in the communication protocol area being assigned Assign power-on value. Then, the CANopen device executes the Boot-up write service and enters the NMT configuration state (or called the pre-operation state).

Boot-Up The message is as follows:

CAN-ID	first byte
0x700+ Node_ID	0x00

3.2.2 NMT configuration state (pre-operational state)

In the configuration state, SDO communication is allowed, and PDO communication is not allowed. This state is usually used to configure PDO parameters and mapping objects (PDO mapping), etc. The CANopen device can start the remote node service by NMT or switch from this state to the running state through local control.

3.2.3 NMT running state

This state allows all communication services, transport PDOs, and access to data dictionaries via SDOs.

3.2.4 NMT stop state

The CANopen device enters the NMT stop state to terminate all communication services (except node guarding and heartbeat).

3.2.5 NMT Relationship between state and communication object

The following table specifies the relationship between the state and the communication object. The services in the table can only be executed when the CANopen device is in the appropriate state.

NMT status and communication object relationship

Service	Pre-Operational	Operational	Stopped
Process Data Object PDO	NO	YES	NO
Service Data Object SDO	YES	YES	NO
sync message	YES	YES	NO
Time Stamp	YES	YES	NO
Emergency message	YES	YES	NO
Network Management and Error Control	YES	YES	YES

3.3 NMT MODULE CONTROL（NMT module control）

Only the NMT-MASTER node can transmit the NMT Module Control message, no response is required

The NMT message format is as follows:

NMT-Master -->NMT-Slave

CAN-ID	1st byte	2nd byte
0x000	CS command word	Node_ID

Start and stop commands are sent by the master node (state transfer), but the dip angle does not respond

CAN-ID	1st byte	2nd byte	NMT service
00	01	00 (or Node_ID)	start the remote node
00	02	00 (or Node_ID)	stop remote node
00	80	00 (or Node_ID)	enter pre-operational state
00	81	00 (or Node_ID)	reset node
00	82	00 (or Node_ID)	reset communication

3.4 NMT Node Guarding

In the CANopen specification, the service and protocol of the monitoring device (error control) are used to detect whether the device in the network is online and the state of the device. Among them, the NMT command is confirmed in the application layer, and the CANopen network management system provides the following functions for equipment monitoring:

Heartbeat message: a message sent periodically to one or more devices, and the devices can monitor each other.

NMT slave monitoring (node protection): The NMT master periodically monitors the status of the slave through remote frames.

NMT master monitoring (life protection): Indirectly monitor the status of the NMT master through the received remote frames used to monitor the slave.

Note: Users can only use one of the two methods of heartbeat message or node/life protection to monitor the device. This product recommends using heartbeat packets.

3.4.1 heartbeat message

The device node can set the cycle to generate heartbeat messages, and the message format is as follows:

The heartbeat message is as follows:

CAN-ID	1st byte
0x700+ Node_ID	Bit7: 0 Bit6-0: status

The state table is as follows:

Status	Meaning
0x00	Boot-Up
0x04	Stopped
0x05	Operational
0X7F	Pre-Operational

The generation cycle time of the heartbeat message is set in the object dictionary 0X1017.

Note: Node guarding and heartbeat messages cannot be used at the same time.

3.4.2 NMT Slave monitoring (node guarding)

Nodes are kept sent by the master node, and through the node protection service, the NMT master node can check the status of each node.

NMT-Master sends remote frames (no data) as follows：

NMT-Master--->NMT-Slave

CAN-ID
0x700+ Node_ID

NMT-Slave The node response message is as follows:

CAN-ID	1st byte
0x700+ Node_ID	BIT7:toggle bit6-0: status

The data part is a byte, including a trigger bit Bit7. The trigger bit is set to "0" or 1 alternately every time the node protection responds, and the trigger bit is set to "0" at the first protection request. Bit 0 to bit 6 (Bit0~6) indicate the node protection status, currently only supports three statuses:

0X04: stop 0X05: operate 0X7F: pre-operate

3.5 PDO Data Object Protocol

3.5.1 OD Dictionary object 1000H Device Type (Device Type, UINT32) (read only)

Default value: 0X0004019A means: dual-axis resolution up to 32 bits.

The data is parsed as follows:

bit0-15: general information 410d (0x019A)

bit16-31: additional information 4d (0x0004)

1: Single axis	resolution	up to 16 bits
2: Dual-axis	resolution	up to 16 bits
3: Single axis	resolution	up to 32 bits
4: Dual-axis	resolution	up to 32 bits

The instruction to read object 1000H device type is as follows:

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

The read object 1000H device type inclination reply command is as follows:

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x43	0x00	0x10	0x00	0x9A	0x01	0x04	0x00

3.5.2 Object 6000H resolution (read only)

This dictionary defines the resolution of the acceleration

1	0.001g
10	0.01g
100	0.1g
1000	1g

The read object 6000H resolution command is as follows:

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x600+ Node_ID	0x40	0x00	0x60	0x00	0x00	0x00	0x00	0x00

The reading object 6000H resolution inclination reply command is as follows:

CAN-ID	1st byte	2nd byte	3rdbyte	4th byte	5th byte	6th byte	7th byte	8th byte
0x580+ Node_ID	0x4B	0x00	0x60	0x00	0xE8	0x03	0x00	0x00

3.5.3 TPOD map

OD Object Dictionary Index 0x1A00 Mapping Parameters

Object 0X1A00 First Transmis PDO		
Sub-index	Value	meaning
0	4	4 objects mapped to PDO
1	0X30000110	Object 0x30000, subindex 01, consists of 16 bits
2	0X30000210	Object 0x30000, subindex 02, consists of 16 bits
3	0X30000310	Object 0x30000, subindex 03, consists of 16 bits
4	0X30000410	Object 0x30000, subindex 04, consists of 16 bits

3.5.4 PDO data analysis

The PDO data consists of 4 parts according to the TPDO mapping, corresponding to the values of X, Y, Z, |X, Y|.

ID	LEN	D0	D1	D2	D3	D4	D5	D6	D7
0x180+ Node_ID	8	XL	XH	YL	YH	ZL	ZH	XYZL	XYZH

D0, D1: X-axis acceleration.

D2, D3: Y-axis acceleration.

D4, D5: |X, Y|-axis acceleration.

D6, D7: |X, Y|-axis acceleration.

XH_XL constitutes a signed 16-bit (INT16), with the low bit in front and the high bit in the back. The value is divided by 1000 and then multiplied by the resolution (the following resolution is 1g).

E.g.: XL XH : E8 03 = 0x03E8 = 1000/1000 = 1*1g = 1.000g
YL YH : 15 FC = 0xFC15 = -1003/1000 = -1.003*1g = -1.003g