GCAN-4038

CANopen Analog Input Data Acquisition Module

User Manual



Document Version: V2.21 (2024/06/13)

Revision History

Version	Date	Reason
V1.00	2020/06/16	Document created
V1.10	2020/07/20	Corrected some parameters
V2.00	2020/08/25	Adjusted document structure
V2.10	2022/05/26	Modified wiring pin descriptions
V2.20	2022/07/05	Document revision
V2.21	2024/06/13	Updated contact information

Contents

Revision History 2
Contents 3
1. Function Overview 5
1.1 General Description 5
1.2 Features 5
1.3 Typical Applications 5
1.4 Block Diagram 6
2. Device Installation 6
2.1 Device Dimensions 6
2.2 Interface Definitions and Functions 6
3. Device Usage 9
3.1 Power Connection 9
3.2 Connection to CAN-bus 9
3.3 CAN-bus Terminating Resistor 10
3.4 System Status Indicators 10
4. Configuration 11
4.1 CAN Node ID Configuration 11
4.2 CAN Baud Rate Configuration 11
5. Communication Message Format 12
5.1 Node State 12
5.2 NMT Command 12
5.3 PDO Command 13
5.4 SDO Command 14
6. GCAN-4038 Object Dictionary 16
7. Technical Specifications 18
Appendix A: CANopen Protocol Introduction 18
A.1 Terms and Writing Rules 18
A.2 Predefined CAN Identifiers 19
A.3 CANopen Object Dictionary 19
A.4 CANopen Communication 20
A.5 CANopen Network Configuration 23
8. Disclaimer 24
1. Copyright and Translation Notice 24

2. Product Information and User Responsibility 24
3. Interpretation 24

1. Function Overview

1.1 General Description

The GCAN-4038 is an industrial-grade CANopen analog input module integrating one standard CANopen interface and eight analog acquisition channels. With the GCAN-4038, users can acquire analog signals and connect remote analog acquisition devices to a CANopen network, forming a control node for analog signal acquisition in the network.

The GCAN-4038 is a key tool for analog signal acquisition. It is compact and easy to wire. Its bus interface integrates isolation protection to prevent damage from transient overvoltage or overcurrent. The DIN-rail mounting design makes it easy to integrate into various control cabinets.

1.2 Features

- The CAN-bus supports CAN 2.0A and CAN 2.0B frame formats and complies with ISO/DIS 11898.
- CAN-bus communication baud rate supports 8 standard rates.
- The CAN-bus interface uses electrical isolation; isolation insulation voltage: 1500 V DC.
- Powered by 9~30 V DC, 140 mA (quiescent current: 40 mA).
- Number of analog input channels: 8.
- Analog input signals: -5 V~+5 V, -10 V~+10 V, -20 mA~+20 mA.
- AI and CANopen interfaces use terminal blocks.
- Baud rate and node ID are configured with DIP switches.
- Can be mounted on a DIN rail with the supplied rail-mounting connector.
- Operating temperature range: -40 °C ~ +85 °C.

1.3 Typical Applications

- Industrial field network data monitoring.
- Remote communication in coal mines and oil wells.
- CAN teaching applications and remote communication.
- CAN industrial automation control systems.
- Intelligent building control data broadcasting systems and other CAN-bus applications.

1.4 Block Diagram

The GCAN-4038 consists of eight isolated analog inputs, a power supply circuit, an MCU minimum system, and a CAN communication circuit. The block diagram is shown in Figure 1.1.

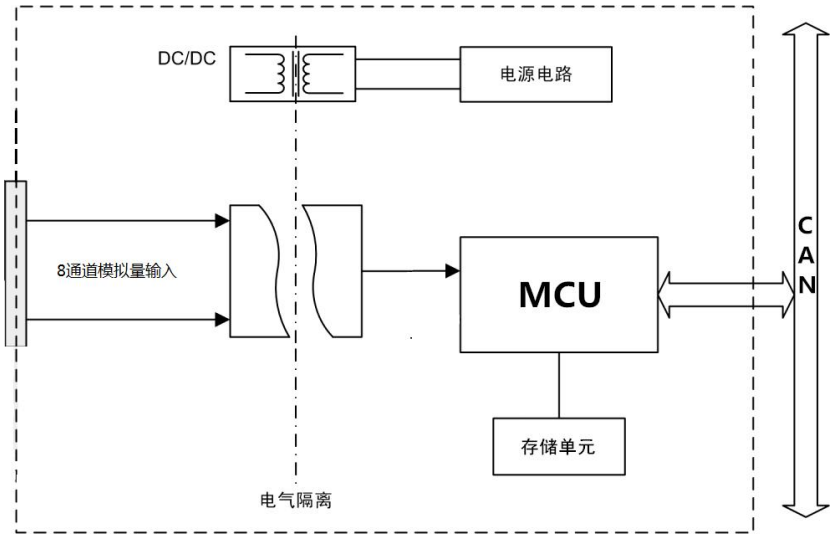


Figure 1.1 GCAN-4038 block diagram

2. Device Installation

2.1 Device Dimensions

The outline dimensions of the device are (length, including terminal block) 122 mm x (width) 70 mm x (height) 25 mm.

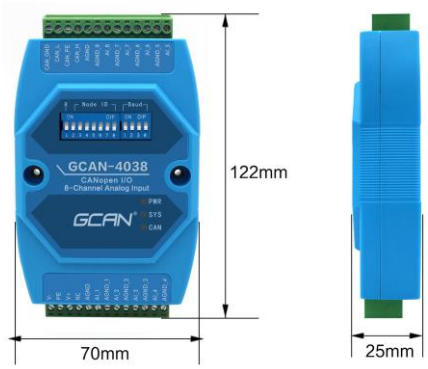


Figure 2.1 Device outline dimensions

2.2 Interface Definitions and Functions

The GCAN-4038 integrates one 9-30 V DC power interface, one standard CANopen interface, and eight analog differential signal input interfaces (each an independent input). The positions of the power interface and four analog differential input interfaces are shown in Figure 2.2; the interface definitions are listed in Table 2.1.



Figure 2.2 Power and four analog input interface positions

Pin (L-to-R)	Port	Name	Function
1	Power	V-	9-30 V DC power negative
2	PE	-	Protective earth
3		V+	9-30 V DC power positive
4		NC	Not connected
5		AGND	Common terminal for single-ended signal acquisition (not required for 2-wire differential signals)
6	Analog input	AI1	Analog differential acquisition channel 1 positive
7		AGND_1	Analog differential acquisition channel 1 negative
8		AI2	Analog differential acquisition channel 2 positive
9		AGND_2	Analog differential acquisition channel 2 negative
10		AI3	Analog differential acquisition channel 3 positive
11		AGND_3	Analog differential acquisition channel 3 negative
12		AI4	Analog differential acquisition channel 4 positive
13		AGND_4	Analog differential acquisition channel 4 negative

Table 2.1 Power and eight analog input interface definitions

The CAN-bus channel of the GCAN-4038 is led out through one 4-pin pluggable terminal block, which can connect to a CAN-bus network or a CAN-bus interface device. The positions of the CAN-bus interface and the other four analog differential input interfaces are shown in Figure 2.3; the interface definitions are listed in Table 2.2.



Figure 2.3 CAN-bus interface and four analog input interface positions

Pin (L-to-R)	Port	Name	Function
1	Analog input	AI5	Analog differential acquisition channel 5 positive
2		AGND_5	Analog differential acquisition channel 5 negative
3		AI6	Analog differential acquisition channel 6 positive
4		AGND_6	Analog differential acquisition channel 6 negative
5		AI7	Analog differential acquisition channel 7 positive
6		AGND_7	Analog differential acquisition channel 7 negative
7		AI8	Analog differential acquisition channel 8 positive
8		AGND_8	Analog differential acquisition channel 8 negative
9		AGND	Common terminal for single-ended signal acquisition (not required for 2-wire differential signals)
10	CAN	CAN_H	CAN_H signal line (CAN high)
11		CAN_PE	CAN-bus shield
12		CAN_L	CAN_L signal line (CAN low)
13		CAN_GND	CAN-bus ground

Table 2.2 CAN-bus interface and four analog input interface definitions

The DIP switches of the GCAN-4038 are shown in Figure 2.4. The leftmost switch is the reset button, by default in the lower position; it is a debug interface and must not be toggled. The seven positions to the right of the reset button set the node ID. The rightmost four positions set the baud rate. The left side is the high bit and the right side is the low bit.

