INOVANCE

MD38CAN2 CANopen Communication Expansion Card User Guide

An Option for the MD Series Modular Vector AC Drive

Data Code: 19012441 A00

Preface

Thank you for using the MD series AC drive and the MD38CAN2 (CANopen) fieldbus adapter from Inovance.

The MD38CAN2 (CANopen) communication card is designed for use with the MD series AC drive for CAN communication. It enables the AC drive to access the high-speed CAN communication network and achieve fieldbus control.

CANopen is an internationally recognized fieldbus standard, and devices that support this protocol can be connected to a CANopen network.

Before using this product, please read through this user guide.



Figure 1.1: MD38CAN2 card appearance

2. Installation and Setup

■ 2.1 MD380CAN2 Adapter Installation

The MD38CAN2 card must be embedded in the Inovance AC drive. Before installation, disconnect the power supply of the AC drive and wait for 10 minutes until the power indicator of the AC drive is off. Install the expansion card in accordance with Figure 2.1.

Insert the MD38CAN2 expansion card into the AC drive and fasten the screw.

Note: The MD38CAN2 expansion card (CANopen) must not be installed or removed while powered.

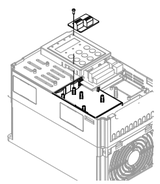

Figure 2.1: MD38CAN2 installation

■ 2.2 Hardware Layout

The hardware layout of MD380CAN2 is shown in Figure 2.2. Terminal CN1 is the CAN bus communication interface. DIP switch S1 is used to set the CAN bus termination resistor. DIP switches S2 and S3 are used to set the CAN communication baud rate and address. Three LEDs indicate the operation status. Pin header SW1 is to be used by the manufacturer for testing purposes. Users must not connect pin header SW1.

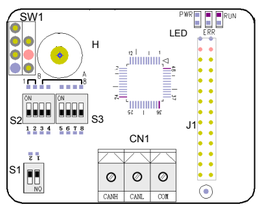

Figure 2.2: Hardware layout of MD38CAN2

Symbol	Hardware Name	Function Description
SW1	SW1 pin header	For manufacturer use only
S2/S3	DIP switch	Sets the CAN communication baud rate and address
S1	Termination resistor settings	Sets the termination resistor for the CAN bus
J1	Pin header connector	Connects to the AC drive
H	Retaining screw hole	Matches an M3x8 self-tapping screw to fix the card
CN1	Wiring terminal	CANopen bus terminal
LED	Status indicator	Indicates operation status

■ 2.3 Interface Description

1) Communication interface

Terminal CN1 is used to connect the CAN bus interface. CN1 includes three interfaces, as explained in Table 2.1.

Table 2.1: Communication interface description

No.	Terminal Mark	Description
1	CANH	Connects to the positive end of the CAN bus
2	CANL	Connects to the negative end of the CAN bus
3	CGND	Connects to the shield of the CAN bus

2) CAN bus topology

Figure 2.3 shows the CAN bus topology. It is recommended to use the shielded twisted pair cable as the CAN bus. Connect a 120 Ω termination resistor at each end of the bus to avoid signal reflection. Ground the shield at one single point.

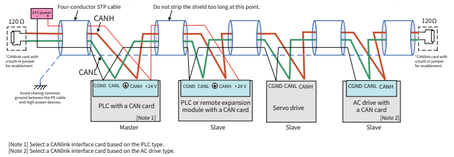

Figure 2.3: CAN bus topology

∗ Inovance PLC and MDCAN2 card come with resistors. When using them at ends of the bus, enable their resistors by setting the DIP switches.

3) CAN transmission distance

The transmission distance of the CANopen bus is directly related to the baud rate and communication cable. Table 2.2 shows the relationship between the maximum transmission distance and the baud rate.

Table 2.2: Relationship between the bus length and baud rate

Baud Rate	1 Mbps	500 kbps	250 kbps	125 kbps
Bus length	30 m	80 m	125 m	250 m

4) DIP switch

The DIP switch S1 on the MD38CAN2 card consists of two switches. It is used to configure the CAN bus termination resistor. It is recommended to enable termination resistors at both ends of the network topology. On the DIP switch, "ON" represents 1, and "OFF" represents 0.

