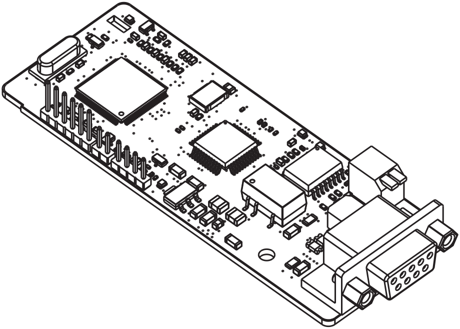
invt

# Operation Manual

## Communication Extension Card



SHENZHEN INVT ELECTRIC CO., LTD.

# Safety precautions

The extension card can be installed and operated only by people who have taken part in professional training on electrical operation and safety knowledge, obtained the certification, and been familiar with all steps and requirements for installing, performing commissioning on, operating, and maintaining the device, and are capable of preventing all kinds of emergencies.

Before installing, removing, or operating the communication card, read the safety precautions described in this manual and the inverter operation manual carefully to ensure safe operation.

For any physical injuries or damage to the device caused due to your neglect of the safety precautions described in this manual and the inverter operation manual, our company shall not be held liable.

- You need to open the housing of the inverter when installing or removing the communication card. Therefore, you must disconnect all power supplies of the inverter and ensure that the voltage inside the inverter is safe. For details, see the description in the inverter operation manual. Severe physical injuries or even death may be caused if you do not follow the instructions.
- Store the communication card in a place that is dustproof and dampproof without electric shocks or mechanical pressure.
- The communication card is electrostatic sensitive. Take measurements to prevent electrostatic discharge when performing operations involving it.
- Tighten the screws up when installing the communication card. Ensure that it is firmly fixed and properly grounded.

# Terminology, abbreviations, and acronyms

| | |
|---|---|
| CAN | Controller Area Network |
| COB | Communication object, a transmitted unit on a CAN network. Communication objects (COBs) carry data and can be transmitted through the whole network. A COB is part of a CAN message frame. |
| EDS | Electronic datasheet, an ASCII file for node configuration, required when a CANopen network is configured. An EDS file contains general information about nodes and their dictionary objects (parameters). |
| NMT | Network management, one of the CAN application-layer service elements in the CAN reference model. It is used for the initialization, configuration, and fault handling of a CAN network. |
| Object dictionary | Stores information about all COBs identified by a device. |
| PDO | Process data object, a type of COBs, used to transmit process data, such as control command, set values, state values, and actual values. |
| PDOn Tx | PDO command transmitted by a slave station to the master station, where n refers to 1, 2, 3, 4. |
| PDOn Rx | PDO command transmitted by the master station and received by a slave station, where n refers to 1, 2, 3, 4. |
| SDO | Service data object, a type of COB, used to transmit non-time key data, such as parameter values. |
| RO | Indicates read-only access. |
| RW | Indicates the read and write access. |
| SYNC | Indicates synchronous transmission. |
| Node-ID | Node ID, that is, address of a communication card. |
| 0x | Indicates that a number with this prefix is a hexadecimal value, for example, 0x10 indicates the decimal value 16. |

# Contents

# 1 Product confirmation

Check the following after receiving a communication extension card product:

- Whether the communication card is damaged.
- Whether the received communication card is the one you purchase according to the bar code label on the PCB.
- Whether all the following items are contained in the product package:

  One communication card, one tie wrap, one tie, one M3 screw, and one manual

If the communication card is damaged, a wrong model is delivered, or some items are missing, contact the supplier in a timely manner.

Obtain the ESD file of the communication card from INVT. The file is named *communication card model*.eds.

Confirm the environmental requirements for application.

Table 1-1 Environmental requirements

| Item | Requirement |
|---|---|
| Operation temperature | -10–+50°C |
| Storage temperature | -20–+60°C |
| Relative humidity | 5%–95% |
| Other weather conditions | No condensation, ice, rain, snow, or hail;<br>solar radiation < 700 $W/m^2$ |
| Air pressure | 70–106 kPa |
| Vibration and impact | 5.9$m/s^2$ (0.6g) at the sine vibration of 9 Hz to 200 Hz |

# 2 PROFIBUS communication card

## 2.1 Overview

PROFIBUS communication cards are optional accessories for inverters. They can be used to connect inverters to PROFIBUS networks. On a PROFIBUS network, inverters are slave devices. The following functions can be performed by using a PROFIBUS communication card:

- Transmit control commands (such as start, stop, and fault reset) to an inverter.
- Transmit speed or torque reference signals to an inverter.
- Obtain state values and actual values from an inverter.
- Modify parameter values of an inverter.

## 2.2 Features

1. PROFIBUS is an international open fieldbus standard that can implement data exchange between various automation components. It is widely applicable to automation in various industries, such as the manufacturing, process, building, transportation, and power industries. It provides effective solutions for implementing integrated automation and intelligentization of field devices.
2. PROFIBUS consists of three mutually compatible components, namely PROFIBUS-Decentralised Peripherals (DP), PROFIBUS-Process Automation (PA), and PROFIBUS-Fieldbus Message Specification (FMS). It adopts the master-slave mode and is generally used for periodic data exchange between inverter devices. PRNV PROFIBUS-DP adapter modules support only the PROFIBUS-DP protocol.
3. The transmission media of a PROFIBUS field bus are twisted pairs (complying with the RS-485 standard), paired cables, or optical cables. The baud rate ranges from 9.6 kbit/s to 12 Mbit/s. The maximum length of a fieldbus cable must be within the range of 100 m to 1200 m, and the specific length depends on the selected transmission rate (see the chapter of "Technical Data" in the inverter manual). A maximum of 31 nodes can be connected to one PROFIBUS network segment when no repeater is used. If repeaters are used, a maximum of 127 nodes (including the repeaters and master stations) can be connected.
4. In PROFIBUS communication, tokens are transmitted between master stations or by master stations to slave stations. Single-master or multi-master systems are supported. The node to respond to the command of a master is selected by the master station, generally a programmable logic controller (PLC). For cyclic master-slave user data transmission and non-cyclic master-master data transmission, a master can also

transmit commands to multiple nodes in broadcast mode. When the broadcast mode is adopted, the nodes do not need to transmit feedback signals to the master. On PROFIBUS networks, nodes cannot communicate with each other.

5. The PROFIBUS protocol is described in details in the EN50170 standard. For more information about PROFIBUS, refer to the EN50170 standard.

## 2.3 Electrical connection

1. Node selection

   The node address of a device is unique on a PROFIBUS bus. The node address is set through the function parameter P15.01, and the value ranges from 0 to 127.

2. Fieldbus terminator

   Each fieldbus segment is configured with two bus terminators, one on each end, to prevent operation errors. Bus terminators can protect the fieldbus signal against electrical reflections. The dual in-line package (DIP) switch on the printed circuit board (PCB) of a communication card is used to connect to the fieldbus terminator. If the communication card is the last or first module on the network, the bus terminator must be set to ON. When a PROFIBUS D-sub connector with a built-in terminator is used, you must disconnect the communication card from the terminator.

## 2.4 Bus network connection

1. Bus communication interfaces

   The most common PROFIBUS transmission mode is the shielded twisted-pair copper cable transmission, in which shielded twisted-pair copper cables (complying with the RS-485 standard) are used.

   The basic characteristics of this transmission technology are described as follows:

   - Network topology: Linear bus with one active fieldbus terminal resistor on each end
   - Transmission rate: 9.6 kbit/s–12 Mbit/s
   - Media: Shielded or unshielded twisted-pair cables, depending on the EMC environmental conditions
   - Number of stations: 32 on each network segment (without repeater); a maximum of 127 (with repeaters)
   - Plug connection: 9-pin D-type plug. The following figure shows the pins of the connector.

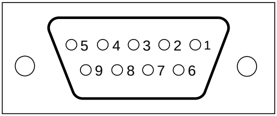



Figure 2-1 Plug of the connector

Table 2-1 Description of the connector pins

| Connector pin | | Description |
|---|---|---|
| 1 | - | Unused |
| 2 | - | Unused |
| 3 | B-Line | Data+ (twisted-pair wire 1) |
| 4 | RTS | Transmitting requests |
| 5 | GND_BUS | Isolation ground |
| 6 | +5V BUS | Isolated 5 V DC power supply |
| 7 | - | Unused |
| 8 | A-Line | Data- (twisted-pair wire 2) |
| 9 | - | Unused |
| Housing | SHLD | PROFIBUS cable shielding wire |

The +5V and GND_BUS pins are used for bus terminators. Optical transceivers (RS-485) and some other devices may need to obtain external power supplies through these pins.

For some devices, the transmission direction is determined by using the RTS pin. In regular application, only the A-Line, B-Line, and SHLD pins are used.

It is recommended that you use the standard DB9 connectors manufactured by Siemens. If the communication baud rate is required to be higher than 187.5 kbps, strictly follow the wiring standards stipulated by Siemens.

2. Repeaters

   A maximum of 32 stations (including the master station) can be connected to each fieldbus segment. If the number of stations to be connected to a fieldbus segment exceeds 32, you need to use repeaters to connect the fieldbus segments. Generally, the number of repeaters connected in series cannot exceed 3.

**Note:** No station address is provided for repeaters, but they are calculated as stations.

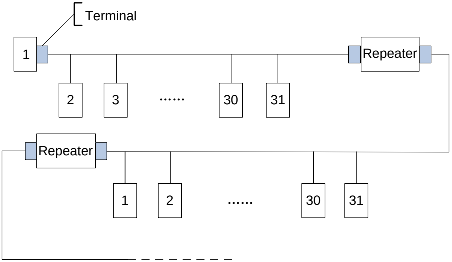



Figure 2-2 Repeaters

3. Transmission rates and maximum transmission distances

The maximum length of a cable depends on the transmission rate. Table 2-2 describes the transmission rates and corresponding transmission distances.

Table 2-2 Transmission rates and corresponding transmission distances

| Transmission rate (kbps) | A-type wire (m) | B-type wire (m) |
|---|---|---|
| 9.6 | 1200 | 1200 |
| 19.2 | 1200 | 1200 |
| 93.75 | 1200 | 1200 |
| 187.5 | 1000 | 600 |
| 500 | 400 | 200 |
| 1500 | 200 | ----- |
| 12000 | 100 | ----- |

Table 2-3 Transmission wire parameters

| Parameter | A-type wire | B-type wire |
|---|---|---|
| Impedance (Ω) | 135–165 | 100–130 |
| Capacitance of a unit length (pF/m) | < 30 | < 60 |
| Circuit resistance (Ω/km) | 110 | -------- |
| Wire core diameter (mm) | 0.64 | > 0.53 |
| Sectional area of wire core ($mm^2$) | > 0.34 | > 0.22 |

Besides the shielded twisted-pair copper cables, you can also use optical fibers for transmission in a PROFIBUS system. When a PROFIBUS system is applied in an environment with strong electromagnetic interference, you can use optical fiber conductors to increase the high-speed transmission distance. Two types of optical fiber conductors can be used. One is low-cost plastic fiber conductors that can be used when the transmission distance is shorter than 50 m; and the other is glass fiber conductors that can be used when the transmission distance is shorter than 1 km.

4. PROFIBUS bus connection diagram

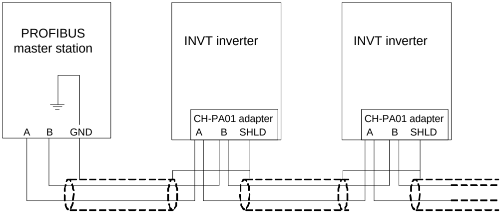



Figure 2-3 PROFIBUS bus connection

Figure 2-3 shows the terminal wiring. The cables are standard PROFIBUS cables, each consisting of a twisted pair and shielding layer. The shielding layers of PROFIBUS cables are directly grounded on all nodes. You can select a proper grounding mode based on the actual situation on site.

**Note:**

1. When connecting the stations, ensure that the data cables are not twisted together. For systems to be used in environments with strong electromagnetic radiation, you need to use cables with shielding layers. The shielding layers can improve electromagnetic compatibility (EMC).

2. If shielding braid or shielding foil is used, connect the two ends of it to the protective ground and cover an area as large as possible to ensure high conductivity. In addition, data cables need to be separated from high-voltage cables.

3. When the data transmission rate is higher than 500 kbit/s, do not use short stub. Use the plugs available in the market. Data input and output cables can be directly connected to those plugs, and the plug of the communication card can be connected or disconnected at any time without interrupting data communication of other stations.

## 2.5 System configuration

1. System configuration

After the communication card is properly installed, you need to configure the master station and inverter to enable the communication between the master station and communication card.