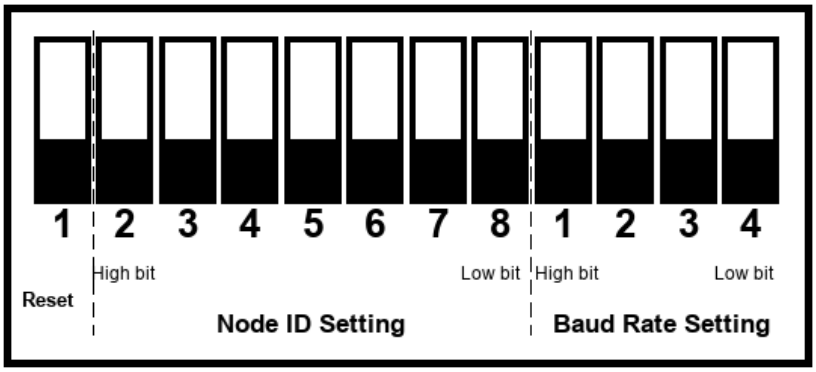


Figure 2.4 GCAN-4038 DIP switches

3. Device Usage

3.1 Power Connection

The GCAN-4038 supports the common +9-30 V DC industrial power supply. For reliable operation, a +12 V or +24 V regulated DC power supply is recommended.

The power input provides reverse-polarity protection, preventing damage caused by incorrect power wiring.

3.2 Connection to CAN-bus

Connect the GCAN-4038 to the CAN bus by linking CAN_H to CAN_H and CAN_L to CAN_L to establish communication.

The CAN-bus network uses a linear topology. The two most distant ends of the bus require 120-ohm terminating resistors; if there are more than two nodes, intermediate nodes do not need 120-ohm terminating resistors. Branch connections must not exceed 3 m. The CAN-bus connection is shown in Figure 3.1.

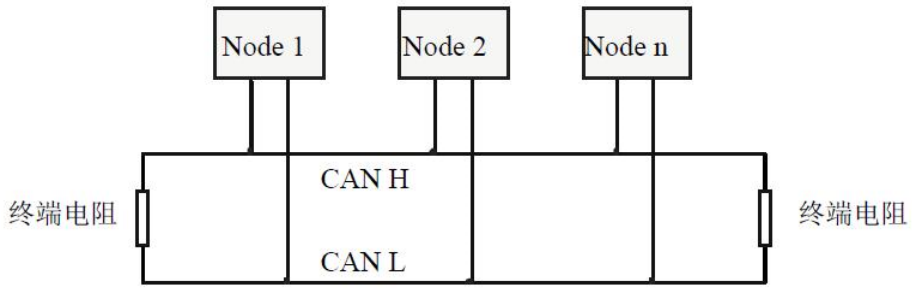


Figure 3.1 CAN-bus network topology

Note: CAN-bus cable may be ordinary twisted pair or shielded twisted pair. The theoretical maximum communication distance depends mainly on the bus baud rate; see Table 3.1. If the communication distance exceeds 1 km, ensure the wire cross-section is greater than 1.0 mm²; the specific gauge depends on the distance and is generally increased as the distance grows.

Baud rate	Bus length
1 Mbit/s	25 m

800 kbit/s	100 m
500 kbit/s	250 m
250 kbit/s	500 m
125 kbit/s	1.0 km
100 kbit/s	2.5 km
50 kbit/s	5.0 km
20 kbit/s	13 km

Table 3.1 Baud rate vs. maximum bus length

3.3 CAN-bus Terminating Resistor

To enhance CAN communication reliability and eliminate signal reflection at the bus ends, terminating resistors are normally added at the two most distant ends of the CAN-bus network, as shown in Figure 3.2. The value of the terminating resistor is determined by the characteristic impedance of the transmission cable. For example, twisted pair has a characteristic impedance of 120 Ω, so 120 Ω terminating resistors should be integrated at both bus ends. If other nodes on the network use different transceivers, the terminating resistance must be recalculated.

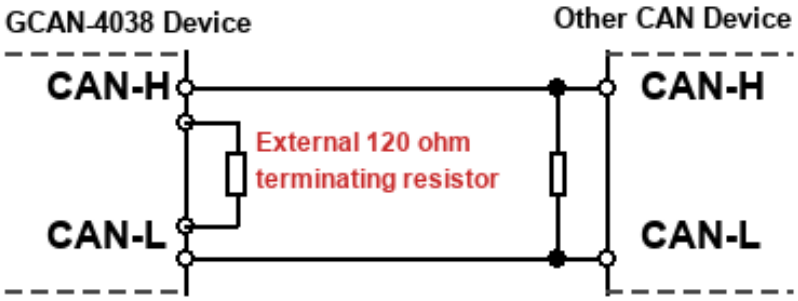


Figure 3.2 GCAN-4038 connected to another CAN node device

Note: The GCAN-4038 does not integrate a 120 Ω terminating resistor. If there are more than two nodes, intermediate nodes do not need 120 Ω terminating resistors. When required, connect the resistor across CAN_H and CAN_L, as shown in Figure 3.2.

3.4 System Status Indicators

The GCAN-4038 has one PWR indicator, one SYS indicator, and one CAN indicator to show the operating state of the device. Their functions and states are listed in Table 3.2.

Indicator	State	Indication
PWR	On	Power supply normal
	Off	Power supply failure
SYS	Flashing	Device initialization passed, in standby state
CAN	Green flashing	CAN interface data transmission
	Red	CAN interface data transmission error

Table 3.2 GCAN-4038 indicator status

- When the GCAN-4038 is powered on, the PWR, SYS and CAN indicators light up immediately.
- After the self-test completes, the SYS indicator flashes.
- When data is transmitted at the CAN end, the CAN indicator flashes green.
- If a CAN-bus communication error occurs, the CAN indicator turns red.

4. Configuration

Users can set the parameters of the GCAN-4038 with the DIP switches on top of the module to meet practical application requirements. The configuration includes the node ID and the baud rate.

Note: After changing the GCAN-4038 parameters with the DIP switches, power-cycle the module for the new parameters to take effect.

4.1 CAN Node ID Configuration

As shown in Figure 4.1, when a DIP switch position is set to ON it represents 1, and when set to OFF it represents 0. Positions 2-8 of the DIP switch set the node ID (NODE ID): position 8 is the least significant bit and position 2 is the most significant bit. The node ID is the sum of the decimal values of the corresponding binary bits. The valid node ID range is 0~127.

Note: DIP switch position 1 is the configuration switch, by default OFF (lower position); do not toggle it.

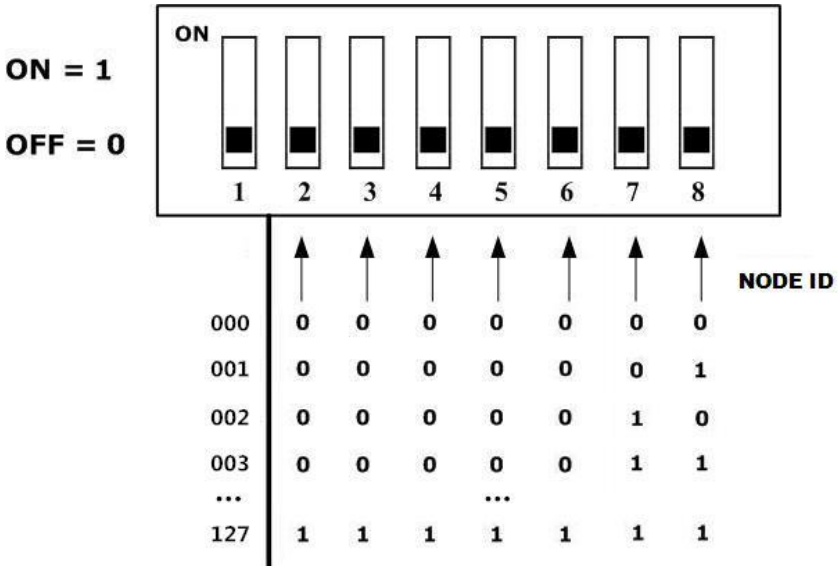
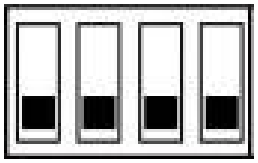


Figure 4.1 CAN node ID configuration

4.2 CAN Baud Rate Configuration

Positions 1-4 of the right-hand DIP switch set the baud rate: position 1 is the low bit and position 4 is the high bit. The baud rate corresponding to each switch setting is listed in Figure 4.2.

The DIP switch settings for each baud rate are illustrated below.



1000k	800k	500k	250k	125k	100k	50k	20k	10k

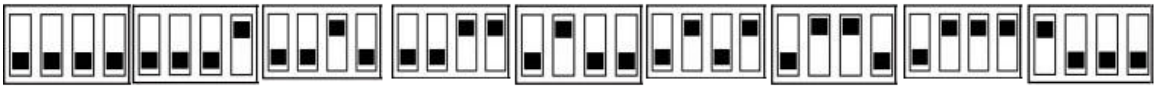


Figure 4.2 CAN baud rate configuration

5. Communication Message Format

This chapter uses the Guangcheng USBCAN-II Pro module and ECANTools software to receive and transmit CAN-bus data. The CANopen master function included in the software makes it very convenient to debug CANopen slaves.