Table 2.3: MD38CAN2 termination resistor

Switch		Termination Resistor
1	2	
0	0	Not connected
1	1	Connected

The DIP switches S2 and S3 on the MD38CAN2 include eight switches, which are used to set the CAN bus communication baud rate and the communication device address. The DIP switch numbering is shown in Figure 2.4. Switches 1 and 2 set the baud rate, and Switches 3 to 8 set the CANopen address. On the DIP switch, "ON" represents 1, and "OFF" represents 0.

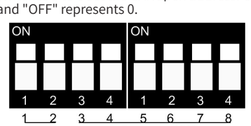

Figure 2.4: MD38CAN2 DIP switch

∗ Baud rate

Table 2.4 lists the mapping relations between the DIP switch settings and the baud rates. Up to four baud rates can be set.

Table 2.4: MD38CAN2 baud rates

Switch		Baud Rate
1	2	
0	0	125 kbps
0	1	250 kbps
1	0	500 kbps
1	1	1 Mbps

∗ CANopen device address

MD38CAN2 provides six switches for setting the CANopen communication address. Switch 3 is the most significant bit, and Switch 8 is the least significant bit. Switches 3 to 8 correspond to bits 5 to 0 of a 16-bit binary integer. The range of addresses that can be set by the DIP switch is from 1 to 63, as shown in Table 2.5. Address 0 is reserved and must not be used. Setting the address to 0 will render the MD38CAN2 card not operating.

Table 2.5: Address settings by the MD38CAN2 DIP switch

Switch						Address
3	4	5	6	7	8	
0	0	0	0	0	0	Reserved
0	0	0	0	0	1	1
0	0	0	0	1	0	2
0	0	0	0	1	1	3
...						...
1	1	1	1	1	1	63

5) Status indicator

Table 2.6: MD38CAN2 status indicators

Indicator	Status	Description
PWR (Red)	ON	Power supply normal
	OFF	Power supply abnormal. Check whether the installation is correct.
ERR (Red)	ON	AC drive communication timeout
	Blinking fast	Incorrect CANopen address setting
	Blinks twice	CANopen emergency message for AC drive fault
RUN (Green)	ON	Running
	Blinking	Pre-operational
	OFF	Stopped

3. MD38CAN2 Card Protocols

■ 3.1 Software Features

The MD38CAN2 card supports five protocols, as explained below:

- Supports the Node Guard protocol, allowing the master to query device status
- Supports the Heartbeat protocol, allowing the slave to report its status to the master at regular intervals
- Only supports the accelerated transmission mechanism for Service Data Object (SDO) (one parameter consisting of two bytes is transmitted at a time)
- Supports three TPDOs and three RPDOs
- Supports emergency objects

■ 3.2 Communication Object COB-ID

CANopen provides multiple communication objects. Each communication object has distinct characteristics (see the CANopen standard protocol for details), allowing for optimal selection based on actual application scenarios. This expansion card uses predefined communication object identifiers (COB-IDs), following the rules below:

(1) NMT object: 0x000

(2) SYNC object: 0x080

(3) SDO object
- Transmit SDO: 0x600 + Node-ID
- Receive SDO: 0x580 + Node-ID

(4) PDO object
- RPDO1: 0x200 + Node-ID
- RPDO2: 0x300 + Node-ID
- RPDO3: 0x400 + Node-ID
- TPDO1: 0x180 + Node-ID
- TPDO2: 0x280 + Node-ID
- TPDO3: 0x380 + Node-ID

(5) EMCY object: 0x80 + Node-ID
- Node-ID: Device ID (station address), set through the DIP switch
- The communication object COB-ID is allocated in a fixed form and cannot be modified.

■ 3.3 Operations on AC Drive Parameters

1) AC drive parameter mapping

∗ AC drive parameter address

The AC drive parameter addresses are divided into parameter number addresses and non-parameter number addresses. See the MODBUS Communication Protocol section in the "MD380 Series High-Performance Vector AC Drive User Guide" for the designation rules of parameter number addresses.

∗ Mapping explanation

The parameter groups of the AC drive are mapped to the CANopen object dictionary range of 0x2000 to 0x20FF. The parameter numbers are mapped to the object dictionary sub-indexes by adding 1 to the parameter number. For example, for the AC drive parameter F0-03, the index in the object dictionary is 0x20F0 and the sub-index is 0x04.

∗ Explanation of AC drive parameter number mapping

MD380 parameter numbers are divided into three groups: F0 to FF, A0 to AF, and U0 to UF.