One device description file named GSD file is required for each PROFIBUS slave station on the PROFIBUS bus. The GSD file is used to describe the characteristics of the PROFIBUS-DP device. The software we provide for users includes information about the GSD file of the inverter. You can obtain the type definition files (GSD files) of various masters from INVT.

Table 2-4 Communication card configuration parameters

<table>
<tr><th>Parameter No.</th><th>Parameter name</th><th colspan="2">Setting options</th><th>Default setting</th></tr>
<tr><td>0</td><td>Module type</td><td colspan="2">Read-only</td><td>PROFIBUS-DP</td></tr>
<tr><td>1</td><td>Node address</td><td colspan="2">0–99</td><td>2</td></tr>
<tr><td rowspan="11">2</td><td rowspan="11">Baud rate setting</td><td rowspan="6">kbit/s</td><td>0: 9.6</td><td rowspan="11">6</td></tr>
<tr><td>1: 19.2</td></tr>
<tr><td>2: 45.45</td></tr>
<tr><td>3: 93.75</td></tr>
<tr><td>4: 187.5</td></tr>
<tr><td>5: 500</td></tr>
<tr><td rowspan="5">Mbit/s</td><td>6: 1.5</td></tr>
<tr><td>7: 3</td></tr>
<tr><td>8: 6</td></tr>
<tr><td>9: 9</td></tr>
<tr><td>10: 12</td></tr>
<tr><td>3</td><td>PZD3</td><td colspan="2">0–65535</td><td>0</td></tr>
<tr><td>4</td><td>PZD4</td><td colspan="2">0–65535</td><td>0</td></tr>
<tr><td>…</td><td>…</td><td colspan="2">0–65535</td><td>0</td></tr>
<tr><td>10</td><td>PZD12</td><td colspan="2">0–65535</td><td>0</td></tr>
</table>

2. Module type

This parameter displays the model of the communication card detected by the inverter. You cannot modify the value of this parameter. If the parameter is not defined, communication between the communication card and inverter cannot be established.

3. Node address

On the PROFIBUS network, each device corresponds to one unique node address. The node address is set through P15.01.

4. GSD file

One device description file named GSD file is required for each PROFIBUS slave station on the PROFIBUS bus. The GSD file is used to describe the characteristics of the PROFIBUS-DP device. The GSD file includes all parameters defined for the device, including the supported bard rate, supported information length, input/output data amount, and definitions of diagnosis data.

You can obtain the type definition files (GSD files) of various masters from INVT's official website and copy the GSD files to the corresponding subdirectories on the configuration tool software. For details about the operation and how to configure the PROFIBUS system, see the instructions for the related system configuration software.

## 2.6 PROFIBUS-DP communication

1. PROFIBUS-DP

PROFIBUS-DP is a distributed input/output (I/O) system. It enables a master to use a large number of peripheral modules and on-site devices. Data transmission is periodic: The master reads information input by a slave and transmits a feedback signal to the slave.

2. SAP

The PROFIBUS-DP system uses the services at the data link layer (Layer 2) through service access points (SAPs). Functions of each SAP are clearly defined. For more information about SAPs, see the related PROFIBUS master user manuals, that is, PROFIdrive—PROFIBUS models or EN50170 standards (PROFIBUS protocol) for variable-speed drives.

3. PROFIBUS-DP information frame data structure

The PROFIBUS-DP system allows fast data exchange between the master and inverter devices. For inverter devices, data is always read and written in the master/slave mode. Inverters always function as slave stations, and one address is clearly defined for each slave station. PROFIBUS transmits 16-bit packets periodically. Figure 2-4 shows the structure of the packet.

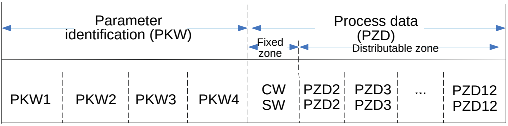



Figure 2-4 PROFIBUS-DP information frame data structure

Parameter zone:

PKW1—Parameter identification

PKW2—Array index number

PKW3—Parameter value 1

PKW4—Parameter value 2

Process data:

CW—Control word (transmitted from the master to a slave. For description, see Table 2-5)

SW—State word (transmitted from a slave to the master. For description, see Table 2-7.)

PZD—Process data (defined by users)

(When the process data is output by the master to a slave, it is a reference value; and when the process data is input by a slave to the master, it is an actual value.)

PZD zone (process data zone): The PZD zone in a communication packet is designed for controlling and monitoring an inverter. The master and slave stations always process the received PZD with the highest priority. The processing of PZD takes priority over that of PKW, and the master and slave stations always transmit the latest valid data on the interfaces.

CWs and SWs

Using CWs is the basic method of the fieldbus system to control inverters. A CW is transmitted by the fieldbus master station to an inverter device. In this case, the EC-TX-103 communication card functions as a gateway. The inverter device responds to the bit code information of the CW and feeds state information back to the master through an SW.

Reference value: An inverter device may receive control information in multiple channels, including analog and digital input terminals, inverter control panel, and communication modules (such as RS485 and EC-TX-103 communication cards). To enable the control over inverter devices through PROFIBUS, you need to set the communication module as the controller of the inverter device.

Actual value: An actual value is a 16-bit word that includes information about inverter device operation. The monitoring function is defined through inverter parameters. The conversion scale of an integer transmitted as an actual value from the inverter device to the master depends on the set function. For more description, see the related inverter operation manual.

**Note:** An inverter device always checks the bytes of a CW and reference value.

Task packet (master station -> inverter)

CW: The first word in a PZD task packet is an inverter CW. The definitions of CWs on the PWM rectifier feedback side are different from those on the inverter side. Table 2-5 and Table 2-6 describe the definitions of CWs on the two sides.

Table 2-5 Goodrive350 series CWs

| Bit | Name | Value | State to be entered/description |
|---|---|---|---|
| 0–7 | Communication-based control command | 1 | Forward running |
| | | 2 | Reverse running |
| | | 3 | Forward inching |
| | | 4 | Reverse inching |
| | | 5 | Decelerating to stop |
| | | 6 | Coasting to stop (emergency stop) |
| | | 7 | Fault reset |
| | | 8 | Inching stopped |
| | | 9 | Pre-excitation |
| 8 | Enable writing | 1 | Enabling writing (mainly through PKW1 to PKW4) |
| 9–10 | Motor group setting | 00 | Motor 1 |
| | | 01 | Motor 2 |
| 11 | Control mode switching | 1 | Enabling the switching between torque control and speed control |
| | | 0 | No switching |
| 12 | Reset power consumption to zero | 1 | Enabling the function for resetting power consumption to zero |
| | | 0 | Disabling the function for resetting power consumption to zero |
| 13 | Pre-excitation | 1 | Enabling pre-excitation |
| | | 0 | Disabling pre-excitation |
| 14 | DC braking | 1 | Enabling DC braking |
| | | 0 | Disabling DC braking |
| 15 | Heartbeat reference | 1 | Enabling heartbeat |
| | | 0 | Disabling heartbeat |

Reference value (REF): The second to twelfth words in a PZD task packet are the main settings. The main frequency settings are provided by the main setting signal source. There is not main frequency setting on the PWM rectifier feedback side, and therefore the corresponding settings are reserved. Table 2-6 describes the settings on the Goodrive350 series inverter side.

Table 2-6 Settings on the Goodrive350 series inverter side

| Function code | Word | Value range | Default value |
|---|---|---|---|
| P15.02 | Received PZD2 | 0: Invalid<br>1: Set frequency (0–Fmax, unit: 0.01 Hz) | 0 |

<table>
<tr><th>Function code</th><th>Word</th><th>Value range</th><th>Default value</th></tr>
<tr><td>P15.03</td><td>Received PZD3</td><td rowspan="10">2: PID reference (0–1000, in which 1000 corresponds to 100.0%)<br>3: PID feedback (0–1000, in which 1000 corresponds to 100.0%)<br>4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01 Hz)<br>6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01 Hz)<br>7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>8: Upper limit of the brake torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>9: Virtual input terminal command, 0x000–0x3FF (corresponding to S8, S7, S6, S5, HDIB, HDIA, S4, S3, S2, and S1 in sequence)<br>10: Virtual output terminal command, 0x00–0x0F (corresponding to RO2, RO1, HDO, and Y1 in sequence)<br>11: Voltage setting (for V/F separation)<br>(0–1000, in which 1000 corresponds to 100.0% of the rated voltage of the motor)<br>12: AO output setting 1 (-1000–+1000, in which 1000 corresponds to 100.0%)<br>13: AO output setting 2 (-1000–+1000, in which 1000 corresponds to 100.0%)<br>14: MSB of position reference (signed number)<br>15: LSB of position reference (unsigned number)<br>16: MSB of position feedback (signed number)<br>17: LSB of position feedback (unsigned number)<br>18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0)</td><td>0</td></tr>
<tr><td>P15.04</td><td>Received PZD4</td><td>0</td></tr>
<tr><td>P15.05</td><td>Received PZD5</td><td>0</td></tr>
<tr><td>P15.06</td><td>Received PZD6</td><td>0</td></tr>
<tr><td>P15.07</td><td>Received PZD7</td><td>0</td></tr>
<tr><td>P15.08</td><td>Received PZD8</td><td>0</td></tr>
<tr><td>P15.09</td><td>Received PZD9</td><td>0</td></tr>
<tr><td>P15.10</td><td>Received PZD10</td><td>0</td></tr>
<tr><td>P15.11</td><td>Received PZD11</td><td>0</td></tr>
<tr><td>P15.12</td><td>Received PZD12</td><td>0</td></tr>
</table>

Response packet (inverter -> master station)

SW: The first word in a PZD response packet is an inverter SW. Table 2-7 describes the definitions of the inverter SWs.

Table 2-7 Goodrive350 series SWs

| Bit | Name | Value | State to be entered/description |
|---|---|---|---|
| 0–7 | Running state | 1 | In forward running |
| | | 2 | In reverse running |
| | | 3 | Stopped |
| | | 4 | Faulty |
| | | 5 | POFF |
| | | 6 | In pre-excitation |
| 8 | Bus voltage established | 1 | Ready to run |
| | | 0 | Not ready to run |
| 9–10 | Motor group feedback | 0 | Motor 1 |
| | | 1 | Motor 2 |
| 11 | Motor type feedback | 1 | Synchronous motor |
| | | 0 | Asynchronous motor |
| 12 | Overload pre-alarm feedback | 1 | Overload pre-alarm generated |
| | | 0 | No overload pre-alarm generated |
| 13 | Run/Stop mode | 0 | Keypad-based control |
| | | 1 | Terminal-based control |
| 14 | | 2 | Communication-based control |
| | | 3 | Reserved |
| 15 | Heartbeat feedback | 1 | Heartbeat feedback |
| | | 0 | No heartbeat feedback |

Actual value (ACT): The second to twelfth words in a PZD task packet are the main actual values. The main actual frequency values are provided by the main actual value signal source.

Table 2-8 Actual state values of the Goodrive350 series

| Function code | Word | Value range | Default value |
|---|---|---|---|
| P15.13 | Transmitted PZD2 | 0: Invalid<br>1: Running frequency (×100, Hz)<br>2: Set frequency (×100, Hz)<br>3: Bus voltage (×10, V)<br>4: Output voltage (×1, V)<br>5: Output current (×10, A)<br>6: Actual output torque (×10, %)<br>7: Actual output power (×10, %)<br>8: Rotating speed of the running (×1, | 0 |
| P15.14 | Transmitted PZD3 | | 0 |
| P15.15 | Transmitted PZD4 | | 0 |
| P15.16 | Transmitted PZD5 | | 0 |
| P15.17 | Transmitted PZD6 | | 0 |
| P15.18 | Transmitted PZD7 | | 0 |
| P15.19 | Transmitted PZD8 | | 0 |
| P15.20 | Transmitted PZD9 | | 0 |
| P15.21 | Transmitted PZD10 | | 0 |

| Function code | Word | Value range | Default value |
|---|---|---|---|
| P15.22 | Transmitted PZD11 | RPM) | 0 |
| P15.23 | Transmitted PZD12 | 9: Linear speed of the running (×1, m/s)<br>10: Ramp frequency reference<br>11: Fault code<br>12: AI1 value (×100, V)<br>13: AI2 value (×100, V)<br>14: AI3 value (×100, V)<br>15: HDIA frequency (×100, kHz)<br>16: Terminal input state<br>17: Terminal output state<br>18: PID reference (×100, %)<br>19: PID feedback (×100, %)<br>20: Rated torque of the motor<br>21: MSB of position reference (signed number)<br>22: LSB of position reference (unsigned number)<br>23: MSB of position feedback (signed number)<br>24: LSB of position feedback (unsigned number)<br>25: State word<br>26: HDIB frequency value (×100, kHz) | 0 |

PKW zone (parameter identification flag PKW1—numerical zone): The PKW zone describes the processing mode of the parameter identification interface. A PKW interface is not a physical interface but a mechanism that defines the transmission mode (such reading and writing a parameter value) of a parameter between two communication ends.