The Guangcheng USBCAN bus analyzer can simulate CAN-bus communication devices for communication tests of the GCAN-4038. It is an intuitive CAN-bus debugging and analysis tool through which CAN data transmission and reception can be monitored and simulated on a PC; it is an essential tool for CAN-bus engineers.

5.1 Node State

The GCAN-4038 complies with the standard CANopen CiA 301 protocol and is a standard CANopen slave device. After startup, it actively sends data to the master with frame ID 0x700 + Node ID. The frame data is 0x7F, with a default cycle time of 1 s (object dictionary 0x100C, guard time), indicating that the node is in the pre-operational state.

For example, if the Node ID of the GCAN-4038 is set to 1 via the DIP switches (the node IDs below are all 1), the USBCAN-II Pro as master receives a node-state message with frame ID 0x701 and frame data 0x7F. Using ECANTools, this message can be received as shown in Figure 5.1.

Frame ID (HEX)	DLC	Frame data (HEX)
701	1	7F

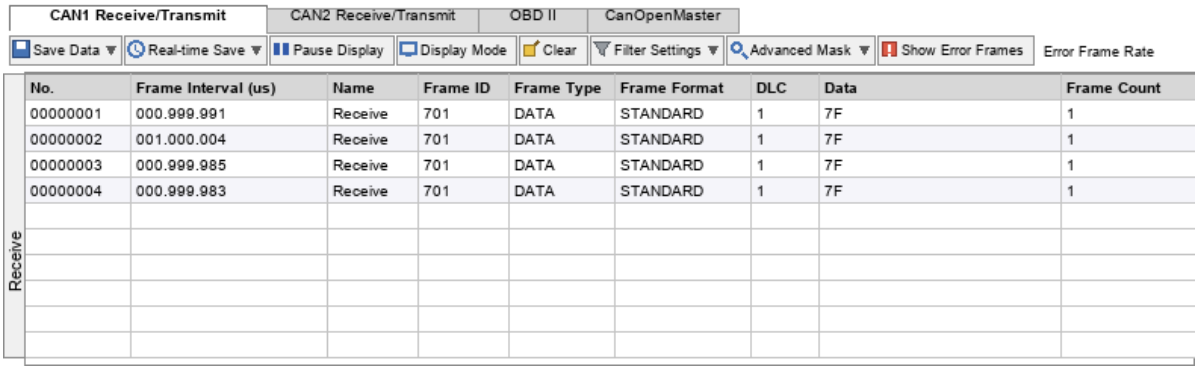


Figure 5.1 Startup command monitor interface

5.2 NMT Command

The GCAN-4038 receives operation commands from the master with frame ID 0x000 and DLC 2. The first byte of the frame data is the command specifier and the second byte is the node ID (00 controls all nodes).

For example, if the Node ID is 1 and the GCAN-4038 is commanded to enter the operational state (01), the NMT command frame ID is 0x000 and the frame data is 0x01, 0x01. This command can also be sent by the CANopenMaster plug-in of ECANTools.

Frame ID (HEX)	DLC	Frame data (HEX)
000	2	01 01 (command specifier, node address)

The steps to command the GCAN-4038 into the operational state with ECANTools are: (1) enter the slave node ID in the text box after ID; (2) click Add Slave; (3) expand the + sign to the left of the node and click NMT; (4) select "Start Remote Node" (default) and click the NMT

button below. As shown in Figure 5.3, after sending, the NMT command can be viewed in the ECANTools monitor interface, and the node state changes to 0x05 (operational).

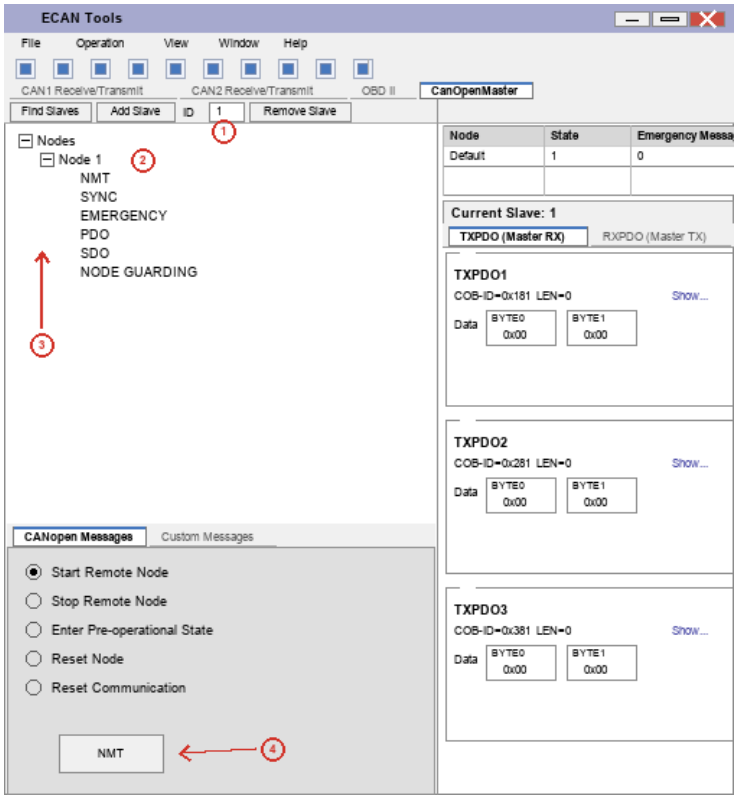


Figure 5.2 NMT command sending interface

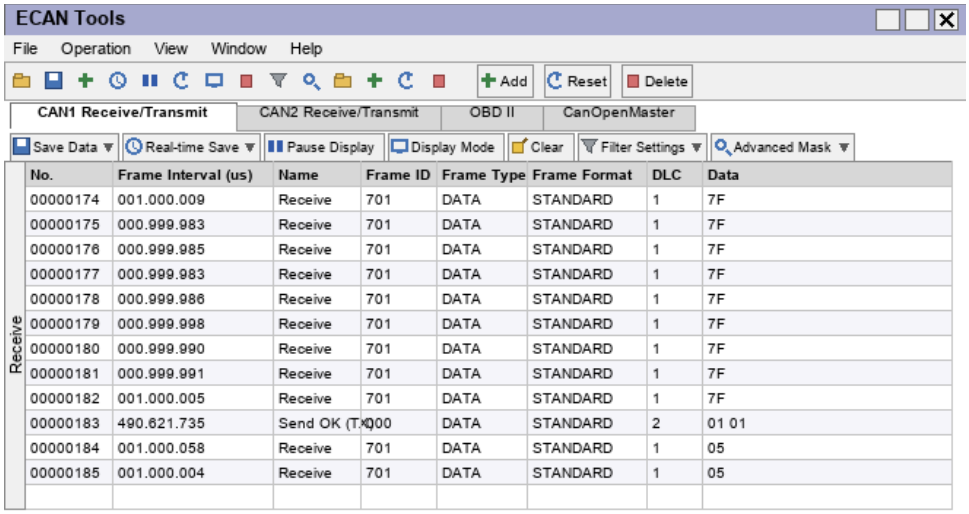


Figure 5.3 NMT command monitor interface

5.3 PDO Command

The GCAN-4038 uses PDO (Process Data Object) to acquire analog signals. TPDO indicates the current analog acquisition state; there are two TPDOs. TPDO1 represents the first four channels and TPDO2 the last four. Each channel of the analog acquisition state is represented by two bytes: in TPDO1, bytes 1-2 represent channel 1, bytes 3-4 channel 2, bytes 5-6 channel 3, and bytes 7-8 channel 4; in TPDO2, bytes 1-2 represent channel 5, bytes 3-4 channel 6, bytes 5-6 channel 7, and bytes 7-8 channel 8. Three analog acquisition ranges are supported: -5 V~+5 V, -10 V~+10 V, and -20 mA~+20 mA, but not simultaneously. The supported range is set via SDO (explained in 5.4); the default is -10 V~+10 V.

For example, set the node ID of the GCAN-4038 to 1. If the port range is the default -10 V~+10 V and channel 1 receives 5 V, the corresponding digital value is 16383, i.e. 3FFF in hexadecimal. Because the low byte is transmitted first and the high byte last, the data is FF 3F. If all other channels have no input, the TPDO1 frame ID is 0x181, DLC is 8, and frame data is FF 3F 00 00 00 00 00 00. The TPDO2 frame ID is 0x281, DLC is 8, and frame data is 00 00 00 00 00 00 00 00.