For a parameter reading operation, the mapping addresses are as follows:

Parameter Group	CANopen Index
F0 to FF	0x20F0 to 0x20FF
A0 to AF	0x20A0 to 0x20AF
U0 to UF	0x2070 to 0x207F

For an EEPROM writing operation, the mapping addresses are as follows:

Parameter Group	CANopen Index
F0 to FF	0x20F0 to 0x20FF
A0 to AF	0x20A0 to 0x20AF

For a RAM writing operation, the mapping addresses are as follows:

Parameter Group	CANopen Index
F0 to FF	0x2000 to 0x200F
A0 to AF	0x2040 to 0x204F

For mapping a parameter number to a CANopen sub-index, add "1" to the corresponding index:

Parameter Group	CANopen Index
0x0 to 0xFE	0x1 to 0xFF

For example, in the case of parameter F0-17 (acceleration time 1):

- To read the F0-17 value, the object dictionary index is 0x20F0, and the sub-index is 0x12.
- To write the F0-17 value and write it to the EEPROM, the object dictionary index is 0x20F0, and the sub-index is 0x12.
- To write the F0-17 value and write it only to RAM, the object dictionary index is 0x2000, and the sub-index is 0x12.

Notes:

- The motor auto-tuning function cannot be implemented by changing parameters through communication.
- Parameters of the AC drive are stored in the EEPROM and can be read repeatedly but must not be written repeatedly. When programming, be cautious with the write commands to parameters. In PLC programs, avoid unconditional driving to prevent cyclic communication write operation that could damage the AC drive's memory.
- If repeated writing to a parameter is necessary, use the mapping address of the parameter for RAM writing. For example, multiple writes can be made to the RAM mapping address 0x2002 for F0-02, but the EEPROM mapping address 0x20F2 must not be repeatedly written to.

2) Examples of operations on the AC drive (SDO)

In the following example, the CANopen address of the AC drive is "0x06".

∗ Reading the AC drive command source

To read the command source of the AC drive (F0-02), the master sends a CANopen message as shown in Table 3.1.

Table 3.1: Message sent by the master during parameter F0-02 reading

Message ID (Hex)	RTR	Data (Hex)
0x606	0	40 F0 20 03 00 00 00 00

The AC drive returns a CANopen response message as shown in Table 3.2. The value of F0-02 is 0x0002, indicating that the current command source of the AC drive is communication control.

Table 3.2: Message returned by the AC drive during parameter F0-02 reading

Message ID (Hex)	RTR	Data (Hex)
0x586	0	4B F0 20 03 02 00 00 00

∗ Setting the command source to the control panel

To set the command source to the control panel, that is, to set parameter F0-02 of the AC drive to 0, the master sends a CANopen message as shown in Table 3.3.

Table 3.3: Message sent by the master during parameter F0-02 writing

Message ID (Hex)	RTR	Data (Hex)
0x606	0	2B F0 20 03 00 00 00 00

The AC drive returns a CANopen response message as shown in Table 3.4. The value of F0-02 is changed to 0, indicating that the current command source is the operating panel.

Table 3.4: Message returned by the AC drive during parameter F0-02 writing

Message ID (Hex)	RTR	Data (Hex)
0x586	0	60 F0 20 03 00 00 00 00

■ 3.4 Read and Write Operations Using SDO

1) Reading the AC drive with SDO

The master uses CANopen Service Data Object (SDO) to perform read operations on the AC drive. Table 3.5 describes the format of data sent by the master.

Table 3.5: SDO sent during a read operation

CAN	CANopen Data	Description
11-bit ID	0x600 + Node-ID	Node-ID (device address) set by the DIP switch
RTR	0	Remote frame flag, 0
DATA0	Command code (0x40)	0x40 read command
DATA1	Low-order byte of index	Parameter group (0xF0 for group F0)
DATA2	High-order byte of index	Mapping address (see Section 4.3.1)
DATA3	Sub-index	Parameter number + 1 ("0x03")
DATA4	Data 1	Reserved, 0
DATA5	Data 2	Reserved, 0
DATA6	Data 3	Reserved, 0
DATA7	Data 4	Reserved, 0

The slave returns a response message for the AC drive reading SDO, as shown in Table 3.6.

If the operation is successful, the return value of the command code is "0x4B", the index remains unchanged, DATA4 and DATA5 are the read data, and DATA6 and DATA7 are 0.

If the operation fails, the return value of the command code is "0x80", the index remains unchanged, while DATA4, DATA5, DATA6, and DATA7 are the SDO failure error code (see relevant CANopen protocol materials for error codes).