Structure of the PKW zone

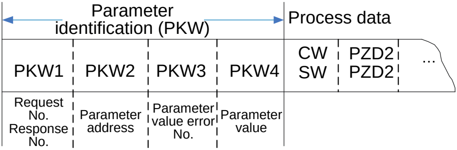



Figure 2-5 Parameter identification zone

In the periodic PROFIBUS-DP communication, the PKW zone consists of four 16-bit words. Table 2-9 describes the definition of each word.

Table 2-9 Definition of each word in the PKW zone

| First word PKW 1 (16 bits) | | |
|---|---|---|
| Bits 15–00 | Task or response identification flag | 0–7 |
| Second word PKW2 (16 bits) | | |
| Bits 15–00 | Basic parameter address | 0–247 |
| Third word PKW3 (16 bits) | | |
| Bits 15–00 | Value (most significant word) of a parameter or error code of the returned value | 00 |
| Fourth word PKW4 (16 bits) | | |
| Bits 15–00 | Value (least significant word) of a parameter | 0–65535 |

**Note:** If the master station requests the value of a parameter, the values in PKW3 and PKW4 of the packet that the master station transmits to the inverter are no longer valid.

Task request and response: When transmitting data to a slave, the master uses a request number, and the slave uses a response number to accept or reject the request. Table 2-10 describes the request and response functions.

Table 2-10 Definition of the task identification flag PKW1

| **Request No. (from the master to a slave)** | | **Response signal** | |
|---|---|---|---|
| **Request No.** | **Function** | **Acceptance** | **Rejection** |
| 0 | No task | 0 | – |
| 1 | Requesting the value of a parameter | 1, 2 | 3 |
| 2 | Modifying a parameter value (one word) [modifying the value only on RAM] | 1 | 3 or 4 |
| 3 | Modifying a parameter value (two words) [modifying the value only on RAM] | 2 | 3 or 4 |
| 4 | Modifying a parameter value (one word) [modifying the value on both RAM and EEPROM] | 1 | 3 or 4 |
| 5 | Modifying a parameter value (two words) [modifying the value only on both RAM and EEPROM] | 2 | 3 or 4 |

The requests #2, #3, and #5 are not supported currently.

Table 2-11 Definition of the response identification flag PKW1

| Response No. (from a slave to the master) | |
|---|---|
| **Response No.** | **Function** |
| 0 | No response |
| 1 | Transmitting the value of a parameter (one word) |
| 2 | Transmitting the value of a parameter (two words) |
| 3 | The task cannot be executed and one of the following error number is returned:<br>1: Invalid command<br>2: Invalid data address<br>3: Invalid data value<br>4: Operation failure<br>5: Password error<br>6: Data frame error<br>7: Parameter read only<br>8: Parameter cannot be modified during inverter running<br>9: Password protection |

PKW examples

Example 1: Reading the value of a parameter

You can set PKW1 to 1 and PKW2 to 10 to read a frequency set through keypad (the address of the frequency set through keypad is 10), and the value is returned in PKW4.

Request (master station -> inverter)

| | PKW1 | | PKW2 | | PKW3 | | PKW4 | | CW | | PZD2 | | PZD3 | | ... | PZD12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Request | 00 | 01 | 00 | 10 | 00 | 00 | 00 | 00 | xx | xx | xx | xx | xx | xx | ... | xx | xx |

0001: Requesting to read a parameter value

0010: Parameter address

Response (inverter -> master station)

| | PKW1 | | PKW2 | | PKW3 | | PKW4 | | CW | | PZD2 | | PZD3 | | … | PZD12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Response | 00 | 01 | 00 | 10 | 00 | 00 | 50 | 00 | xx | xx | xx | xx | xx | xx | … | xx | xx |

5000: Parameter value in address 10

0001: Response (parameter value updated)

Example 2: Modifying the value of a parameter (on both RAM and EEPROM)

You can set PKW1 to 4 and PKW2 to 10 to modify a frequency set through keypad (the address of the frequency set through keypad is 10), and the value to be modified (50.00) is in PKW4.

Request (master station -> inverter)

| | PKW1 | | PKW2 | | PKW3 | | PKW4 | | CW | | PZD2 | | PZD3 | | … | | PZD12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Request | 00 | 04 | 00 | 10 | 00 | 00 | 50 | 00 | xx | xx | xx | xx | xx | xx | | | xx | xx |

5000: Parameter value in address

004: Parameter value to be modified

Response (inverter-> master station)

| | PKW1 | | PKW2 | | PKW3 | | PKW4 | | CW | | PZD2 | | PZD3 | | … | | PZD12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Response | 00 | 01 | 00 | 10 | 00 | 00 | 50 | 00 | xx | xx | xx | xx | xx | xx | … | | xx | xx |

0001: Response (parameter value updated)

PZD examples: The transmission of the PZD zone is implemented through inverter function code settings. For the function codes, see the related INVT inverter operation manual.

Example 1: Reading the process data of an inverter

In this example, PZD3 is set to "8: Rotating speed of the running" through the inverter parameter P15.14. This operation sets the parameter forcibly. The setting remains until the parameter is set to another option.

Response (inverter -> master station)

| | PKW1 | | PKW2 | | PKW3 | | PKW4 | | CW | | PZD2 | | PZD3 | | … | | PZD12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Response | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | 00 | 0A | … | | xx | xx |

Example 2: Writing process data to an inverter device

In this example, PZD3 is set to "2: PID reference" through the inverter parameter P15.03. The parameter specified in each request frame is updated with the information contained in PZD3 until another parameter is specified.

Request (master station -> inverter)

| | PKW1 | | PKW2 | | PKW3 | | PKW4 | | CW | | PZD2 | | PZD3 | | … | | PZD12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Response | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | xx | 00 | 00 | … | | xx | xx |

Subsequently, the information contained in PZD3 is used as tractive force reference in each request frame until another parameter is specified.

# 3 CANopen communication card

## 3.1 Overview

1. Thanks for choosing INVT CANopen communication cards. This manual describes the function specifications, installation, basic operation and settings, and information about the network protocol. To ensure that you install and operate the product properly, read this manual and the communication protocol section in the inverter operation manual carefully before you use the product.
2. This manual only describes how to operate the CANopen communication card and the related commands but does not provide details about the CANopen protocol. For more information about the CANopen protocol, read the related specialized articles or books.
3. This communication card is defined as a CANopen slave station communication card and is used on an inverter that supports CANopen communication.
4. The CANopen communication of this communication card supports access to inverters through process data objects (PDOs) and service data objects (SDOs). PDOs and SDOs are used to read the object dictionary defined by the manufacturer.

## 3.2 Features

1. Supported functions
   - ➢ Supports the CAN2.0A protocol.
   - ➢ Supports CANopen DS301.
2. Supported CANopen services
   - ➢ PDO: Supports four pairs of PDO services (PDO1 TX to PDO4 TX, and PDO1 RX to PDO4 RX), where the PDO1 pair is used to read and write parameters of an inverter, and the PDO2 to PDO4 pairs are used to control and obtain the actual parameter values of the inverter in real time.
   - ➢ SDO: SDO information adopts the "client/server" mode and is used to configure slave nodes and provide access to the object dictionary of each node.
   - ➢ Supports the emergency service.
   - ➢ Supports node protection (NMT Node Guarding).
   - ➢ Supports heartbeat packets (Heartbeat Producer).
   - ➢ Supports network management (NMT).
     - • Supports NMT module control.

- Supports NMT broadcast addresses.
- Supports NMT error control.
- Supports boot-up.

➢ Supports SYNC (1–240).

➢ Supports asynchronous transmission of 254 and 255.

➢ Supports disabled time.

➢ Supports event timers.

➢ Supports manufacturer-defined object dictionary. You can use SDOs to control and obtain the actual parameter values of an inverter in real time.

3. Non-supported CANopen services

➢ Saves object dictionary parameters at power outage

➢ Time stamp service

4. Supported CANopen addresses and baud rates

Table 3-1 Supported addresses and baud rates

| Item | Supported specification |
| --- | --- |
| Address | 1–127 (decimal) |
| Baud rate | 1000 kbps |
| | 800 kbps |
| | 500 kbps |
| | 250 kbps |
| | 125 kbps |
| | 100 kbps |
| | 50 kbps |
| | 20 kbps |

**Note:** To enable the CANopen functions (except the CANopen communication timeout fault time and baud rate), you need only to select the related PROFIBUS channels. If modification is made on the inverter operation manual, the operation is subject to the CANopen channel, without prior notice in this manual.

## 3.3 Electrical wiring

Use shielding wires in the bus cable, if possible. It is recommended that you connect the shielding wire to the CANG terminal of the inverter. When the communication card functions as the terminal slave, it is recommended that you switch on the terminal resistor. Figure 3-1 shows the electrical wiring.

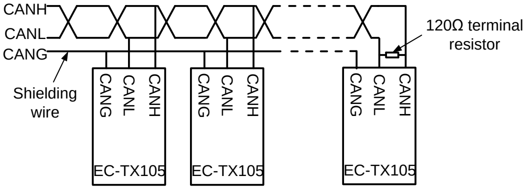



Figure 3-1 Electrical wiring diagram

## 3.4 Communication

### 3.4.1 Packet format

CAN2.0A packets are used transmit data between the master station and bus nodes through data frames.

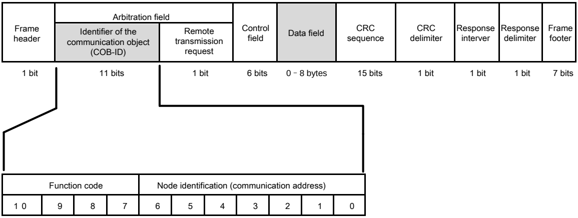



Figure 3-2 Packet structure

| Communication object | Function code (binary) | COB-ID (hexadecimal) |
|---|---|---|
| NMT | 0 | 0x00 |
| SYNC | 1 | 0x80 |
| EMERGENCY | 1 | 0x81–0xFF |
| PDO1 Tx | 11 | 0x181–0x1FF |
| PDO1 Rx | 100 | 0x201–0x27F |
| PDO2 Tx | 101 | 0x281–0x2FF |
| PDO2 Rx | 110 | 0x301–0x37F |
| PDO3 Tx | 111 | 0x381–0x3FF |
| PDO3 Rx | 1000 | 0x401–0x47F |
| PDO4 Tx | 1001 | 0x481–0x4FF |
| PDO4 Rx | 1010 | 0x501–0x57F |

| Communication object | Function code (binary) | COB-ID (hexadecimal) |
|---|---|---|
| SDO Tx | 1011 | 0x581–0x5FF |
| SDO Rx | 1100 | 0x601–0x67F |
| Node protection | 1110 | 0x701–0x77F |

COB-IDs vary according to communication address, but for one command, the COB-IDs are within a certain range.

**Note:** The commands described in this manual are all data frames if it is not specified that they are remote frames.

### 3.4.2 CANopen state transition

The start sequence defined in the CANopen communication protocol is supported. Figure 3-3 shows the NMT state transition diagram.

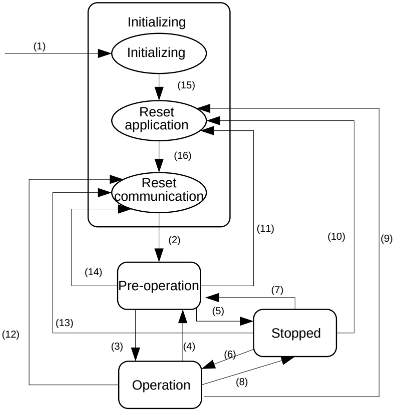



Figure 3-3 NMT state diagram

Table 3-2 NMT state transition

| State transition | Required triggering event |
|---|---|
| (1) | Automatic initialization after power-on |
| (2) | Automatic change after initialization |
| (3), (6) | Command of the NMT master station for starting a remote node |
| (4), (7) | Command of the NMT master station for entering the |

| State transition | Required triggering event |
|---|---|
| | pre-operation state |
| (5), (8) | Command of the NMT master station for entering the stopped state |
| (9), (10), (11) | Command of the NMT master station for resetting a remote node |
| (12), (13), (14) | Command of the NMT master station for resetting a remote node communication parameter |

Different services are supported in different states, as described in Table 3-3.

Table 3-3 Services supported in various NMT states

| Service | Pre-operation state | Operation state | Stopped state |
|---|---|---|---|
| PDOs | No | Yes | No |
| SDOs | Yes | Yes | No |
| SYNC packets | Yes | Yes | No |
| Emergency packets | Yes | Yes | No |
| Network management | Yes | Yes | No |
| Error control | Yes | Yes | Yes |

### 3.4.3 Management service command (NMT)

This function is used by the master station to control the NMT states of slave station nodes.