As shown in Figure 5.4, after sending the startup command, 5 V is applied to the first input and DC 3F is displayed. There is a small error, but it is within the allowable range. TPDO is sent cyclically with a period of 1 ms.

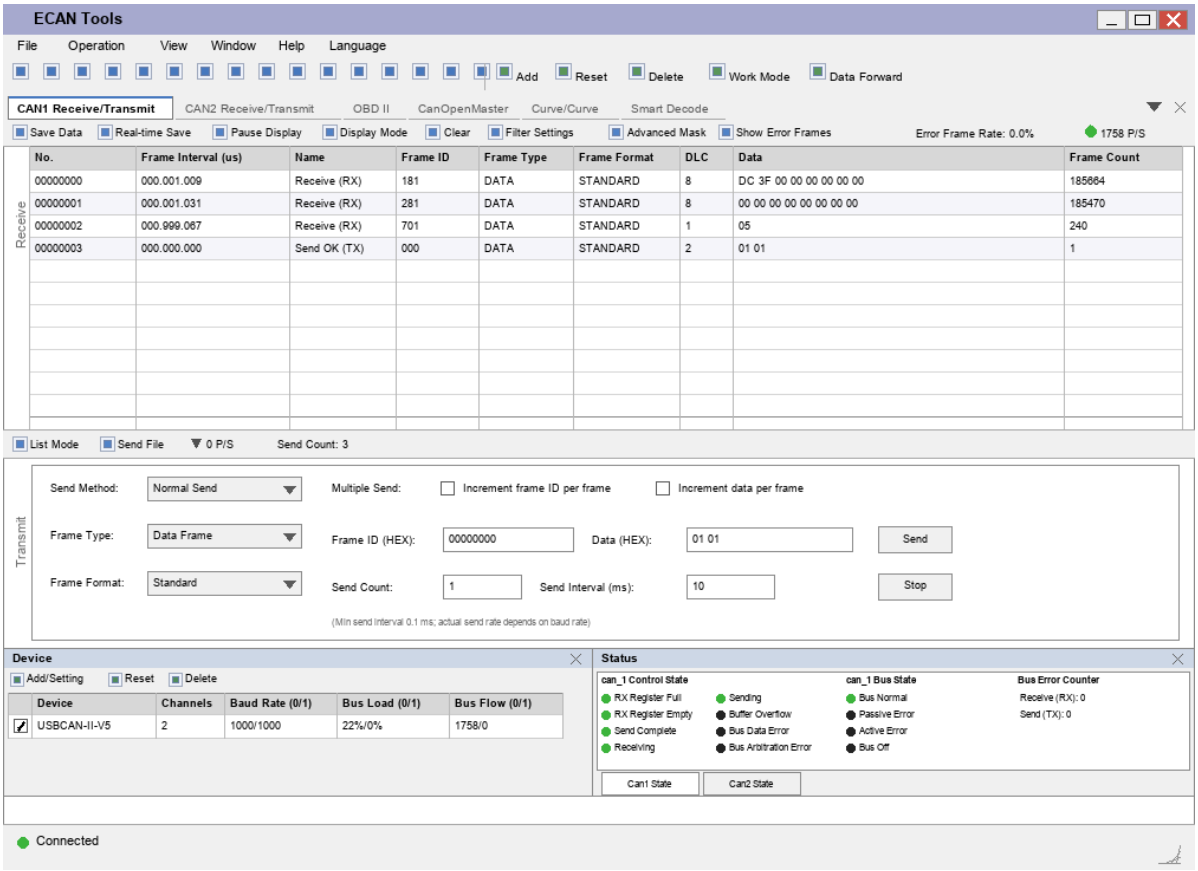


Figure 5.4 PDO command receive interface

Frame ID (HEX)	DLC	Frame data (HEX)
181	8	AI_1 AI_2 AI_3 AI_4 = FF 3F 00 00 00 00 00 00
281	8	AI_5 AI_6 AI_7 AI_8 = 00 00 00 00 00 00 00 00

5.4 SDO Command

The GCAN-4038 can write parameters via SDO commands to change the analog input type. This parameter is at object dictionary index 0x2400, subindex 0x00, data type Unsigned8. Data 0x02 and 0x07 correspond to -10 V~+10 V, 0x00 to -5 V~+5 V, and 0x05 to -20 mA~+20 mA. The default is -10 V~+10 V.

SDO commands must be sent while the GCAN-4038 is in the pre-operational state (0x7F). The steps to change the default -10 V~+10 V to -5 V~+5 V are: (1) if the GCAN-4038 is not in the pre-operational state, power-cycle it, open the CANopenMaster plug-in of ECANTools, click NMT, select Enter Pre-operational State, then click the NMT button; (2) in the CANopenMaster interface, click SDO on the left; (3) click SDO Download (Write Parameter); (4) enter 0x2400,

0x00, Unsigned8, 0x00 and Hexadecimal in the index, subindex and other fields as shown in Figure 5.7, then click Send. If the SDO response area below receives the data shown in Figure 5.7, communication is normal.

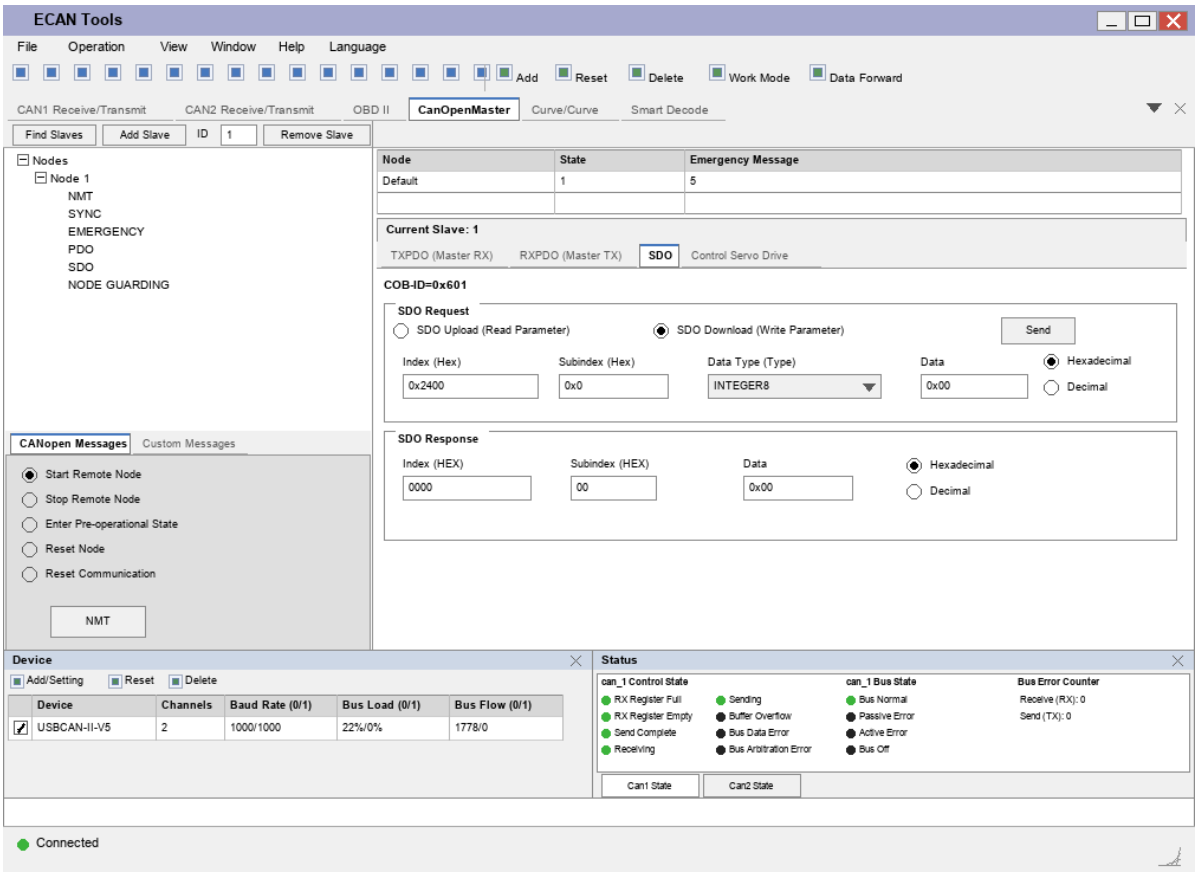


Figure 5.7 SDO command sending interface A

After sending the SDO command, click NMT in the CANopenMaster plug-in to switch to the NMT interface. Select Start Remote Node and click the NMT button to put the device into the operational state. ECANTools will then receive the TPDO sent by the GCAN-4038, as shown in Figure 5.9.