Table 3.6: SDO returned during a read operation

CAN	CANopen Data	Description
11-bit ID	0x580 + Node-ID	Node-ID (device address) set by the DIP switch
RTR	0	Remote frame flag, 0
DATA0	Return value of the command code	Success: 0x4B; failure: 0x80
DATA1	Low-order byte of index	Parameter group (0xF0 for group F0)
DATA2	High-order byte of index	Mapping address (see Section 4.3.1)
DATA3	Sub-index	Parameter number + 1 ("0x03")

CAN	CANopen Data	Description
DATA0	Return value of the command code	Success: 0x4B; failure: 0x80
DATA1	Low-order byte of index	Parameter group (0xF0 for group F0)
DATA2	High-order byte of index	Mapping address (see Section 4.3.1)
DATA3	Sub-index	Parameter number + 1 ("0x03")
DATA4	Data 1	Low-order byte of data
DATA5	Data 2	High-order byte of data
DATA6	Data 3	Success: 0 Failure: SDO operation failure error code (See the CANopen protocol.)
DATA7	Data 4	

2) Writing to the AC drive with SDO

The CANopen Service Data Object (SDO) is used to perform write operations on the AC drive. Table 3.7 describes the format of data sent by the master.

Table 3.7: SDO sent during a write operation

CAN	CANopen Data	Description
11-bit ID	0x600 + Node-ID	Node-ID (device address) set by the DIP switch
RTR	0	Remote frame flag, 0

DATA0	Command code	0x2B
DATA1	Low-order byte of index	Parameter group (0xF0 for group F0)
DATA2	High-order byte of index	Mapping address (see Section 4.3.1)
DATA3	Sub-index	Parameter number + 1 ("0x03")
DATA4	Data 1	Low-order byte of data
DATA5	Data 2	High-order byte of data
DATA6	Data 3	Reserved, 0
DATA7	Data 4	Reserved, 0

The slave returns a response message for the AC drive writing SDO, as shown in Table 3.8.

If the operation is successful, the return value of the command code is "0x60", the index remains unchanged, and DATA4, DATA5, DATA6, and DATA7 are 0.

If the operation fails, the return value of the command code is "0x80", the index remains unchanged, while DATA4, DATA5, DATA6, and DATA7 are the SDO failure error code (see relevant CANopen protocol materials for error codes).

Table 3.8: SDO returned during a write operation

CAN	CANopen Data	Description
11-bit ID	0x580 + Node-ID	Node-ID (device address) set by the DIP switch
RTR	0	Remote frame flag, 0
DATA0	Return value of the command code	Success: 0x60; failure: 0x80
DATA1	Low-order byte of index	Parameter group (0xF0 for group F0)
DATA2	High-order byte of index	Mapping address (see Section 4.3.1)
DATA3	Sub-index	Parameter number + 1 ("0x03")
DATA4	Data 1	Success: 0 Failure: SDO operation failure error code (See the CANopen protocol.)
DATA5	Data 2	
DATA6	Data 3	
DATA7	Data 4	

3) SDO Error Codes

Error Code	Description	Error Code	Description
0x05040000	SDO access timeout	0x06070010	Data type mismatch: service parameter length mismatch
0x06010000	Inaccessible object	0x06090011	Sub-index nonexistent
0x06010001	Reading a write-only object	0x06090030	Write access beyond the parameter value range
0x06010002	Writing to a read-only object	0x08000022 ∗	Failure to transmit or save data to the application due to the current device state
0x06020000	Object dictionary nonexistent	0x08000000	General error

Note∗: Error codes will be returned upon operations on the AC drive's object dictionary parameters, parameters that cannot be changed during running, and address parameters, as well as write operations on read-only parameters.

■ 3.5 AC Drive Operations Using PDO

1) AC drive operations using RPDO

Three RPDOs (RPOD1, RPDO2, and RPDO3) are supported. The mapping relations of these PDOs are summarized in Table 3.9. For more details, see the "MD380CANopen.eds" file.