- Command

Master station -> slave station

| COB-ID | Byte0 | Byte1 |
|---|---|---|
| 0x000 | Command specifier (CS) | Node-ID (Node ID) |

- Description

In this command, the COB-ID is 0×00. If Node-ID is set to 0, the command is broadcast to all CANopen slave stations, and each slave station must execute the NMT command. Table 3-4 describes the function of each CS.

Table 3-4 Function of each CS

| NMT CS | NMT service (control action) |
|---|---|
| 0x01 | Starts a slave station device. |
| 0x02 | Stops a slave station device. |

| NMT CS | NMT service (control action) |
|---|---|
| 0x80 | Enables a slave station to enter the pre-operation state. |
| 0x81 | Resets a slave station. |
| 0x82 | Resets communication of a node. |

- Example

For example, the command to enable EC-TX105, whose node ID is 3, to enter the pre-operation state is described as follow.

| COB-ID | Byte0 | Byte1 |
|---|---|---|
| 0x000 | 0x80 | 0x03 |

For another example, the command to start all EC-TX105 nodes on the CANopen network is described as follows.

| COB-ID | Byte0 | Byte1 |
|---|---|---|
| 0x000 | 0x01 | 0x00 |

### 3.4.4 Node protection (NMT Node Guarding)

By using the node protection service, the NMT master node can detect the current state of each node.

- **Command**

Request: Master station (remote frame) –> slave station

| COB-ID | No data |
|---|---|
| 0x700 + Node-ID | |

Response: Slave station -> master station

| COB-ID | Byte0 (state value) |
|---|---|
| 0x700 + Node-ID | Bit 7: Triggering bit; Bits 0 to 6: State |

- Description

The most significant bit (MSB) bit 7 of Byte0 (state value) in the response command is the triggering bit, that is, the value of bit 7 is alternated between 0 and 1 each time when the slave station transmits a response frame to distinguish frames. Bits 0 to 6 indicate the state of the slave station. Table 3-5 describes the state values and their corresponding state.

Table 3-5 State values and their corresponding states

| State value (Byte0: Bits 0–6) | State |
|---|---|
| 0x00 | Initializing |
| 0x04 | Stopped |
| 0x05 | Operation |
| 0x7F | Pre-operational |

- Example

For example, the command for the master station to detect the state of slave station 3.

Master station (remote frame) -> slave station

| COB-ID | No data |
|---|---|
| 0x703 | / |

After receiving the node protection command transmitted by the master station, the slave station transmits the following command response to the master station.

| COB-ID | Byte0 (state value) |
|---|---|
| 0x703 | 0x85 |

In the command, bit 7 of Byte0 is 1, and the state value is 0×05, indicating that slave station 3 is in the operation state. If receiving another node protection command, the slave station transmits a command frame in which the state value is 0×05 to the master station, and the value of bit 7 is alternated to 0.

### 3.4.5 Heartbeat packet (Heartbeat Producer)

In some cases, the master station requires that a slave station automatically transmits a frame of heartbeat packets at an interval, so that it can learn the state of the slave station in real time. The interval parameter (data length: 16 bits; unit: ms) is defined in the object dictionary 0x1017. If the interval is set to 0, the slave station does not transmit heartbeat packets. For this CANopen communication card, the interval is set to 0 by default.

- Command

Slave station -> master station

| COB-ID | Byte0 |
|---|---|
| 0x700 + Node-ID | State value |

- Description

The heartbeat packets are in the same format with the node protection response frames. The difference between them is that no triggering bit alternation is performed for heartbeat packets (the triggering bit is always 0). Table 3-5 describes the state values.

- Example

For example, if slave station 3 is in the operation state and the interval parameter in 0x1017 is set to 100, slave station 3 transmits a frame of heartbeat packets every 100 ms.

| COB-ID | Byte0 |
|---|---|
| 0x703 | 0x05 |

SDOs can be used to disable heartbeat packets, transmitting 2B 17 10 00 00 00 00 00 (setting the interval to 0).

**Note:** On the communication card, node protection and heartbeat packets cannot be used simultaneously.

### 3.4.6 Start packet (NMT Boot-up)

After being initialized (booted up), the communication card transmits a start packet.

- Command

Slave station -> master station

| COB-ID | Byte0 |
|---|---|
| 0x700 +Node-ID | 0x00 |

- Example

For example, after being initialized, the communication card whose node ID is 3 transmits the following start packet.

| COB-ID | Byte0 |
|---|---|
| 0x703 | 0x00 |

### 3.4.7 Synchronous packet object (SYNC)

Generally, SYNC signals are transmitted by the CANopen master station cyclically. A SYNC signal does not contain any data and is used mainly to request PDO Tx of a slave station node of the synchronous transmission type. 0x1005 in the object dictionary defines COB-IDs of the objects that receive synchronous packets, and they are set to 0x80 in the CANopen pre-defined connection set. For PDO Tx, the transmission types of 1 to 240 indicate synchronous transmission.

- Command

Master station -> slave station

| COB-ID | No data |
|---|---|
| 0x80 | / |

### 3.4.8 Emergency packet object (EMCY)

This packet is transmitted when an internal error occurs on the communication card or inverter, or an error is deleted.

- Command

Slave station -> master station

<table>
<tr><th>COB-ID</th><th>Byte0</th><th>Byte1</th><th>Byte2</th><th>Byte3</th><th>Byte4</th><th>Byte5</th><th>Byte6</th><th>Byte7</th></tr>
<tr><td rowspan="2">0x80 + Node-ID</td><td colspan="2">Emergency error code</td><td rowspan="2">Error register</td><td colspan="5">Inverter error code</td></tr>
<tr><td>LSB</td><td>MSB</td><td>bit7-0</td><td>bit15-8</td><td>bit23-16</td><td>bit31-24</td><td>bit39-32</td></tr>
</table>

- Description

An emergency error code is two bytes. Byte0 is the least significant byte (LSB), and Byte1 is the most significant byte (MSB). An inverter error code is five bytes. Byte3 is the LSB, and Byte7 is the MSB.

An emergency error code indicates the type of the current error, as described in Table 3-6. The error register stores the type of the current error. You can determine the error type indicated by the current emergency packet according to the value stored in the register. Table 3-7 describes the indication of the bits of the error register. For information about the inverter error codes, see the inverter operation manual. The function code P07.27 in Appendix B describes the error codes of the GD350 inverter.

Table 3-6 Indication of the emergency error codes

| Emergency error code (hex) | Code function description |
|---|---|
| 00xx | Error reset or no error |
| 10xx | Generic error |
| 20xx | Current |
| 21xx | Current error on the, device input side |
| 22xx | Current error inside the device |
| 23xx | Current error on the device output side |
| 30xx | Voltage error |
| 31xx | Mains voltage |
| 32xx | Voltage inside the device |
| 33xx | Output voltage |
| 40xx | Temperature |
| 41xx | Ambient temperature |
| 42xx | Device temperature |
| 50xx | Device hardware |
| 60xx | Device software |
| 61xx | Internal software |
| 62xx | User software |
| 63xx | Data set |
| 70xx | Additional modules |
| 80xx | Monitoring |
| 81xx | Communication error |
| 8110 | CAN overrun |
| 8120 | Error passive |
| 8130 | Life guard Error or heartbeat error |
| 8140 | Recovered from Bus-Off |
| 82xx | Protocol error |

| Emergency error code (hex) | Code function description |
|---|---|
| 8210 | PDO not processed due to length error |
| 8220 | Length exceeded |
| 90xx | External error |
| F0xx | Additional functions |
| FFxx | Device specific |

Table 3-7 Indication of the bits of the error register

| Error register (bit) | Error type |
|---|---|
| 0 | Generic error or no error |
| 1 | Current error |
| 2 | Voltage error |
| 3 | Temperature error |
| 4 | Communication error |
| 5 | Device description error |
| 6 | Reserved (=0) |
| 7 | Manufacturer-defined error |

- Example

For example, if the "inverter unit phase U protection (OUT1)" fault occurs on the GD350 inverter whose node ID is 3, and the fault type is 1 (that is, the inverter error code is 1), the communication card transmits the following emergency packet.

| COB-ID | Emergency error code | | Error register | Inverter error code | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x83 | 0x00 | 0x30 | 0x04 | 0x01 | 0x00 | 0x00 | 0x00 | 0x00 |

As you can see in the command, the emergency error code is 0x3000, indicating a voltage error. The error register is 0x04, that is, the second bit is "1", indicating a voltage error. The device error code is 0x0000000001. See the GD350 inverter operation manual, and you can find that the error code 1 indicates the "inverter unit phase U protection (OUT1)" fault.

After the fault is reset, the communication card transmits the following emergency packet to notify the master station that the slave station is no longer faulty.

| COB-ID | Emergency error code | | Error register | Inverter error code | | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x83 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

### 3.4.9 Service data object (SDO)

SDOs are mainly used to transmit non-time key data. By using SDOs, the master station can read data from and write data to the object dictionary of a device.

- Command

Request: master station -> slave station

| COB-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 0x600+NodeID | Request code | Object index | | Subindex | Response data | | | |
| | | LSB | MSB | | bit7-0 | bit15-8 | bit23-16 | bit31-24 |

Response: Slave station -> master station

| COB-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 0x580+NodeID | Response code | Object index | | Subindex | Response data | | | |
| | | LSB | MSB | | bit7-0 | bit15-8 | bit23-16 | bit31-24 |

- Description

An object index is two bytes. Byte1 is the LSB, and Byte2 is the MSB. For information about the indexes and subindexes, see the object dictionary in the appendix. Request codes include request codes for reading and those for writing.

Request codes for writing vary according to the character length of items in the object dictionary, and the request code for reading are 0×40. See Table 3-8.

Response codes indicating successful reading vary according to the character length of items in the object dictionary, and the response code indicating successful writing are 0×60. The response codes indicating reading failure and writing failure are both 0x80. See Table 3-9.

Table 3-8 SDO request codes and requested data

| Request code type | Request code | Command description | Requested data | | | |
|---|---|---|---|---|---|---|
| | | | Byte4 | Byte5 | Byte6 | Byte7 |
| Write | 0x23 | Writes 4-byte data | bit7-0 | bit15-8 | bit23-16 | bit31-24 |
| | 0x2B | Writes 2-byte data | bit7-0 | bit15-8 | - | - |
| | 0x2F | Writes 1-byte data | bit7-0 | - | - | - |
| Read | 0x40 | Reads data | - | - | - | - |

Table 3-9 SDO response codes and response data

| Response code type | Response code | Command description | Response data | | | |
|---|---|---|---|---|---|---|
| | | | Byte4 | Byte5 | Byte6 | Byte7 |
| Read | 0x43 | Reads 4-byte data | bit7-0 | bit15-8 | bit23-16 | bit31-24 |
| | 0x4B | Reads 2-byte data | bit7-0 | bit15-8 | - | - |
| | 0x4F | Reads 1-byte data | bit7-0 | - | - | - |
| Write | 0x60 | Writing succeeds | - | - | - | - |
| Read/write | 0x80 | Reading/writing fails | Interruption error code | | | |
| | | | bit7-0 | bit15-8 | bit23-16 | bit31-24 |

**Note:** The symbol "-" in Table 3-8 and Table 3-9 indicates that the byte is reserved and provides no function.

Table 3-10 describes the interruption error codes.

Table 3-10 Interruption error codes

| Interruption code | Code function description |
|---|---|
| 0503 0000 | Triggering bit not alternated |
| 0504 0000 | SDO protocol times out |
| 0504 0001 | Invalid or unknown client/server |
| 0504 0002 | Invalid block size |
| 0504 0003 | Invalid sequence number |
| 0504 0004 | CRC error |
| 0504 0005 | Memory overflow |
| 0601 0000 | No access to the object |
| 0601 0001 | Attempts to read a write-only object |
| 0601 0002 | Attempts to write information to a read-only object |
| 0602 0000 | Object cannot be found in the object dictionary |
| 0604 0041 | Object cannot be mapped to PDO |
| 0604 0042 | Number and length of the object to be mapped exceeds the PDO length |
| 0604 0043 | Common parameter incompatibility |
| 0604 0047 | Common internal incompatibility of the device |
| 0606 0000 | Object access failure caused by hardware error |
| 0607 0010 | Data type not matched; service parameter length not matched |
| 0609 0011 | Subindex cannot be found in the object dictionary |
| 0609 0030 | Parameter value range exceeded |
| 0609 0031 | Written parameter value too large |
| 0609 0032 | Written parameter value too small |
| 0609 0036 | Max. value less than Min. value |
| 0800 0000 | Common error |
| 0800 0020 | Data failed to be transmitted or stored in the application |
| 0800 0021 | Data failed to be transmitted or stored in the application due to device control |
| 0800 0022 | Data failed to be transmitted or stored in the application due to the current state of the device |
| 0800 0023 | Error occurs dynamically on the object dictionary or object dictionary cannot be found |

- Example

For example, slave station 3 reads data from and writes data to the object whose index is 0x1801 and subindex is 03. (The object whose index is 0x1801 and subindex is 03 indicates the disabled time of PDO2 Tx. For more information, see Appendix A.)