To change to -20 mA~+20 mA, simply change the data 0x00 to 0x05. Note that the -10 V~+10 V and -5 V~+5 V modes do not require a jumper cap, while the -20 mA~+20 mA mode requires the jumper cap on the circuit board, as shown in Figure 5.8. You may also contact customer service before purchase to confirm the analog acquisition type in advance, and we will install the jumper before shipment. (Note: the change takes effect after power-cycling.)

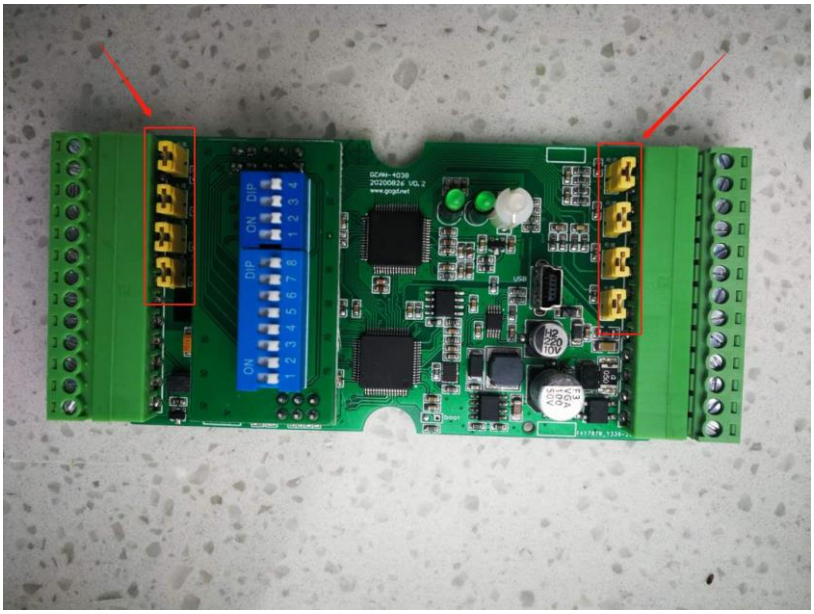


Figure 5.8 Jumper cap installed

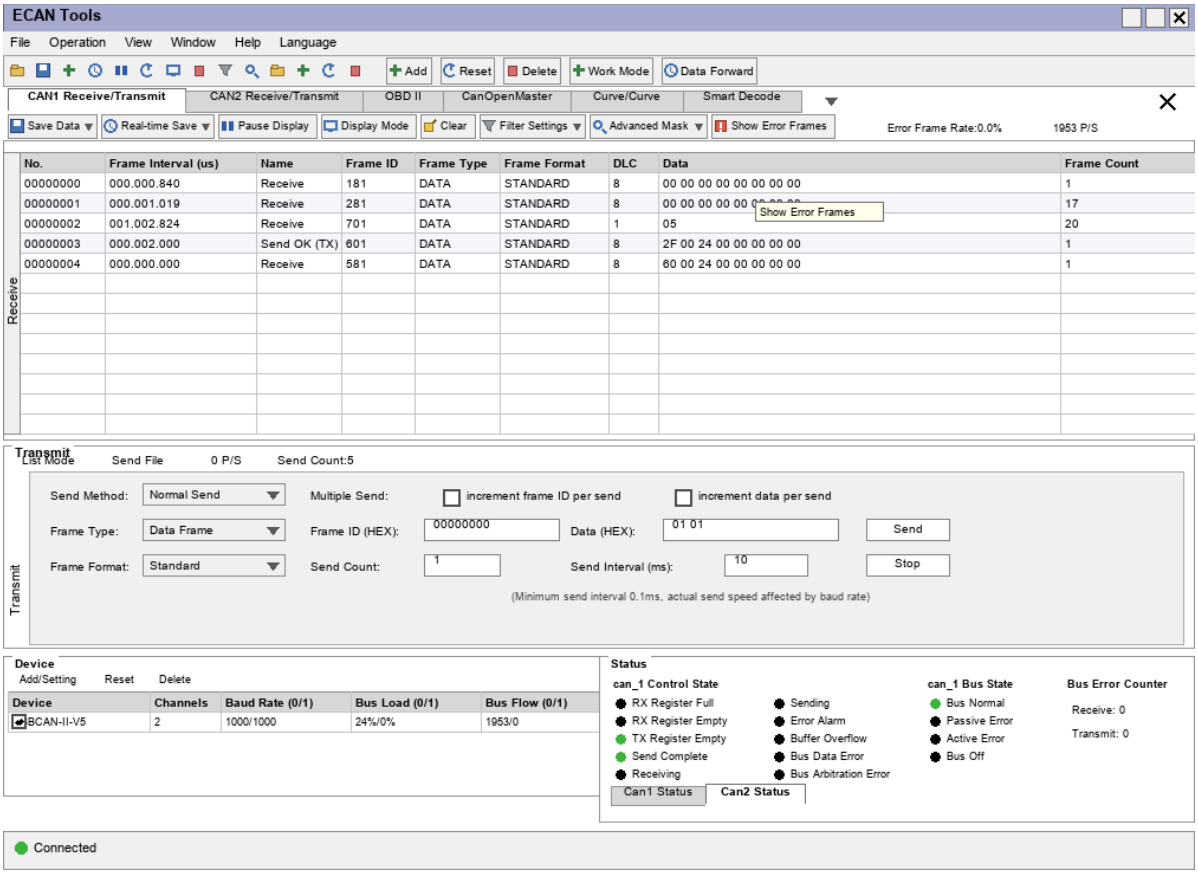


Figure 5.9 SDO command monitor interface

6. GCAN-4038 Object Dictionary

Index	Sub	Name	Type	Attr.	Default	Description
Communication profile area						
0x1000	-	Device Type	UINT32	RO	0x000A0011	Device type
0x1001	-	Error Register	UINT8	RO	0	Current error type
0x1003	0	number of errors	UINT8	RW	0	-
	1~4	standard error field	UINT32	RO	0	Historical emergency error code
0x1005	-	COB-ID SYNC	UINT32	RW	0x00000080	-

0x1006	-	Communication Cycle Period	UINT16	RW	0x2710	Communication cycle period
0x1007	-	Sync Window Length	UINT32	RW	0	-
0x1008	-	GCAN-4038 Name	STRING	CONST	GCAN-4028	Device name
0x1009	-	GCAN4038 Hardware Version	STRING	CONST	V1.01	Hardware version
0x100A	-	GCAN4038 Software Version	STRING	CONST	V5.18	Software version
0x100C	-	Guard Time	UINT16	RW	0x03E8	Node guard time, default 1 s
0x100D	-	Life Time Factor	UINT8	RW	0x03	Node guard factor
0x1010	0	largest supported Sub-Index	UINT8	RO	1	Store parameters
	1	save all parameters	UINT32	RO	0	-
0x1011	0	largest supported Sub-Index	UINT8	RO	1	Restore default parameters
	1	save all parameters	UINT32	RO	0	-
0x1014	-	COB-ID Emergency message	UINT32	RW	$NODEID+0x00000080	COB-ID emergency message
0x1016	0	Number Of Entries	UINT8	RO	1	-
	1	Consumer Heartbeat Time #1	UINT32	RW	0	-
0x1017	-	Producer Heartbeat Time	UINT16	RW	0	-
0x1018	0	number of Entries	UINT8	RO	0x04	-
	1	Vendor-ID	UINT32	RO	0x00000449	Manufacturer code assigned by CiA
	2	Product code	UINT32	RO	0x04038	Product code
	3	Revision number	UINT32	RO	0x00000000	Revision number
	4	Serial number	UINT32	RO	0x00000001	Serial number
TPDO communication parameters						
0x1800	0	largest subindex supported	UINT8	RO	5	-
	1	COB-ID used	UINT32	RW	NODEID+0x80000180	COB-ID used by TPDO
	2	transmission type	UINT8	RW	0xFF (255)	0xFF trigger mode, 0xFE cyclic, default 1 ms
	3	inhibit time	UINT16	RW	0x64	Inhibit time
	5	event timer	UINT16	RW	0x01	Event timer
0x1801	0	largest subindex supported	UINT8	RO	5	-
	1	COB-ID used	UINT32	RW	NODEID+0x80000180	COB-ID used by TPDO
	2	transmission type	UINT8	RW	0xFF (255)	0xFF trigger mode, 0xFE cyclic, default 1 ms
	3	inhibit time	UINT16	RW	0x64	Inhibit time
	5	event timer	UINT16	RW	0x01	Event timer
0x1A00	0	number of mapped objects	UINT8	RO	4	-
	1	PDO mapping 1. app. object	UINT32	RW	0x20100110	Map AI1 to TXPDO1
	2	PDO mapping 2. app. object	UINT32	RW	0x20100210	Map AI2 to TXPDO1
	3	PDO mapping 3. app. object	UINT32	RW	0x20100310	Map AI3 to TXPDO1
	4	PDO mapping 4. app. object	UINT32	RW	0x20100410	Map AI4 to TXPDO1
0x1A01	0	number of mapped objects	UINT8	RO	4	-
	1	PDO mapping 1. app. object	UINT32	RW	0x20100510	Map AI5 to TXPDO2
	2	PDO mapping 2. app. object	UINT32	RW	0x20100610	Map AI6 to TXPDO2
	3	PDO mapping 3. app. object	UINT32	RW	0x20100710	Map AI7 to TXPDO2
	4	PDO mapping 4. app. object	UINT32	RW	0x20100810	Map AI8 to TXPDO2
Device status						
0x2010	0	Number Of Entries	UINT8	const	0x8	-
	1	Application data Input 16bit #0	INT16	RO	0x0	Data acquired by channel 1
	2	Application data Input 16bit #1	INT16	RO	0x0	Data acquired by channel 2
	3	Application data Input 16bit #2	INT16	RO	0x0	Data acquired by channel 3
	4	Application data Input 16bit #3	INT16	RO	0x0	Data acquired by channel 4
	5	Application data Input 16bit #4	INT16	RO	0x0	Data acquired by channel 5
	6	Application data Input 16bit #5	INT16	RO	0x0	Data acquired by channel 6
	7	Application data Input 16bit #6	INT16	RO	0x0	Data acquired by channel 7
	8	Application data Input 16bit #7	INT16	RO	0x0	Data acquired by channel 8
0x2400	-	Input type	UINT8	rwr	0	Set input mode
0x2401	-	SetOS	UINT8	rwr	0	Set oversampling mode
0x2402	-	Range	UINT8	RO	0	Maximum input voltage
0x2403	-	NOW Type	UINT8	RO	0	Current input mode
0x2404	-	ModleStatus	UINT8	RO	0	Reserved
0x2405	-	ModleStatus	UINT8	RO	0	Reserved
0x2406	-	ModleStatus	UINT8	RO	0	Reserved