Table 3.9: RPDO mapping table

RPDO	Name	Description
RPDO1	FE-00	Drivecom (fixed mapping)
	FE-01	Frequency
	FE-02	
	FE-03	
RPDO2	FE-04	
	FE-05	
	FE-06	
	FE-07	

RPDO3	FE-08	
	FE-09	
	FE-10	
	FE-11	

∗ Drivecom: Drive control command (valid when the command source is set to communication control, that is, F0-02 is set to 2)

◆ 0001: Forward running 0004: Reverse jog

◆ 0002: Reverse running 0005: Coast to stop

◆ 0003: Forward jog 0006: Decelerate to stop 0007: Fault reset

∗ Frequency: Drive frequency reference (valid when the frequency source is set to communication control, that is, F0-03 is set to 9, with the frequency value in the unit of 0.01 Hz)

2) AC drive operations using TPDO

Three TPDOs (TPOD1, TPDO2, and TPDO3) are supported. The mapping relations of these PDOs are summarized in Table 3.10. For more details, see the "MD380CANopen.eds" file.

Table 3.10: TPDO mapping table

TPDO	Name	Description
TPDO1	FE-20	Drive status (fixed mapping)
	FE-21	Frequency
	FE-22	
	FE-23	
TPDO2	FE-24	
	FE-25	
	FE-26	
	FE-27	
TPDO3	FE-28	
	FE-29	
	FE-30	
	FE-31	

∗ Drive status: Operation status of the drive

◆ Bit 0: 0 indicates "stopped"; 1 indicates "running".

◆ Bit 1: 0 indicates "forward running"; 1 indicates "reverse running".

◆ Bit 2: 0 indicates "no fault"; 1 indicates "AC drive fault".

◆ Bit 3: 0 indicates "not reached"; 1 indicates "reached".

◆ Bit 4 to bit 7: Reserved

◆ Bit 8 to bit 15: AC drive fault code

∗ Frequency: Operation frequency of the drive (in the unit of 0.01 Hz)

NOTE PDO mapping must include a command word and a status word.

4. CANopen Communication Parameters

■ 4.1 Parameters for CANopen Settings

To use the CANopen expansion card, the AC drive parameters in Table 4.1 must be set.

Table 4.1: Related device parameters

Parameter	Name	Value Range
F0-28	Communication expansion card type	0: RS485 communication card 1: Communication expansion card (CANopen or PROFIBUS DP)
F0-02	Command source selection	0: Operating panel control 1: Terminal I/O control 2: Communication control
F0-03	Main frequency source X selection	9: Communication control
F0-04	Auxiliary frequency source Y selection	Same as F0-03 (main frequency reference source selection)

■ 4.2 Control Parameters in Communication

Common control parameters in communication are summarized in Table 4.2. The index address is 0x2073. For other parameters, see the "MD380 Series High-Performance Vector AC Drive User Guide".

Table 4.2: Common control parameters in communication

Parameter	Name	CANopen Sub-index	Value Range	
U3-16	Frequency reference	0x11	–Maximum frequency to +Maximum frequency 0.01 Hz	
U3-17	Control command	0x12	0001: Forward running 0002: Reverse running 0003: Forward jog 0004: Reverse jog	0005: Coast to stop 0006: Decelerate to stop 0007: Fault reset
U3-18	DO control	0x13	Bit 0: DO1 output control Bit 1: DO2 output control Bit 2: RELAY1 output control Bit 3: RELAY2 output control Bit 4: FMR output control	Bit 5: VDO1 Bit 6: VDO2 Bit 7: VDO3 Bit 8: VDO4 Bit 9: VDO5
U3-19	AO1 control	0x14	0 to 7FFF indicate 0% to 100%.	
U3-20	AO2 control	0x15	0 to 7FFF indicate 0% to 100%.	
U3-21	FMP control	0x16	0 to 7FFF indicate 0% to 100%.	
U3-22	Reserved	Reserved	User write operations not allowed	
U3-23	Speed control	0x18	Signed data, 1 RPM	

■ 4.3 Monitoring Parameters in Communication

Common monitoring parameters in communication are summarized in Table 4.3. For other parameters, see the "MD380 Series High-Performance Vector AC Drive User Guide".