Write operation example: To modify the disabled time of PDO2 Tx to 1000 ms, the master station transmits the following write operation command.

| COB-ID | Request code | Object index | | Subindex | Requested data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x603 | 0x2B | 0x01 | 0x18 | 0x03 | 0xe8 | 0x03 | 0x00 | 0x00 |

After receiving the command transmitted by the master station, the slave station transmits the following command response if the modification is successful.

| COB-ID | Response code | Object index | | Subindex | Response data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x583 | 0x60 | 0x01 | 0x18 | 0x03 | 0x00 | 0x00 | 0x00 | 0x00 |

Read operation example: To read the disabled time of PDO2 Tx, the master station transmits the following read operation command.

| COB-ID | Request code | Object index | | Subindex | Requested data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x603 | 0x40 | 0x01 | 0x18 | 0x03 | 0x00 | 0x00 | 0x00 | 0x00 |

After receiving the command transmitted by the master station, the slave station transmits the following command response if the current disabled time of PDO2 Tx is 1000 ms.

| COB-ID | Response code | Object index | | Subindex | Response data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x583 | 0x43 | 0x01 | 0x18 | 0x03 | 0xe8 | 0x03 | 0x00 | 0x00 |

Read/write error example: The master station transmits the following read operation command to read an object (whose index is 0x6000 and subindex is 0x00) that cannot be found.

| COB-ID | Request code | Object index | | Subindex | Requested data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x603 | 0x40 | 0x00 | 0x60 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

The object cannot be found, and therefore the slave station transmits the following read/write error command response.

| COB-ID | Response code | Object index | | Subindex | Response data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x583 | 0x80 | 0x00 | 0x60 | 0x00 | 0x00 | 0x00 | 0x02 | 0x06 |

The error code in the response is 0x06020000, indicating that "Object cannot be found in the object dictionary".

## 3.5 Process data object (PDO)

The communication card provides four PDO Tx commands (whose indexes are 0x1800 to 0x1803) and four PDO Rx commands (whose indexes are 0x1400 to 0x1403). PDO Rx is a PDO command transmitted by the master station to a slave station, that is, it is a master station command. PDO Tx is a PDO command transmitted by a slave station to the master station.

The CW, SW, setting, and return value of each PDO of the communication card are all defined with a "manufacturer-defined object dictionary". In this way, the process data of an inverter can be monitored not only through PDOs but also through SDOs. For more information, see the next chapter. Each PDO command is labeled with "manufacturer-defined object dictionary" in the format of 0xXXXX.HH, where XXXX indicates an index, HH indicates a subindex, and both of them are hexadecimal.

### 3.5.1 Triggering mode of PDO Tx

Each PDO Tx is defined with a transmission type, disabled time, and event timer. The corresponding subindex of the transmission type is 0x02, that of the disabled time is 0x03, and that of the event timer is 0x05. Therefore, the object dictionary index corresponding to PDO2 Tx is 0x1801, and the subindex is 0x02. The same principle applies to other PDO Tx commands. For more information, see Appendix A.

Synchronous triggering: When the transmission type is set to 1 to 240, PDO Tx is synchronous transmission. For example, if you set the transmission type of PDO2 Tx to $n$ ($1 \le n \le 240$), a slave station transmits one PDO2 Tx command every time after it receives $n$ synchronous packet objects. The same principle applies to other PDO Tx commands.

Asynchronous triggering (254): When the value of the event timer is not zero, a slave station transmits PDO Tx commands periodically. For example, if the event timer of PDO2 Tx is set to 200, the slave station transmits a PDO2 Tx command at the interval of 200 ms. When the value of the event timer is zero, the slave station transmits a PDO Tx command once the corresponding PDO Tx data changes, and the transmission interval is subject to the disabled time. A PDO Tx packet can be transmitted only once in the disabled time, which effectively reduces the load of the bus. When the disabled time is set to a period shorter than 50 ms, 50 ms is used as the disabled time.

Asynchronous triggering (255): When the value of the event timer is not zero, a slave station transmits PDO Tx commands periodically. For example, if the event timer of PDO2 Tx is set to 200, the slave station transmits a PDO2 Tx command at the interval of 200 ms. When the value of the event timer is zero, the slave station transmits a PDO Tx command once a corresponding PDO Rx command is received. For example, after receiving a PDO2 Rx command, the slave station transmits a PDO2 Tx command.

Table 3-11 Triggering modes supported by the communication card

| Triggering mode | Transmission type (decimal) | Event triggering | PDO1 TX | PDO2 TX | PDO3 TX | PDO4 TX |
|---|---|---|---|---|---|---|
| Synchronous | 1–240 | / | Non-supported | Supported | Supported | Supported |
| Asynchronous | 254 | Event timer | Non-supported | Supported | Supported | Supported |
| | | Disabled time | Non-supported | Supported | Supported | Supported |
| | 255 | Event timer=0 | Supported | Supported | Supported | Supported |
| | | Event timer=0 | Non-supported | Supported | Supported | Supported |

Table 3-12 Default PDO Tx settings of the communication card

| | PDO1 TX | PDO2 TX | PDO3 TX | PDO4 TX |
|---|---|---|---|---|
| **Transmission type** | 255 | 254 | 254 | 254 |
| **Event timer (ms)** | 0 | 0 | 0 | 0 |
| **Disabled time (ms)** | 500 | 500 | 500 | 500 |

For how to set the triggering type of PDO Tx, see the description of SDO commands.

### 3.5.2 PDO1

PDO1 is used to read and write parameters of the inverter. The function of PDO1 is similar to that of an SDO. SDOs are used to read and write objects of an object dictionary, and PDO1 is used to read and write parameters of the inverter.

**Note:** PDO1 Tx support only the transmission type of asynchronous transmission 255. Do not set it to other transmission types, and do not try to set the event timer to periodically transmits PDO1 Tx to the master station.

### 3.5.2.1 PDO1 Rx

- Command

Request: Master station –> slave station

| COB-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 |
|---|---|---|---|---|---|---|
| 0x200+NODE-ID | Request code | | Parameter address | | Requested data | |
| | 0x2100.00 | | 0x2100.01 | | 0x2100.02 | |

- Description

A request code is two bytes. Byte0 is the LSB, and Byte1 is the MSB. The manufacturer defines the index 0x2100 and subindex 0x00 for the request codes. Table 3-13 describes the functions of the request codes.

Table 3-13 Functions of the request codes

| Request code | Function |
|---|---|
| 0 | No task |
| 1 | Reading the value of a parameter |
| 2 | Modifying a parameter value [modifying the value only on RAM] |
| 4 | Modifying a parameter value [modifying the value only on both RAM and EEPROM] (reserved) |

A parameter address is two bytes. Byte2 is the LSB, and Byte3 is the MSB. It indicates the address of the parameter to be read or modified.

GD series inverter function code address representation rules: The MSB is the hexadecimal form of the number before the dot mark, and LSB is that of the number behind the dot mark. Take P10.01 as an example, the number before the dot mark is 10, that is, the MSB of the parameter address is 0×0A; and the number behind the dot mark is 01, that is, the LSB is 0×01. Therefore, the function code address is 0×0A01.

Table 3-14 GD series inverter parameter addresses

| Function code | Name | Detailed parameter description | Default value | Modify |
|---|---|---|---|---|
| P10.00 | Simple PLC mode | 0: Stops after running once<br>1: Keeps running in the final value after running once<br>2: Cyclic running | 0 | ○ |
| P10.01 | Simple PLC memory selection | 0: Not saving data at power outage<br>1: Saving data at power outage | 0 | ○ |

Inverter parameter address representation rules: You can see the function code in the function parameter list in the inverter operation manual. The hexadecimal form of the value corresponding to the function code is the parameter address. For example, the value corresponding to the function code P13.14 is 1314, and therefore the parameter address of the function code is 0×522 (that is, 1314 in the decimal form).

A piece of requested data is two bytes. Byte4 is the LSB, and Byte5 is the MSB. It indicates the data to be modified. When the command is transmitted for reading data, the requested data is not used.

**Note:** The data domain of PDO1 Rx must be six bytes. Otherwise, the communication card reports an emergency packet.

### 3.5.2.2 PDO1 Tx

- Command

Response: Slave station -> master station

<table>
<tr><th>COB-ID</th><th>Byte0</th><th>Byte1</th><th>Byte2</th><th>Byte3</th><th>Byte4</th><th>Byte5</th><th>Byte6</th><th>Byte7</th></tr>
<tr><td rowspan="2">0x180+NODEID</td><td colspan="2">Response code</td><td colspan="2">Error code</td><td colspan="2">Response data</td><td>0x00</td><td>0x00</td></tr>
<tr><td colspan="2">0x2000.00</td><td colspan="2">0x2000.01</td><td colspan="2">0x2000.02</td><td>-</td><td>-</td></tr>
</table>

- Description

Byte6 and Byte7 are reserved and both are 0x00.

A response code is two bytes. Byte0 is the LSB, and Byte1 is the MSB. Table 3-15 describes the functions of the response codes.

Table 3-15 Functions of the response codes

| Response code | Function |
|---|---|
| 0 | No response |
| 1 | Reading or writing succeeds |
| 3 | A reading or writing error occurs. Table 3-16 describes the error codes. |

A piece of response data is four bytes. Byte4 is the LSB, and Byte7 is the MSB. When a write command is responded, the response data is the data to be modified; and when a read command is responded, the response data is the data to be read.

An error code is two bytes. Byte2 is the LSB, and Byte3 is the MSB. Error codes are valid only when the response code is 3. An error code indicates the reason why it fails to respond to PDO1 Rx. Table 3-16 describes the definitions of the error codes.

Table 3-16 Error codes

| Code | Name | Definition |
|---|---|---|
| 00H | No error | / |
| 01H | Invalid command | The operation corresponding to the request code is not allowed to be executed. The possible causes are as follows:<br>・ The function code is applicable only on new devices and is not implemented on this device.<br>・ The slave station is in the faulty state when processing this request. |
| 02H | Invalid data address | For a slave device, the data address in the request of the master station is not allowed. In particular, the combination of the register address and the number of the to-be-transmitted bytes is invalid. |
| 03H | Invalid data value | The received data domain contains a value that is not allowed. The value indicates the error of the remaining structure in the combined request.<br>**Note:** It does not mean that the data item submitted for storage in the register includes a value unexpected by the program. |
| 04H | Operation failure | The parameter is set to an invalid value in the write operation. For example, a function input terminal cannot be set repeatedly. |
| 05H | Password error | The password entered in the password verification address is different from that set by the user. |
| 06H | Data frame error | The length of the data frame transmitted by the upper computer is incorrect, or in the RTU format, the value of the CRC check bit is inconsistent with the CRC value calculated by the lower computer. |
| 07H | Parameter read-only | The parameter to be modified in the write operation of the master station is a read-only parameter. |
| 08H | Parameter cannot be modified in running | The parameter to be modified in the write operation of the master station cannot be modified during the running of the inverter. |
| 09H | Password protection | A user password is set, and the master station does not provide the password to unlock the system when performing a read or write operation. The error of system locked is reported. |

- Example of PDO1

The inverter a GD350 series inverter, and the slave station address is 3. Assume that you want to set the function code P15.13 of the inverter to 1.

Command analysis: The parameter address of P15.13 is 0×0F0D. According to the protocol, the request code of PDO1 Rx is 0×02, the parameter address is 0x0F0D, and the requested data is 0x01, and therefore PDO1 Rx transmitted by the master station is as follows.

| COB-ID | Request code | | Parameter address | | Requested data | |
|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 |
| 0x203 | 0x02 | 0x00 | 0x0D | 0x0F | 0x01 | 0x00 |

If the inverter parameter is successfully modified, the following PDO1 Tx command is returned.

| COB-ID | Response code | | Error code | | Response data | | - | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x183 | 0x01 | 0x00 | 0x00 | 0x00 | 0x01 | 0x00 | 0x00 | 0x00 |

### 3.5.3 PDO2 Rx

PDO2 Rx is used to modify CWs and real-time process data (setting 1, setting 2, and setting 3) of an inverter. A CW is used to control the start and stop of an inverter, and settings are used to control the real-time running values of the inverter, such as set frequency.