0x2407	-	ModleStatus	UINT8	RO	0	Reserved

7. Technical Specifications

Category	Item	Specification
Connection	DI/DO interface	Terminal block
	CAN interface	Terminal block
Interface features	CAN interface	Compliant with ISO 11898, supports standard CANopen protocol and CAN 2.0A/B
	CAN baud rate	1000K, 500K, 250K, 200K, 125K, 100K, 50K, 20K, 10K
	Electrical isolation	1500 V DC
	CAN terminating resistor	Not integrated; add between CAN_H and CAN_L if required
Power supply	Supply voltage	+9~30 V DC
	Supply current	Max. 140 mA (quiescent current: 40 mA)
Environmental	Operating temperature	-40 °C ~ +85 °C
	Operating humidity	15%~90% RH, non-condensing
	EMC test	EN 55024:2011-09, EN 55022:2011-12
	Protection class	IP 20
General	Dimensions	121 mm x 70 mm x 26 mm
	Weight	110 g

Appendix A: CANopen Protocol Introduction

The CANopen protocol was developed in the late 1990s by CiA (CAN-in-Automation) based on CAL (CAN Application Layer) and quickly gained wide recognition and use in Europe. Through several revisions of the CANopen specification, its stability, real-time performance and anti-interference capability have been further improved. CiA continues to release device sub-protocols for various industries, promoting faster development and adoption of CANopen. Today CANopen is widely used in motion control, vehicles, motor drives, construction machinery, shipping and other industries.

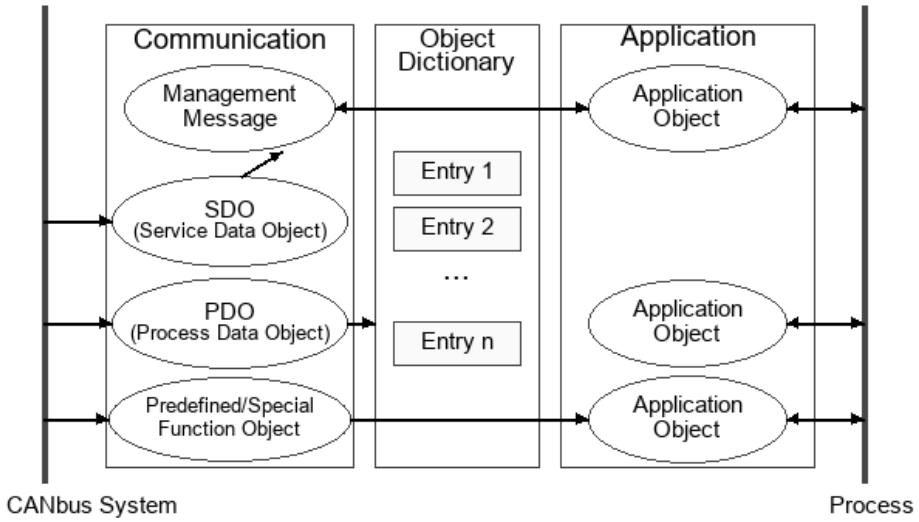


Figure A.1 CANopen device structure

Figure A.1 shows the structure of a CANopen device. The CANopen protocol is generally divided into three parts: user application layer, object dictionary, and communication.

A.1 Terms and Writing Rules

1. Terms:

- PDO: Process Data Object.
- TPDO: Transmit Process Data Object.
- RPDO: Receive Process Data Object.
- SDO: Service Data Object.
- NMT: Network Management.
- SYNC: Synchronization Objects.
- EMCY: Emergency Objects.
- OD: Object Dictionary.
- EDS: Electronic Data Sheet.
- CAN-ID: Controller Area Network Identifier.
- COB-ID: Communication Object Identifier.
- SSDO: Servers Service Data Object.
- DS: Draft Standard.

2. Writing rules:

In this manual, object dictionary indices and subindices follow the rule shown in Figure A.2: the index is expressed in hexadecimal, the subindex in decimal, and the two are separated by a space.

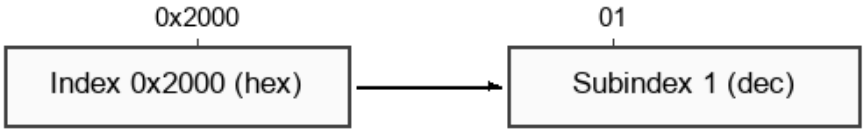


Figure A.2 Index/subindex writing rule

A.2 Predefined CAN Identifiers

Object	Function code	CAN-ID range
NMT network management command	0000b	000h
Sync message	0001b	080h
Time Stamp message	0010b	100h
Emergency message	0001b	081h-0FFh
TPDO1 transmit process data object 1	0011b	181h-1FFh
RPDO1 receive process data object 1	0100b	201h-27Fh
TPDO2 transmit process data object 2	0101b	281h-2FFh
RPDO2 receive process data object 2	0110b	301h-37Fh
TPDO3 transmit process data object 3	0111b	381h-3FFh
RPDO3 receive process data object 3	1000b	401h-47Fh
TPDO4 transmit process data object 4	1001b	481h-4FFh
RPDO4 receive process data object 4	1010b	501h-57Fh
SDO Server-to-Client (response)	1011b	581h-5FFh
SDO Client-to-Server (request)	1100b	601h-67Fh
NMT error control	1110b	701h-77Fh

A.3 CANopen Object Dictionary

The CANopen Object Dictionary (OD) is the most core concept of the CANopen protocol. It is an ordered group of objects, each addressed by a 16-bit index (valid range 0x1000 to 0x9FFF). To allow access to individual elements in the data structure, an 8-bit subindex is also defined. Every CANopen device has an object dictionary containing all parameters that describe the device and its network behavior. These parameters are usually recorded in an Electronic Data Sheet (EDS) rather than on paper. For the master node in a CANopen network, it is not necessary to access every object dictionary entry of a slave.

Items in the CANopen object dictionary are described by a series of sub-protocols. Each sub-protocol describes the function, name, index, subindex, data type, whether the object is

mandatory, and access attributes, ensuring compatibility between devices of the same type from different manufacturers. The core descriptive sub-protocol is DS301, which covers the application layer and communication structure; other sub-protocols supplement and extend DS301. The main types are:

- Communication Profile: describes the main form of the object dictionary and the communication objects and parameters; applies to all CANopen devices, index range 0x1000~0x1FFF.
- Manufacturer-specific Profile: for special functions not defined in device profiles, manufacturers can define object dictionary objects here; the same index may differ between manufacturers, index range 0x2000~0x5FFF.
- Device Profile: defines object dictionary objects for different device types, e.g. DS401, DS402, DS406; index range 0x6000~0x9FFF.