Table 4.3: Common monitoring parameters in communication

Parameter	Name	CANopen Sub-index	Unit
U0-00	Operation frequency (Hz)	0x01	0.01 Hz
U0-01	Frequency reference (Hz)	0x02	0.01 Hz
U0-02	Bus voltage (V)	0x03	0.1 V
U0-03	Output voltage (V)	0x04	1 V
U0-04	Output current (A)	0x05	0.01 A
U0-05	Output power (kW)	0x06	0.1 kW
U0-06	Output torque (%)	0x07	0.1%
U0-07	DI state	0x08	1
U0-08	DO state	0x09	1
U0-09	AI1 voltage (V)	0x0A	0.01 V
U0-10	AI2 voltage (V)	0x0B	0.01 V
U0-11	AI3 voltage (V)	0x0C	0.01 V
U0-12	Count value	0x0D	1
U0-13	Length value	0x0E	1
U0-14	Load speed display	0x0F	1
U0-15	PID reference	0x10	1
U0-16	PID feedback	0x11	1
U0-17	PLC stage	0x12	1
U0-18	Pulse input frequency (Hz)	0x13	0.01 kHz
U0-19	Feedback speed (Hz)	0x14	0.01 Hz
U0-20	Remaining running time	0x15	0.1 min
U0-21	AI1 uncalibrated voltage	0x16	0.001 V
U0-22	AI2 uncalibrated voltage	0x17	0.001 V
U0-23	AI3 uncalibrated voltage	0x18	0.001 V
U0-24	Linear speed	0x19	1 m/min
U0-25	Current power-on time	0x1A	1 min
U0-26	Current running time	0x1B	0.1 min
U0-27	Pulse input frequency	0x1C	1 Hz
U0-28	Communication reference	0x1D	0.01%
U0-29	Encoder feedback speed	0x1E	0.01 Hz
U0-30	Main frequency X display	0x1F	0.01 Hz
U0-31	Auxiliary frequency Y display	0x20	0.01 Hz
U0-32	Any memory address	0x21	1
U0-33	Synchronous motor rotor position	0x22	0.1°
U0-34	Motor temperature	0x23	1°C
U0-35	Target torque (%)	0x24	0.1%
U0-36	Resolver position	0x25	1
U0-37	Power factor angle	0x26	0.1°
U0-38	ABZ position	0x27	1

Parameter	Name	CANopen Sub-index	Unit
U0-39	Target voltage upon V/f separation	0x28	1 V
U0-40	Output voltage upon V/f separation	0x29	1 V
U0-41	DI state display	0x2A	1
U0-42	DO state display	0x2B	1
U0-43	DI function state display 1 (functions 01 to 40)	0x2C	1
U0-44	DI function state display 2 (functions 41 to 80)	0x2D	1
U0-45	Fault information	0x2E	1
U0-58	Z signal counting	0x3B	1
U0-59	Frequency reference (%)	0x3C	0.01%
U0-60	Operation frequency (%)	0x3D	0.01%
U0-61	AC drive state	0x3E	1
U0-62	Current fault code	0x3F	1
U0-63	Data sent by master during point-to-point communication	0x40	0.01%
U0-64	Data received by slave during point-to-point communication	0x41	0.01%
U0-65	Torque upper limit	0x42	0.1%
U0-66	Expansion card model	0x43	100: CANopen 200: PROFIBUS DP
U0-67	Expansion card version	0x44	Two decimal places
U0-68	AC drive state	0x45	
U0-69	Operation frequency	0x46	
U0-70	Motor speed	0x47	1 RMP
U0-71	Output current	0x48	0.1 A
U0-72	Bus status		CAN bus shutdown count

■ 4.4 Emergency Message and AC Drive Fault Description

1) Emergency message data

An emergency message includes seven bytes, as shown in Table 4.4.

Table 4.4: Emergency message data

Emergency Error Code	Error Register	Manufacturer-Specified Error Code
0 to 1	2	3 to 7

∗ Emergency error code: See relevant sections of the DS301 documentation. 0x8100 indicates a communication error, and 0xFF00 indicates a manufacturer-specified error.

∗ Error register: See the 1001H data value in the object dictionary in relevant sections of the DS301 documentation. Bit 0 is the error flag, bit 4 is the communication error flag, and bit 7 indicates the manufacturer-specified error.

∗ Manufacturer-specified error code: The corresponding AC drive fault code.

2) Fault codes

Table 4.5 lists the standard fault codes of the MD380 series AC drive. For more details, see the user guide.