- Command

Master station -> slave station

| COB-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 0x300+NODEID | CW | | Setting 1 | | Setting 2 | | Setting 3 | |
| | 0x2101.00 | | 0x2100.03 | | 0x2100.04 | | 0x2100.05 | |

- Description

A CW is two bytes. Byte0 is the LSB, and Byte1 is the MSB. Table 3-17 describes the definitions of the GD350 series inverter CWs.

Table 3-17 Definitions of the GD350 series inverter CWs

| Bit | Name | Value | Description |
|---|---|---|---|
| 0–7 | Communication-based control command | 1 | Forward running |
| | | 2 | Reverse running |
| | | 3 | Forward inching |
| | | 4 | Reverse inching |
| | | 5 | Stop |

| Bit | Name | Value | Description |
|---|---|---|---|
| | | 6 | Coast to stop (emergency stop) |
| | | 7 | Fault reset |
| | | 8 | Stop inching |
| 8 | Enable write | 1 | Enable writing (mainly through PKW1 to PKW4) |
| 9–10 | Motor group setting | 00 | Motor 1 |
| | | 01 | Motor 2 |
| 11 | Control mode switching | 1 | Enable torque/speed control switching |
| | | 0 | Disable switching |
| 12 | Reset power consumption to zero | 1 | Enable |
| | | 0 | Disable |
| 13 | Pre-excitation | 1 | Enable |
| | | 0 | Disable |
| 14 | DC braking | 1 | Enable |
| | | 0 | Disable |
| 15 | Heartbeat reference | 1 | Enable |
| | | 0 | Disable |

The function of each setting can be set through the corresponding function code of the inverter. The setting method is the same as that for "received PZD" in PROFIBUS communication. For details, see the inverter operation manual. Setting 1, setting 2, and setting 3 correspond to received PZD2, received PZD3, and received PZD4, respectively. To set the function of setting 1 to "Set frequency", you need only to set "Received PZD2" to "1: Set frequency". The same principle applies to other settings. When multiple settings are enabled, the failure to set one setting (for example, the set value exceeds the setting range) does not affect the setting of other settings.

- Example

Assume that the inverter is a GD350 series inverter, the slave station address is 3, you control the running of the inverter through CANopen communication, and you want to set the running frequency to 50 Hz through CANopen communication.

Command analysis: You need to set the inverter start mode and frequency reference mode to CANopen communication (P00.01=2, P00.02=1, P00.06=9) first. In this example, use Setting 2 to set the running frequency (P15.03=1, that is, set Received PZD3 to "1: Set frequency").

When a CW is 0×01, it indicates that the inverter is to be run. To set the frequency to 50 Hz, you need to set Setting 2 to 5000, that is, 0x1388.

The PDO2 Rx command transmitted by the master station is as follows.

| COB-ID | CW | | Setting 1 | | Setting 2 | | Setting 3 | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x303 | 0x01 | 0x00 | 0x00 | 0x00 | 0x88 | 0x13 | 0x00 | 0x00 |

### 3.5.4 PDO2 Tx

PDO2 Tx is a command transmitted by an inverter to the master station. It contains a SW and real-time process data (Returned value 1, returned value 2, and returned value 3). A SW is used to notify of the state of the inverter, and returned values are used to transmit the real-time running values of inverter, such as running frequency.

The default transmission type of PDO2 Tx is 254, and therefore PDO2 Tx is transmitted once data corresponding to a SW or returned value changes.

- Command

Slave station -> master station

| COB-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 0x280+NODEID | SW | | Returned value 1 | | Returned value 2 | | Returned value 3 | |
| | 0x2001.00 | | 0x2000.03 | | 0x2000.04 | | 0x2000.05 | |

- Description

A SW is two bytes. Byte0 is the LSB, and Byte1 is the MSB. Table 3-18 describes the definitions of the GD350 series inverters SWs. For inverter of other series, see the corresponding inverter operation manual.

Table 3-18 Definitions of the GD350 series inverters SWs

| Bit | Name | Value | Description |
|---|---|---|---|
| 0–7 | Running state | 1 | In forward running |
| | | 2 | In reverse running |
| | | 3 | Stopped |
| | | 4 | Faulty |
| | | 5 | POFF |
| 8 | Bus voltage established | 1 | Ready to run |
| | | 0 | Not ready to run |
| 9–10 | Motor group feedback | 0 | Motor 1 |
| | | 1 | Motor 2 |
| 11 | Motor type feedback | 1 | Synchronous motor |
| | | 0 | Asynchronous motor |
| 12 | Overload pre-alarm feedback | 1 | Overload pre-alarm generated |
| | | 0 | No overload pre-alarm generated |
| 13–14 | Run/stop mode | 0 | Keypad-based control |

| Bit | Name | Value | Description |
|---|---|---|---|
| | | 1 | Terminal-based control |
| | | 2 | Communication-based control |
| | | 3 | Reserved |
| 15 | Heartbeat feedback | 1 | Heartbeat feedback |
| | | 0 | No heartbeat feedback |

The function of each returned value can be set through the corresponding function code of the inverter. The setting method is the same as that for "transmitted PZD" in PROFIBUS communication. For details, see the inverter operation manual. Returned value 1, returned value 2, and returned value 3 correspond to transmitted PZD2, transmitted PZD3, and transmitted PZD4, respectively. To set the function of returned value 1 to "Running frequency", you need only to set "Transmitted PZD2" to "1: Running frequency". The same principle applies to other returned values. Multiple returned values can be enabled simultaneously.

- Example

Assume that the inverter is a GD350 series inverter, the slave station address is 3, the inverter is running, and the running frequency is 50.00 Hz. Returned value 1 is set to "Running frequency", returned value 2 is set to "Output voltage", and returned value 3 is set to no function.

Command analysis: You need to set returned value 1 to the running frequency of the inverter (P15.13=1), returned value 2 to the output voltage of the inverter (P15.14=4), and returned value 3 to invalid (P15.15=0) first.

The inverter is running and the bus voltage has been established, and therefore the SW is 0x0101. The running frequency is 50.00 Hz, and therefore returned value 1 is 5000, that is, 0x1388. If the output voltage is 380 V, returned value 2 is 0x017C.

The PDO2 Tx command transmitted by the inverter is as follows.

| COB-ID | SW | | Returned value 1 | | Returned value 2 | | Returned value 3 | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x283 | 0x01 | 0x01 | 0x88 | 0x13 | 0x7C | 0x01 | 0x00 | 0x00 |

### 3.5.5 PDO3 Rx and PDO4 Rx

PDO3 Rx and PDO4 Rx are used to modify the real-time process data of an inverter, such as set frequency.

- PDO3 Rx command

Master station -> slave station

| COB-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 0x400+NODEID | Setting 4 | | Setting 5 | | Setting 6 | | Setting 7 | |
| | 0x2100.06 | | 0x2100.07 | | 0x2100.08 | | 0x2100.09 | |

- PDO4 Rx command

Master station -> slave station

| COB-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 0x500+NODEID | Setting 8 | | Setting 9 | | Setting 10 | | Setting 11 | |
| | 0x2100.0a | | 0x2100.0b | | 0x2100.0c | | 0x2100.0d | |

- Description

The application methods for PDO3 Rx and PDO4 Rx are the same as that for PDO2 Rx. For the relationship between the settings and PZD in PROFIBUS communication, see Table 3-19.

### 3.5.6 PDO3 Tx and PDO4 Tx

PDO3 Tx and PDO4 Tx are used by the inverter to transmit real-time process data to the master station, such as running frequency.

The default transmission type of PDO3 Tx and PDO4 Tx is 254, and therefore PDO3 Tx or PDO4 Tx is transmitted once data corresponding to a returned value in the same command changes.

- PDO3 Tx command

Slave station -> master station

| COB-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 0x380+NODEID | Returned value 4 | | Returned value 5 | | Returned value 6 | | Returned value 7 | |
| | 0x2000.06 | | 0x2000.07 | | 0x2000.08 | | 0x2000.09 | |

- PDO4 Tx command

Slave station -> master station

| COB-ID | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
|---|---|---|---|---|---|---|---|---|
| 0x480+NODEID | Returned value 8 | | Returned value 9 | | Returned value 10 | | Returned value 11 | |
| | 0x2000.0a | | 0x2000.0b | | 0x2000.0c | | 0x2000.0d | |

- Description

The application methods for PDO3 Tx and PDO4 Tx are the same as that for PDO2 Tx. For the relationship between the returned values and PZD in PROFIBUS communication, see

Table 3-20.

## 3.6 Monitoring process data through SDO commands

The communication can use SDOs as well as PDOs to to monitor the process data of an inverter. You can select a monitoring mode as required. You can monitor the inverter by using SDOs to read the manufacturer-defined object dictionary.

For the definition and application of the CWs, SWs, settings, and returned values in the manufacturer-defined object dictionary, see the PDO description section. For application of SDOs, see the SDO description section. Do not try to use SDOs to read and write inverter parameters.

Table 3-19 and Table 3-20 describe the manufacturer-defined object dictionary.

Table 3-19 Objects with the control function in the manufacturer-defined object dictionary

| Index (hexadecimal) | Subindex (hexadecimal) | Function | Access permission | Data length | Corresponding to |
|---|---|---|---|---|---|
| 2100 | 0 | Request code (do not use it) | RW | 2 bytes | / |
| | 1 | Parameter address (do not use it) | RW | 2 bytes | / |
| | 2 | Requested data (do not use it) | RW | 2 bytes | / |
| | 3 | Setting 1 | RW | 2 bytes | Received PZD2 |
| | 4 | Setting 2 | RW | 2 bytes | Received PZD3 |
| | 5 | Setting 3 | RW | 2 bytes | Received PZD4 |
| | 6 | Setting 4 | RW | 2 bytes | Received PZD5 |
| | 7 | Setting 5 | RW | 2 bytes | Received PZD6 |
| | 8 | Setting 6 | RW | 2 bytes | Received PZD7 |
| | 9 | Setting 7 | RW | 2 bytes | Received PZD8 |
| | A | Setting 8 | RW | 2 bytes | Received PZD9 |
| | B | Setting 9 | RW | 2 bytes | Received PZD10 |
| | C | Setting 10 | RW | 2 bytes | Received PZD11 |
| | D | Setting 11 | RW | 2 bytes | Received PZD12 |
| | E | Reserved | RW | 2 bytes | / |
| | F | Reserved | RW | 2 bytes | / |
| 2101 | 0 | CW | RW | 2 bytes | / |

Table 3-20 Objects with the monitoring function in the manufacturer-defined object dictionary

| Index (hexadecimal) | Subindex (hexadecimal) | Function | Access permission | Data length | Corresponding to |
|---|---|---|---|---|---|
| 2000 | 0 | Response code (do not use it) | RO | 2 bytes | / |
| | 1 | Error code (do not use it) | RO | 2 bytes | / |
| | 2 | Response data (do not use it) | RO | 2 bytes | / |
| | 3 | Returned value 1 | RO | 2 bytes | Transmitted PZD2 |
| | 4 | Returned value 2 | RO | 2 bytes | Transmitted PZD3 |
| | 5 | Returned value 3 | RO | 2 bytes | Transmitted PZD4 |
| | 6 | Returned value 4 | RO | 2 bytes | Transmitted PZD5 |
| | 7 | Returned value 5 | RO | 2 bytes | Transmitted PZD6 |
| | 8 | Returned value 6 | RO | 2 bytes | Transmitted PZD7 |
| | 9 | Returned value 7 | RO | 2 bytes | Transmitted PZD8 |
| | A | Returned value 8 | RO | 2 bytes | Transmitted PZD9 |
| | B | Returned value 9 | RO | 2 bytes | Transmitted PZD10 |
| | C | Returned value 10 | RO | 2 bytes | Transmitted PZD11 |
| | D | Returned value 11 | RO | 2 bytes | Transmitted PZD12 |
| | E | Reserved | RO | 2 bytes | / |
| | F | Reserved | RO | 2 bytes | / |
| 2001 | 0 | SW | RO | 2 bytes | / |

- Examples

Example 1: To instruct the inverter whose address is 3 to run forwardly, the master station transmits the following SDO command.

| COB-ID | Request code | Object index | | Subindex | Requested data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x603 | 0x2B | 0x01 | 0x21 | 0x00 | 0x01 | 0x00 | 0x00 | 0x00 |

Example 2: Assume that the address of the inverter slave station is 3, and the function of setting 1 is defined as "Set frequency". To set the frequency to 50.00 Hz (that is, setting 1=0x1388), the master station transmits the following SDO command.

| COB-ID | Request code | Object index | | Subindex | Requested data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x603 | 0x2B | 0x00 | 0x21 | 0x03 | 0x88 | 0x13 | 0x00 | 0x00 |

Example 3: To read the running state of the inverter whose address is 3, the master station transmits the following SDO command.

| COB-ID | Request code | Object index | | Subindex | Requested data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x603 | 0x40 | 0x01 | 0x20 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

If the inverter is running forward, the following SDO command is returned to the master station.

| COB-ID | Request code | Object index | | Subindex | Requested data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x583 | 0x4B | 0x01 | 0x20 | 0x00 | 0x01 | 0x01 | 0x00 | 0x00 |

Example 4: Assume that the address of the inverter slave station is 3, and the function of setting 1 is defined as "Set frequency". To set the frequency to 50.00 Hz (that is, setting 1=0x1388), the master station transmits the following SDO command.

| COB-ID | Request code | Object index | | Subindex | Requested data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x603 | 0x40 | 0x00 | 0x20 | 0x03 | 0x00 | 0x00 | 0x00 | 0x00 |

If the running frequency of the inverter is 50.00 Hz, the following SDO command is returned to the master station.

| COB-ID | Request code | Object index | | Subindex | Requested data | | | |
|---|---|---|---|---|---|---|---|---|
| | Byte0 | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 |
| 0x583 | 0x4B | 0x00 | 0x20 | 0x03 | 0x88 | 0x13 | 0x00 | 0x00 |

## 3.7 Baud rate and communication address setting

### 3.7.1 Baud rate setting

After setting the CANopen baud rate and communication address, you need to restart the inverter to enable the settings to take effect.