A.4 CANopen Communication

The CANopen protocol mainly defines four types of objects: Network Management (NMT), Service Data Object (SDO), Process Data Object (PDO), and predefined or special-function objects.

1. Network Management (NMT): management messages handle layer management, network management and ID allocation, e.g. initialization, configuration and network management (including node guarding). In network management, only one master and one or more slaves are allowed in a network, following a master-slave model. Through NMT services, nodes can be initialized, started, monitored, reset and stopped. All nodes are considered NMT slaves.

For example, after a CANopen slave is powered on it sends a message with frame ID 0x702 and data 0x00, indicating that the device has booted up and its node ID is 2. As another example, when a CANopen master sends a frame with ID 0x000 and data 0x01, 0x02 to a slave, the slave with node ID 2 enters the operational state.

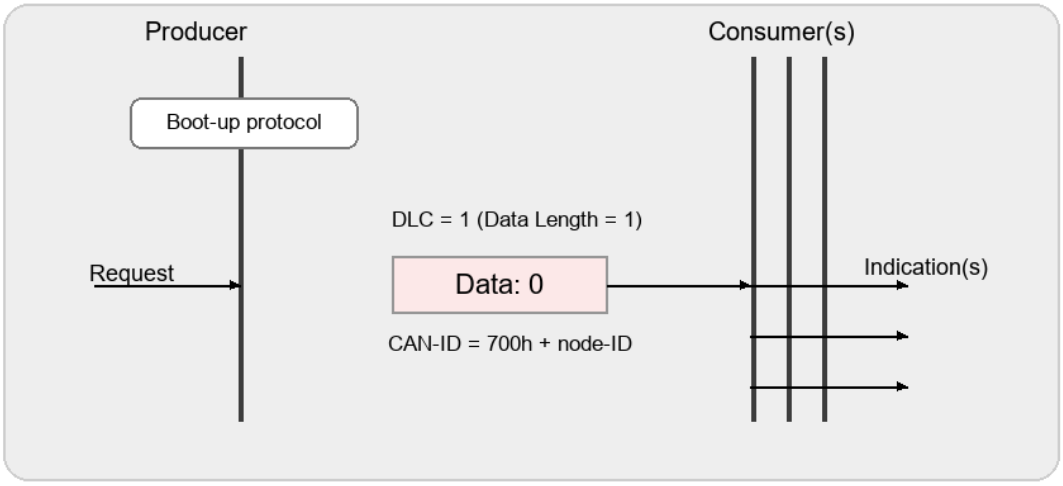


Figure A.4 Boot-up message producer/consumer model

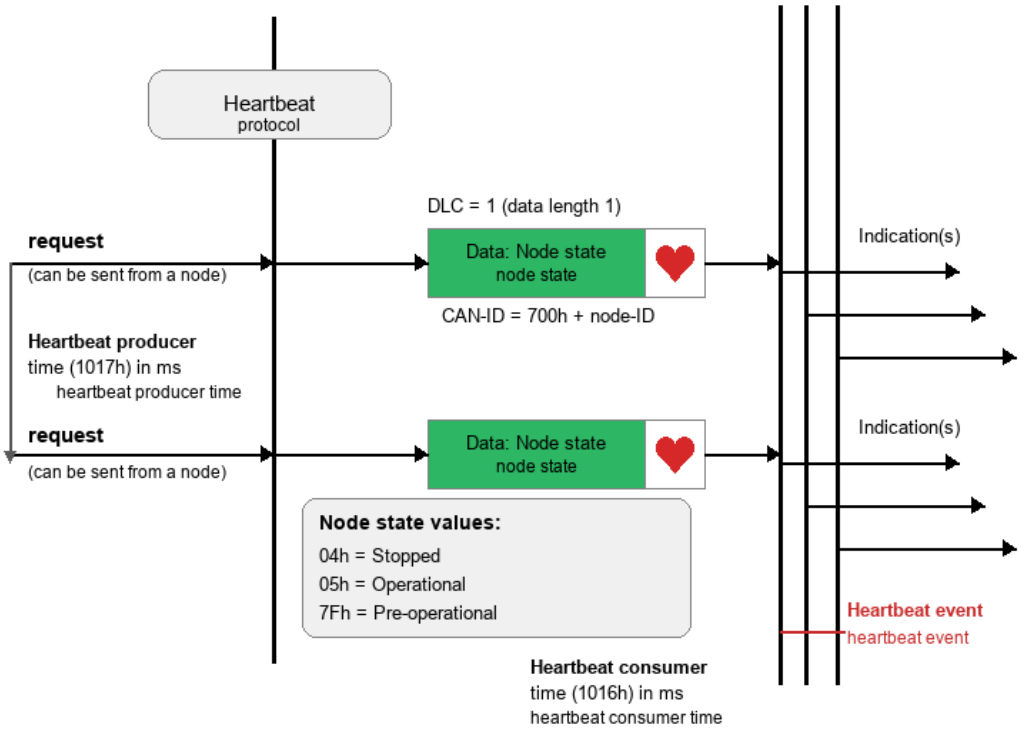


Figure A.5 Heartbeat protocol

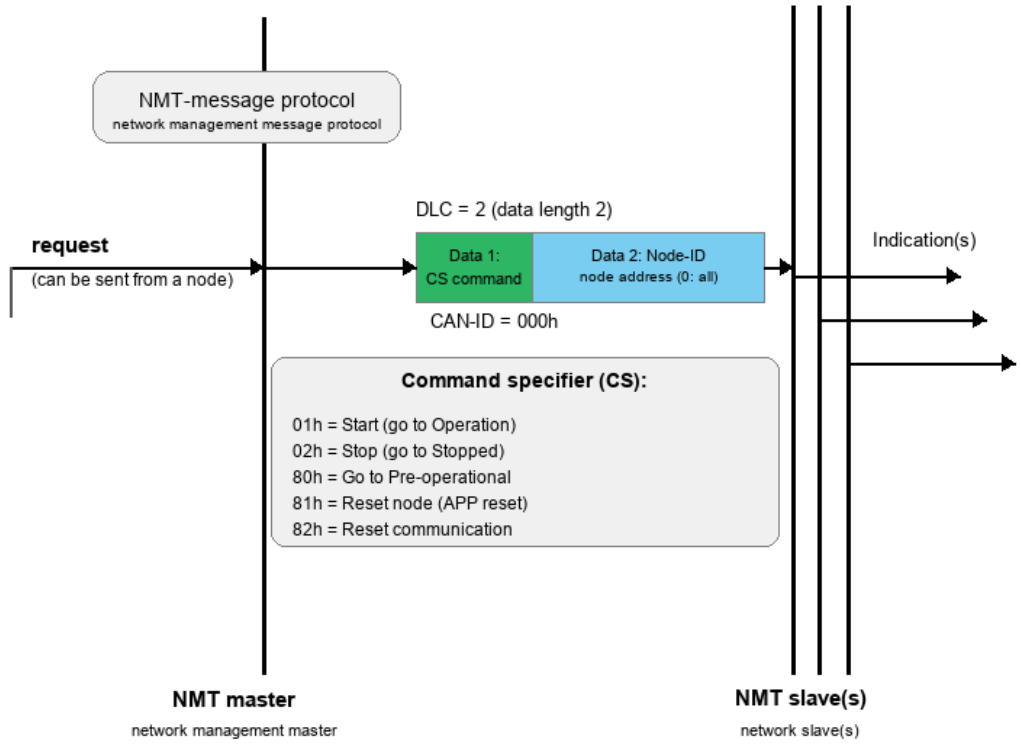


Figure A.6 NMT message protocol

2. Service Data Object (SDO): SDO is mainly used for parameter configuration of slaves by the master. Its greatest feature is service confirmation: each message generates a response to ensure accurate data transmission. In a CANopen system, the slave usually acts as the SDO server and the master as the client. Through index and subindex, the client can access objects in the server's object dictionary. Thus the master can access any object dictionary entry of a

slave, and SDO can transfer data of any length (data longer than 4 bytes is split into multiple messages).