Table 4.5: Fault codes

AC Drive Fault Information	AC Drive Fault Information
0000: No fault 0001: Reserved 0002: Overcurrent during acceleration 0003: Overcurrent during deceleration 0004: Overcurrent during operation at constant speed 0005: Overvoltage during acceleration 0006: Overvoltage during deceleration 0007: Overvoltage during operation at constant speed 0008: Pre-charge resistor overload 0009: Undervoltage 000A: AC drive overload 000B: Motor overload 000C: Input phase loss 000D: Output phase loss 000E: Module over-temperature 000F: External fault 0010: Communication fault 0011: Contactor fault 0012: Current detection fault 0013: Motor auto-tuning fault 0014: Encoder/PG card fault	0015: Parameter read-write fault 0016: AC drive hardware fault 0017: Motor short-circuited to ground 0018: Reserved 0019: Reserved 001A: Running time reached 001B: User-defined fault 1 001C: User-defined fault 2 001D: Power-on time reached 001E: Load lost 001F: PID feedback lost during operation 0028: Fast current limit timeout 0029: Motor switchover fault during operation 002A: Excessive speed deviation 002B: Motor overspeed 002D: Motor over-temperature 005A: Encoder PPR reference error 005B: Encoder not connected 005C: Initial position error 005E: Speed feedback error

5. Simple Diagnosis

■ 5.1 Overview

Parameter U0-72 of the AC drive provides a simple diagnosis function. This parameter shows the number of times that the CAN bus has been shut down due to strong interference after power-on.

■ 5.2 Diagnosis

If the displayed value is greater than 0 and does not continue to increase, it indicates that the network has experienced prolonged strong interference. If the value is greater than 0 and increases within 5 minutes, it indicates that the network is experiencing interference or the configuration is incorrect, which requires troubleshooting.

■ 5.3 Action

Check whether all the nodes have the same baud rate. Check for duplicate node addresses. Check whether the DIP switches are set correctly and in place. Check whether the baud rate and address of the master are set correctly.

Check whether the termination resistors are connected only at both ends of the bus. Power off all devices and check whether the resistance between CANH and CANL of the bus is between 50 Ω and 60 Ω by using a multimeter.

Check whether CANH and CANL of a node are reversely connected and whether CGND of the bus interface is connected (typically, CGND of all devices are connected together and not grounded).

6. Overview of CANopen Protocol

■ 6.1 CANopen Protocol Introduction

CANopen is an application layer protocol for a network transmission system based on the CAN serial bus. The CAN bus adheres to the ISO/OSI standard model and defines the data link layer and part of the physical layer within the OSI model. CANopen supports a multi-master approach where any node on the network can actively send information to other nodes. Nodes in the network are assigned with different priorities based on system real-time requirements, allowing for reduced bus arbitration time in the case of bus conflicts. The CAN network adopts communication data block coding instead of traditional address coding. With data block coding, the number of nodes in the network is theoretically not limited, and different nodes can receive the same data at the same time. This coding mode also features short transmission byte, high speed, great fault tolerance, and reliable data transmission, making it suitable for industrial control and distributed real-time control. The CANopen device model is shown in the Figure 6.1.

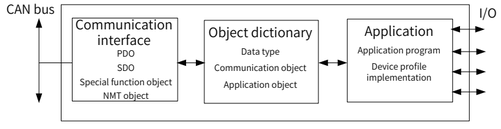

Figure 6.1: CANopen device model

■ 6.2 Object Dictionary

The object dictionary is the most crucial part of device specifications. It is an ordered collection of parameters and variables. It includes all parameters of the device profile and device network status. Essentially, the object dictionary is a group of objects accessible through the network in an ordered predefined manner.

The CANopen protocol utilizes an object dictionary with 16-bit indexes and 8-bit sub-indexes. The structure of the object dictionary is illustrated in Figure 6.3. A master node or configuration tool can access all values in the object dictionary of a slave node.

Index	Object
000	Not in use
0001— 001F	Static data type (standard data type such as Boolean and Integer16)
0020— 003F	Complex data type (predefined structure composed of simple types, such as PDOCommPar and SDOParmeter)
0040— 005F	Complex data type specified by the manufacturer
0060— 007F	Static data type specified by the device profile
0080— 009F	Complex data type specified by the device profile
00A0— 0FFF	Reserved
1000 — 1FFF	Communication profile area (such as device type, error register, and supported PDO quantity)
2000 — 5FFF	Manufacturer-specific profile area
6000 — 9FFF	Standard device profile area (such as "DSP-401 I/O module device profile": Read State 8 Input Lines)
A000— FFFF	Reserved

Figure 6.3: Structure of the object dictionary

■ 6.3 Common Communication Objects

☞ Network management object (NMT)

NMT includes Boot-up messages, the Heartbeat protocol, and NMT messages. Operating in a master-slave communication mode, NMT is used to manage and monitor nodes in the network, primarily facilitating three functions: node status control, error control, and node startup.

☞ Service data object (SDO)

By using indexes and sub-indexes, SDO enables clients to access items in the device object dictionary.