The CANopen baud rate is set through the corresponding inverter function parameter. For description of function code addresses, see the inverter operation manual. Table 3-21 describes the values of the function parameter and their corresponding baud rates.

Table 3-21 Baud rate setting

| Function parameter value | Baud rate (bit/s) |
|---|---|
| 0 | 1000 k |
| 1 | 800 k |
| 2 | 500 k |
| 3 | 250 k |
| 4 | 125 k |
| 5 | 100 k |
| 6 | 50 k |
| 7 | 20 k |

### 3.7.2 Communication address setting

The CANopen communication address is set through the function parameter P15.01.

### 3.7.3 Function codes related to transmitted and received PZD

Table 3-22 Received PZD

| Function code | Word | Value range | Default value |
|---|---|---|---|
| P15.02 | Received PZD2 | 0: Invalid<br>1: Set frequency (0–Fmax, unit: 0.01 Hz)<br>2: PID reference (0–1000, in which 1000 corresponds to 100.0%)<br>3: PID feedback (0–1000, in which 1000 corresponds to | 0 |
| P15.03 | Received PZD3 | | 0 |
| P15.04 | Received | | 0 |

<table>
<tr><th>Function code</th><th>Word</th><th>Value range</th><th>Default value</th></tr>
<tr><td></td><td>PZD4</td><td rowspan="9">100.0%)<br>4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01 Hz)<br>6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01 Hz)<br>7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>8: Upper limit of the brake torque (0–2000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>9: Virtual input terminal command, 0x000–0x3FF (corresponding to S8, S7, S6, S5, HDIB, HDIA, S4, S3, S2, and S1 in sequence)<br>10: Virtual output terminal command, 0x00–0x0F (corresponding to RO2, RO1, HDO, and Y1 in sequence)<br>11: Voltage setting (for V/F separation)<br>(0–1000, in which 1000 corresponds to 100.0% of the rated voltage of the motor)<br>12: AO output setting 1 (-1000–+1000, in which 1000 corresponds to 100.0%)<br>13: AO output setting 2 (-1000–+1000, in which 1000 corresponds to 100.0%)<br>14: MSB of position reference (signed number)<br>15: LSB of position reference (unsigned number)<br>16: MSB of position feedback (signed number)<br>17: LSB of position feedback (unsigned number)<br>18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0)</td><td></td></tr>
<tr><td>P15.05</td><td>Received PZD5</td><td>0</td></tr>
<tr><td>P15.06</td><td>Received PZD6</td><td>0</td></tr>
<tr><td>P15.07</td><td>Received PZD7</td><td>0</td></tr>
<tr><td>P15.08</td><td>Received PZD8</td><td>0</td></tr>
<tr><td>P15.09</td><td>Received PZD9</td><td>0</td></tr>
<tr><td>P15.10</td><td>Received PZD10</td><td>0</td></tr>
<tr><td>P15.11</td><td>Received PZD11</td><td>0</td></tr>
<tr><td>P15.12</td><td>Received PZD12</td><td>0</td></tr>
</table>

Table 3-23 Transmitted PZD

<table>
<tr><th>Function code</th><th>Word</th><th>Value range</th><th>Default value</th></tr>
<tr><td>P15.13</td><td>Transmitted PZD2</td><td rowspan="4">0: Invalid<br>1: Running frequency (×100, Hz)<br>2: Set frequency (×100, Hz)<br>3: Bus voltage (×10, V)</td><td>0</td></tr>
<tr><td>P15.14</td><td>Transmitted PZD3</td><td>0</td></tr>
<tr><td>P15.15</td><td>Transmitted PZD4</td><td>0</td></tr>
<tr><td>P15.16</td><td>Transmitted PZD5</td><td>0</td></tr>
</table>

| Function code | Word | Value range | Default value |
|---|---|---|---|
| P15.17 | Transmitted PZD6 | 4: Output voltage (×1, V)<br>5: Output current (×10, A)<br>6: Actual output torque (×10, %)<br>7: Actual output power (×10, %)<br>8: Rotating speed of the running (×1, RPM)<br>9: Linear speed of the running (×1, m/s)<br>10: Ramp frequency reference<br>11: Fault code<br>12: AI1 value (×100, V)<br>13: AI2 value (×100, V)<br>14: AI3 value (×100, V)<br>15: HDIA frequency (×100, kHz)<br>16: Terminal input state<br>17: Terminal output state<br>18: PID reference (×100, %)<br>19: PID feedback (×100, %)<br>20: Rated torque of the motor<br>21: MSB of position reference (signed number)<br>22: LSB of position reference (unsigned number)<br>23: MSB of position feedback (signed number)<br>24: LSB of position feedback (unsigned number)<br>25: State word<br>26: HDIB frequency value (×100, kHz) | 0 |
| P15.18 | Transmitted PZD7 | | 0 |
| P15.19 | Transmitted PZD8 | | 0 |
| P15.20 | Transmitted PZD9 | | 0 |
| P15.21 | Transmitted PZD10 | | 0 |
| P15.22 | Transmitted PZD11 | | 0 |
| P15.23 | Transmitted PZD12 | | 0 |

# 4 PROFINET communication card

## 4.1 Overview

1. Thanks for choosing INVT PROFINET communication cards. This manual describes the function specifications, installation, basic operation and settings, and information about the network protocol. To ensure that you install and operate the product properly, read this manual and the communication protocol section in the inverter operation manual carefully before you use the product.
2. This manual only describes how to operate the PROFINET communication card and the related commands but does not provide details about the PROFINET protocol. For more information about the PROFINET protocol, read the related specialized articles or books.
3. This communication card is defined as a PROFINET slave station communication card and is used on an inverter that supports PROFINET communication.
4. The communication card supports the linear network topology and star-shaped network topology.
5. The communication card supports 32 inputs/outputs to read and write process data, read state data, and read and write function parameters of an inverter.

## 4.2 Features

**1. Supported functions**

- Supports the PROFINET protocol, and supports PROFINET I/O devices
- Provides two PROFINET I/O ports and supports the 100 M full-duplex operation
- Supports the linear network topology and star-shaped network topology.

**2. Supported communication types**

- Standard Ethernet channels

  Standard Ethernet channels are non-realtime communication channels that use the TCP/IP protocol, and are mainly used for device parameterization and configuration and to read diagnosis data.

- Real-time (RT) communication channels

  RT channels are optimized channels for real-time communication. They take precedence over TCP (UDP)/IP, which ensures that various stations on a network perform data transmission with high time requirements at a certain interval. The bus period may reach the precision of millisecond. These channels are used to transmit process data, alarm data, etc.

➢ Isochronous real-time (IRT) communication channels

IRT channels are implemented through the built-in Switch-ASIC IRT chip. IRT communication can further shorten the processing time of the communication stack software, synchronizing data transmission of the program and device. The transmission delay is less than 1 ms, and the jitter is less than 1 μs. The typical application is motion control.

## 3. Communication ports

Standard RJ45 ports are used in PROFINET communication. The communication card provides two RJ45 ports with no transmission direction defined, and therefore you can insert a cable into the port without regarding to its direction. Figure 4-1 shows the ports, and Table 4-1 describes the functions of the ports.


Figure 4-1 Two standard RJ45 ports

Table 4-1 Functions of the standard RJ45 ports

| Pin | Name | Description |
|---|---|---|
| 1 | TX+ | Transmit Data+ |
| 2 | TX- | Transmit Data- |
| 3 | RX+ | Receive Data+ |
| 4 | n/c | Not connected |
| 5 | n/c | Not connected |
| 6 | RX- | Receive Data- |
| 7 | n/c | Not connected |
| 8 | n/c | Not connected |

## 4. State indicators

The PROFINET communication card provides nine LED indicators to indicate its states.

Table 4-2 describes the definitions of the indicators.

Table 4-2 Definitions of the state indicators

| LED | Color | State | Description |
|---|---|---|---|
| LED1 | Green | | 3.3 V power indicator |
| LED2<br>(Bus state indicator) | Red | On | Not connected through a network cable |
| | | Blinking | Connected to the PROFINET controller |

| LED | Color | State | Description |
|---|---|---|---|
| | | | through a network cable, but no communication established |
| | | Off | Communication established with the PROFINET controller |
| LED3 (System fault indicator) | Red | On | PROFINET diagnosis enabled |
| | | Off | PROFINET diagnosis disabled |
| LED4 (Slave ready indicator) | Green | On | TPS-1 communication stack started |
| | | Blinking | TPS-1 waits for the initialization of MCU |
| | | Off | TPS-1 communication stack not started |
| LED5 (Maintenance state indicator) | Green | | Defined by the manufacturer, depending on the characteristics of the device |
| LED6/7 (Network port state indicator) | Green | On | PROFINET communication card connected to the PC/PLC through a network cable |
| | | Off | PROFINET communication card not connected to the PC/PLC |
| LED8/9 (Network port communication indicator) | Green | On | PROFINET communication card communicating with the PC/PLC |
| | | Off | PROFINET communication card not communicating with the PC/PLC |

## 4.3 Electrical wiring

PROFINET communication card provides standard RJ45 ports and supports the linear network topology and star-shaped network topology. Figure 4-2 and Figure 4-3 shows the electrical wiring diagrams.

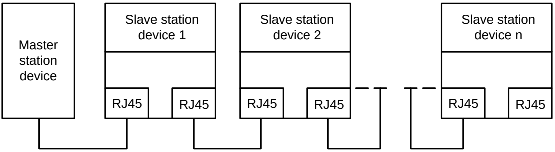



Figure 4-2 Electrical wiring diagram of the linear network topology

**Note:** For the star-shaped network topology, you need to use a PROFINET switch.

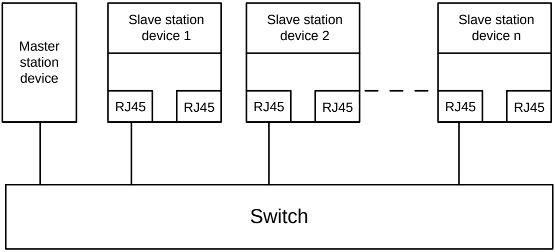



Figure 4-3 Electrical wiring diagram of the star-shaped network topology

## 4.4 Communication

### 4.4.1 Packet format

Table 4-3 describes the structure of an RT frame (non-synchronous).

Table 4-3 Structure of an RT frame

| Data header | Ethernet type | VLAN | Ethernet type | Frame identifier | RT user data | Period counter | Data state | Transmission state | FCS |
|---|---|---|---|---|---|---|---|---|---|
| | 2 bytes | 2 bytes | 2 bytes | 2 bytes | 36–1440 bytes | 2 bytes | 1 byte | 1 byte | 4 bytes |
| | 0x8100 | | 0x8892 | | | | | | |
| | VLAN flag | | | | | APDU state | | | |

| Data header | | | |
|---|---|---|---|
| 7-byte preamble | 1-byte synchronization information | 6-byte source MAC address | 6-byte destination MAC address |

Table 4-4 describes the structure of the IRT frame (synchronous).

Table 4-4 Structure of an IRT frame

| Data header | | | | Ethernet type | VLAN | Ethernet type | Frame identifier | IRT user data | FCS |
|---|---|---|---|---|---|---|---|---|---|
| 7-byte preamble | 1-byte synchronization | 6-byte source MAC address | 6-byte destination MAC address | 2 bytes | 2 bytes | 2 bytes | 2 bytes | 36–1440 bytes | 4 bytes |

### 4.4.2 PROFINET I/O communication

The PROFINET communication card supports 16-word input/output. Figure 4-4 shows the packet format for transmitting data with an inverter.