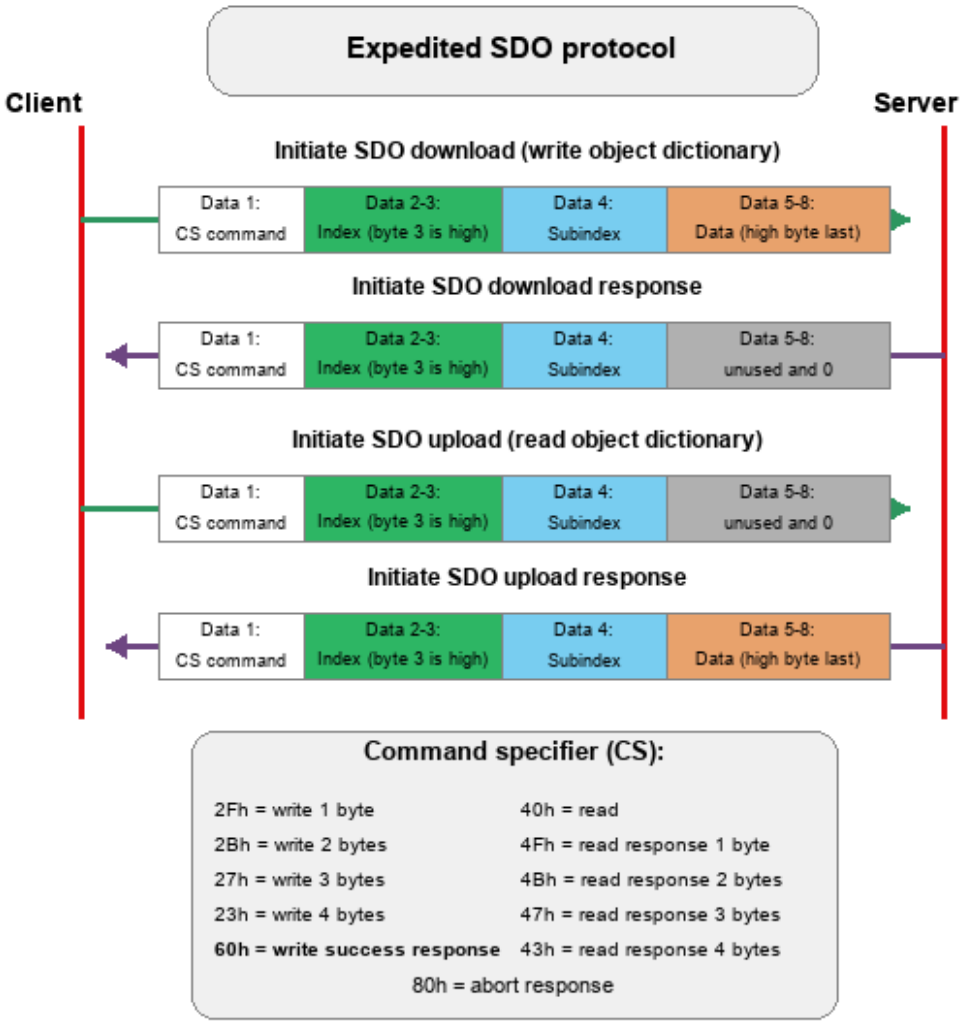


Figure A.7 Expedited SDO protocol

3. Process Data Object (PDO): PDO transfers real-time data using a producer-consumer model, as shown in Figure A.3. The data length is limited to 1~8 bytes. PDO communication objects have these features:

- PDO communication has no protocol-defined content; the PDO data content is defined by its CAN-ID (also called COB-ID).
- Each PDO is described by two objects in the object dictionary: PDO communication parameters (COB-ID, transmission type, timer period) and PDO mapping parameters (a list of object dictionary objects mapped to the PDO, including data length in bits). Both producer and consumer must know the mapping parameters to interpret PDO content correctly.
- PDO message content is predefined; if variable PDO mapping is supported, the PDO can be configured via SDO.
- PDO can use several transmission methods: synchronous transmission (non-cyclic or cyclic, triggered by receiving SYNC, 1~240 sync objects can be set) and asynchronous transmission (triggered by a specific event, either a remote frame with the same COB-ID or an object-specific event defined in the device profile, e.g. timed transmission or change of data state).

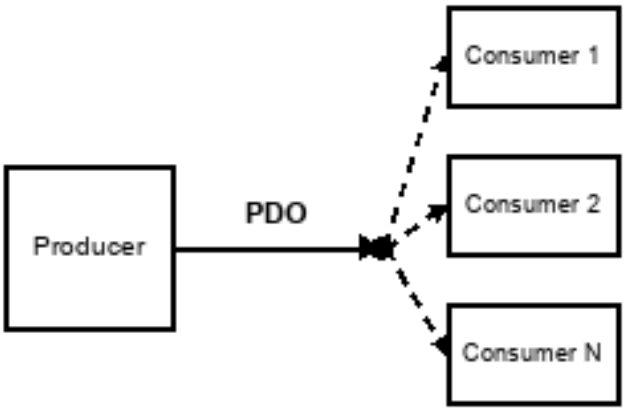


Figure A.3 Producer-consumer model

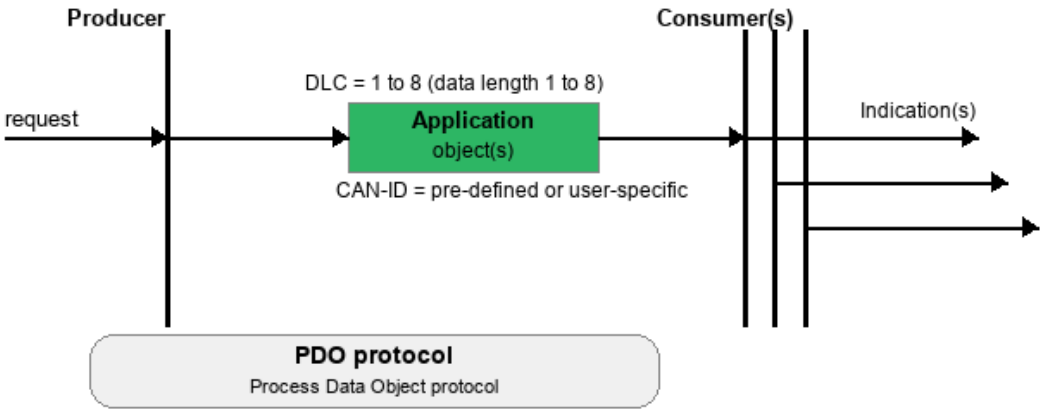


Figure A.8 PDO protocol communication

4. Predefined or special-function objects: these provide specific functions for CANopen devices and facilitate master management of slaves. COB-IDs are predefined for special functions, mainly:

- SYNC: implements synchronized transmission for the whole network; each node uses it as the PDO sync trigger, so its COB-ID has high priority and the shortest transmission time.
- Time Stamp: provides a common time reference for all nodes.
- Emergency: triggered by an internal device error, sending the internal error code.
- Node/Life Guarding: the master obtains slave status via node guarding; a slave obtains master status via life guarding.
- Boot-up: sent by a slave after initialization, entering the pre-operational state.

A.5 CANopen Network Configuration

The CANopen specification DS305 defines a network configuration protocol, the Layer Setting Service (LSS), which uses the CAN bus to query or modify certain parameters of CANopen modules with LSS slave functionality via a module with LSS master functionality. Using LSS, the following parameters can be queried or modified:

- Node-ID of the CANopen slave.
- Bit timing parameters of the physical layer (baud rate).
- LSS address (identity object 1018h).

8. Disclaimer

Thank you for purchasing GCAN series hardware and software products.

1. Copyright and Translation Notice

The original Chinese-language version of this document was prepared by Shenyang Guangcheng Technology Co., Ltd., which retains the copyright to the original document and its contents.

This English version has been independently translated and prepared by Shenzhen MoCAN Electric Co., Ltd. based on its understanding of the original Chinese documentation. It is intended solely for overseas customer reference and technical support purposes. It has not been officially reviewed, verified or approved by Shenyang Guangcheng Technology Co., Ltd., and should not be regarded as an official or authoritative English version of the original documentation.

The English translation, editorial work and original layout elements created by Shenzhen MoCAN Electric Co., Ltd. are protected by applicable copyright laws. No part of this English version may be reproduced, redistributed, republished or used for commercial purposes without prior written permission from Shenzhen MoCAN Electric Co., Ltd., except as permitted by applicable law.

GCAN is a registered trademark of Shenyang Guangcheng Technology Co., Ltd.

2. Product Information and User Responsibility

Product specifications, technical parameters, functions and other information provided in this document are for reference purposes and may be subject to change without prior notice.

Users are responsible for evaluating the suitability of the products for their specific applications and ensuring proper installation, configuration and operation in accordance with the relevant technical documentation and applicable safety requirements.

For product selection, technical clarification or application-specific requirements, users are advised to consult the manufacturer or an authorized supplier before making purchasing or implementation decisions.

3. Interpretation

This English document is provided as a reference translation of the original Chinese-language documentation. While reasonable efforts have been made to ensure the accuracy and consistency of the translation, differences in technical terminology, interpretation or expression may occur.

In the event of any discrepancy, ambiguity or inconsistency between this English version and the original Chinese-language documentation, the original Chinese-language version shall prevail.

For technical questions or clarification regarding the product specifications and intended functions, users are encouraged to contact Shenyang Guangcheng Technology Co., Ltd. or their authorized supplier.