SDO is implemented through the CAN-based message specification (CMS) object in the multivariate domain at the CAN application layer (CAL). SDO allows the transmission of data of any length (segmented into several messages when the data length exceeds four bytes).

The protocol ensures a service acknowledgment type, generating a response for each message. SDO request and response messages each always contain eight bytes.

☞ Process data object (PDO)

Used for real-time data transmission, PDO transfers data from one creator to one or more receivers. Data transmission is limited to one to eight bytes.

Each CANopen device has eight default PDO channels: four TPDO channels and four RPDO channels.

PDO supports synchronous and asynchronous transmission modes, determined by the corresponding communication parameters.

7. CANopen Message Format

■ 7.1 NMT Module Control Message

Only the NMT-Master node can send NMT Module Control messages. The message format is as shown in Table 7.1. The COB-ID is always "0x000". Data0 represents a single-byte command as explained in Table 7.2. Data1 represents a single-byte CANopen network device address, with "0" indicating a broadcast message for all slave devices in the network.

For example, to set a device with address "6" to an operational state, the command is "0x000 0x01 0x06".

Table 7.1: NMT message

COB-ID	RTR	Data0	Data1
0x000	0	Command word	Node ID

Table 7.2: Command in an NMT message

Command	Description
0x01	Start Remote Node
0x02	Stop Remote Node
0x80	Enter Pre-operational State
0x81	Reset Node
0x82	Reset Communication

■ 7.2 NodeGuarding Message

Through the NodeGuarding service, the NMT master node can check the current status of each node. This is particularly meaningful when these nodes are not actively transmitting data.

The standard protocol object 0x100C sets the Guard Time, and 0x100D sets the Guard Time Factor, jointly determining the time period for node protection.

The NMT master node sends remote frames as shown in Table 7.3.

Table 7.3: Message sent by the NodeGuarding master node

COB-ID	RTR
0x700 + Node-ID	1

The response message from the NMT slave node is shown in Table 7.4, with a one-byte status word format as explained in Table 7.5.

Table 7.4: Response message returned by the NodeGuarding slave node

COB-ID	RTR	Data0
0x700 + Node-ID	0	Status word

Table 7.5: Status word returned by the NodeGuarding slave node

Data Bit	Description
Bit 7	Set to 0 or 1 alternatively
Bit 6 to bit 0	Status: 4: Stopped 5: Running 127: Pre-operational

■ 7.3 Heartbeat Message

A node can be configured to generate periodic messages known as Heartbeat messages, as shown in Table 7.6. The status word bit 7 is "0", and bit 6 to bit 0 align with the NodeGuarding response shown in Table 7.5.

The heartbeat time is defined by the standard protocol object 0x1017. A node cannot simultaneously support both the NodeGuarding and Heartbeat protocols.

Table 7.6: Heartbeat message

COB-ID	RTR	Data0
0x700 + Node-ID	0	Status word

INOVANCE Warranty Agreement

1. Inovance provides an 18-month free warranty (subject to information indicated by the barcode on the equipment) to the equipment itself from the date of manufacturing for the failure or damage under normal use conditions.
2. Within the warranty period, a maintenance fee will be charged for damage caused by the following reasons:
   A. Improper use or repair/modification without prior permission
   B. Fire, flood, abnormal voltage, natural disasters, and secondary disasters
   C. Hardware damage caused by dropping or transportation after procurement
   D. Operations not following the user guide
   E. Factors out of the equipment (for example, external device factors)
3. If the equipment is faulty or damaged, fill in the Warranty Card correctly and in detail.
4. The maintenance fee will be charged according to the latest Maintenance Price List of Inovance.
5. The Warranty Card is not re-issuable. Keep the card well and present it to the maintenance personnel during maintenance.
6. For service-related issues, contact Inovance's agent or Inovance directly.
7. By purchasing this product, you agree to this warranty agreement. Inovance reserves the rights for explanation of this agreement.

INOVANCE Warranty Card

Customer information	Company add.:	
	Company name:	Contact:
	Post code:	Tel.:
Product information	Product model:	
	Barcode (paste the barcode here):	
	Agent:	
Fault information	(Maintenance date and description): Maintained by:	

Suzhou Inovance Technology Co., Ltd.

Add.: No. 52, Tian'e Dang Road, Wuzhong District, Suzhou 215104, P.R. China
Tel: (0512) 6637 6666 Fax: (0512) 6285 6720
www.inovance.com