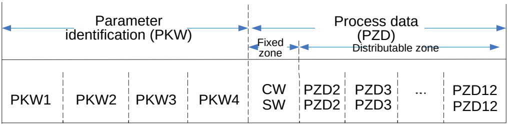



Figure 4-4 Packet structure

By using the 32 inputs/outputs, you can set the reference parameters of the inverter, monitor the state values, transmit control commands, monitor the running state, and read/write the function parameters of the inverter. For specific operations, see the following description.

Parameter zone:

PKW1—Parameter identification

PKW2—Array index number

PKW3—Paramter value 1

PKW4—Paramter value 2

Process data:

CW—Control word (transmitted from the master to a slave. For description, see Table 4-5)

SW—State word (transmitted from a slave to the master. For description, see Table 4-7.)

PZD—Process data (defined by users)

(When the process data is output by the master to a slave, it is a reference value; and when the process data is input by a slave to the master, it is an actual value.)

PZD zone (process data zone): The PZD zone in a communication packet is designed for controlling and monitoring an inverter. The master and slave stations always process the received PZD with the highest priority. The processing of PZD takes priority over that of PKW, and the master and slave stations always transmit the latest valid data on the interfaces.

CWs and SWs

Using CWs is the basic method of the fieldbus system to control inverters. A CW is transmitted by the fieldbus master station to an inverter device. In this case, the adapter module functions as a gateway. The inverter device responds to the bit code information of the CW and feeds state information back to the master through an SW.

Reference value: An inverter device may receive control information in multiple channels, including analog and digital input terminals, inverter control panel, and communication modules (such as RS485 and CH-PA01 adapter modules). To enable the control over inverter devices through PROFINET, you need to set the communication module as the controller of the inverter device.

Actual value: An actual value is a 16-bit word that includes information about inverter device operation. The monitoring function is defined through inverter parameters. The conversion scale of an integer transmitted as an actual value from the inverter device to the master depends on the set function. For more description, see the related inverter operation manual.

**Note:** An inverter device always checks the bytes of a CW and reference value.

**Task packet (master station -> inverter)**

CW: The first word in a PZD task packet is an inverter CW. The definitions of CWs on the PWM rectifier feedback side are different from those on the inverter side. Table 4-5 and Table 4-6 describe the definitions of CWs on the two sides.

Table 4-5 Goodrive350 series CWs

| Bit | Name | Value | Description |
|---|---|---|---|
| 0–7 | Communication-based control command | 1 | Forward running |
| | | 2 | Reverse running |
| | | 3 | Forward inching |
| | | 4 | Reverse inching |
| | | 5 | Stop |
| | | 6 | Coast to stop (emergency stop) |
| | | 7 | Fault reset |
| | | 8 | Stop inching |
| 8 | Enable writing | 1 | Enable writing (mainly through PKW1 to PKW4) |
| 9–10 | Motor group setting | 00 | Motor 1 |
| | | 01 | Motor 2 |
| 11 | Control mode switching | 1 | Enable torque/speed control switching |
| | | 0 | Disable switching |
| 12 | Reset power consumption to zero | 1 | Enable |
| | | 0 | Disable |
| 13 | Pre-excitation | 1 | Enable |
| | | 0 | Disable |
| 14 | DC braking | 1 | Enable |
| | | 0 | Disable |

| Bit | Name | Value | Description |
|---|---|---|---|
| 15 | Heartbeat reference | 1 | Enable |
| | | 0 | Disable |

Reference value (REF): The second to twelfth words in a PZD task packet are the main settings. The main frequency settings are provided by the main setting signal source. There is not main frequency setting on the PWM rectifier feedback side, and therefore the corresponding settings are reserved. Table 4-6 describes the settings on the Goodrive350 inverter side.

Table 4-6 Settings on the Goodrive350 series inverter side

| Function code | Word | Value range | Default value |
|---|---|---|---|
| P16.32 | Received PZD2 | 0: Invalid<br>1: Set frequency (0–Fmax, unit: 0.01 Hz)<br>2: PID reference (0–1000, in which 1000 corresponds to 100.0%)<br>3: PID feedback (0–1000, in which 1000 corresponds to 100.0%)<br>4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01 Hz)<br>6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01 Hz)<br>7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>8: Upper limit of the brake torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>9: Virtual input terminal command, 0x000–0x3FF (corresponding to S8, S7, S6, S5, HDIB, HDIA, S4, S3, S2, and S1 in sequence)<br>10: Virtual output terminal command, 0x00–0x0F (corresponding to RO2, RO1, HDO, and Y1 in sequence)<br>11: Voltage setting (for V/F separation)<br>(0–1000, in which 1000 corresponds to 100.0% of the rated voltage of the motor)<br>12: AO output setting 1 (-1000–+1000, in which 1000 corresponds to 100.0%)<br>13: AO output setting 2 (-1000–+1000, in which 1000 | 0 |
| P16.33 | Received PZD3 | | 0 |
| P16.34 | Received PZD4 | | 0 |
| P16.35 | Received PZD5 | | 0 |
| P16.36 | Received PZD6 | | 0 |
| P16.37 | Received PZD7 | | 0 |
| P16.38 | Received PZD8 | | 0 |
| P16.39 | Received PZD9 | | 0 |
| P16.40 | Received PZD10 | | 0 |
| P16.41 | Received PZD11 | | 0 |
| P16.42 | Received PZD12 | | 0 |

| Function code | Word | Value range | Default value |
|---|---|---|---|
| | | corresponds to 100.0%)<br>14: MSB of position reference (signed number)<br>15: LSB of position reference (unsigned number)<br>16: MSB of position feedback (signed number)<br>17: LSB of position feedback (unsigned number)<br>18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0) | |

**Response packet (inverter -> master station)**

SW: The first word in a PZD response packet is an inverter SW. Table 4-7 describes the definitions of the inverter SWs.

Table 4-7 Goodrive350 series SWs

| Bit | Name | Value | Description |
|---|---|---|---|
| 0–7 | Running state | 1 | In forward running |
| | | 2 | In reverse running |
| | | 3 | Stopped |
| | | 4 | Faulty |
| | | 5 | POFF |
| 8 | Bus voltage established | 1 | Ready to run |
| | | 0 | Not ready to run |
| 9–10 | Motor group feedback | 0 | Motor 1 |
| | | 1 | Motor 2 |
| 11 | Motor type feedback | 1 | Synchronous motor |
| | | 0 | Asynchronous motor |
| 12 | Overload pre-alarm feedback | 1 | Overload pre-alarm generated |
| | | 0 | No overload pre-alarm generated |
| 13–14 | Run/Stop mode | 0 | Keypad-based control |
| | | 1 | Terminal-based control |
| | | 2 | Communication-based control |
| | | 3 | Reserved |
| 15 | Heartbeat feedback | 1 | Heartbeat feedback |
| | | 0 | No heartbeat feedback |

Actual value (ACT): The second to twelfth words in a PZD task packet are the main actual values. The main actual frequency values are provided by the main actual value signal source.

Table 4-8 Actual state values of the Goodrive350 series

<table>
<tr><th>Function code</th><th>Word</th><th>Value range</th><th>Default value</th></tr>
<tr><td>P16.43</td><td>Transmitted PZD2</td><td rowspan="11">0: Invalid<br>1: Running frequency (×100, Hz)<br>2: Set frequency (×100, Hz)<br>3: Bus voltage (×10, V)<br>4: Output voltage (×1, V)<br>5: Output current (×10, A)<br>6: Actual output torque (×10, %)<br>7: Actual output power (×10, %)<br>8: Rotating speed of the running (×1, RPM)<br>9: Linear speed of the running (×1, m/s)<br>10: Ramp frequency reference<br>11: Fault code<br>12: AI1 value (×100, V)<br>13: AI2 value (×100, V)<br>14: AI3 value (×100, V)<br>15: HDIA frequency (×100, kHz)<br>16: Terminal input state<br>17: Terminal output state<br>18: PID reference (×100, %)<br>19: PID feedback (×100, %)<br>20: Rated torque of the motor<br>21: MSB of position reference (signed number)<br>22: LSB of position reference (unsigned number)<br>23: MSB of position feedback (signed number)<br>24: LSB of position feedback (unsigned number)<br>25: State word<br>26: HDIB frequency value (×100, kHz)</td><td>0</td></tr>
<tr><td>P16.44</td><td>Transmitted PZD3</td><td>0</td></tr>
<tr><td>P16.45</td><td>Transmitted PZD4</td><td>0</td></tr>
<tr><td>P16.46</td><td>Transmitted PZD5</td><td>0</td></tr>
<tr><td>P16.47</td><td>Transmitted PZD6</td><td>0</td></tr>
<tr><td>P16.48</td><td>Transmitted PZD7</td><td>0</td></tr>
<tr><td>P16.49</td><td>Transmitted PZD8</td><td>0</td></tr>
<tr><td>P16.50</td><td>Transmitted PZD9</td><td>0</td></tr>
<tr><td>P16.51</td><td>Transmitted PZD10</td><td>0</td></tr>
<tr><td>P16.52</td><td>Transmitted PZD11</td><td>0</td></tr>
<tr><td>P16.53</td><td>Transmitted PZD12</td><td>0</td></tr>
</table>

**PKW zone**

PKW zone (parameter identification flag PKW1—numerical zone): The PKW zone describes the processing mode of the parameter identification interface. A PKW interface is not a physical interface but a mechanism that defines the transmission mode (such reading and writing a parameter value) of a parameter between two communication ends.

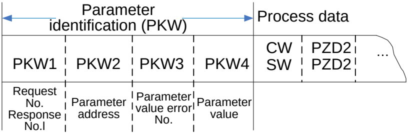



Figure 4-5 Parameter identification zone

In the periodic communication, the PKW zone consists of four 16-bit words. The following table describes the definition of each word.

| First word PKW1 (16 bits) | | |
|---|---|---|
| Bits 15–00 | Task or response identification flag | 0–7 |
| Second word PKW2 (16 bits) | | |
| Bits 15–00 | Basic parameter address | 0–247 |
| Third word PKW3 (16 bits) | | |
| Bits 15–00 | Value (most significant word) of a parameter or error code of the returned value | 00 |
| Fourth word PKW4 (16 bits) | | |
| Bits 15–00 | Value (least significant word) of a parameter | 0–65535 |

**Note:** If the master station requests the value of a parameter, the values in PKW3 and PKW4 of the packet that the master station transmits to the inverter are no longer valid.

Task request and response: When transmitting data to a slave, the master uses a request number, and the slave uses a response number to accept or reject the request.

Table 4-9 Definition of the task identification flag PKW1

| **Request No. (from the master to a slave)** | | **Response signal** | |
|---|---|---|---|
| **Request No.** | **Function** | **Acceptance** | **Rejection** |
| 0 | No task | 0 | — |
| 1 | Requesting the value of a parameter | 1, 2 | 3 |
| 2 | Modifying a parameter value (one word) [modifying the value only on RAM] | 1 | 3 or 4 |
| 3 | Modifying a parameter value (two words) [modifying the value only on RAM] | 2 | 3 or 4 |
| 4 | Modifying a parameter value (one word) [modifying the value on both RAM and EEPROM] | 1 | 3 or 4 |
| 5 | Modifying a parameter value (two words) [modifying the value on both RAM and EEPROM] | 2 | 3 or 4 |

**Note:** The requests #2, #3, and #5 are not supported currently.

Table 4-10 Definition of the response identification flag PKW1

| Response No. (from a slave to the master) | |
|---|---|
| **Response No.** | **Function** |
| 0 | No response |
| 1 | Transmitting the value of a parameter (one word) |
| 2 | Transmitting the value of a parameter (two words) |
| 3 | The task cannot be executed and one of the following error number is returned:<br>1: Invalid command<br>2: Invalid data address<br>3: Invalid data value<br>4: Operation failure<br>5: Password error<br>6: Data frame error<br>7: Parameter read only<br>8: Parameter cannot be modified during inverter running<br>9: Password protection |

# Appendix A CANopen object dictionary

| Index (hexadecimal) | Subindex | Description | Access permission | Data type | Default value |
|---|---|---|---|---|---|
| 1000 | 0 | Device type | RO | Unsigned32 | 0x0000 0000 |
| 1001 | 0 | Error register | RO | Unsigned8 | / |
| 1003 | **Error code register** | | | | |
| | 0 | Number of subindexes | RW | / | / |
| | 1 | Error code | RO | Unsigned32 | / |
| 1005 | 0 | COB-ID SYNC | RW | Unsigned32 | / |
| 1006 | 0 | Communication cycle period | RW | Unsigned32 | / |
| 1007 | 0 | Length of synchronous window | RW | Unsigned32 | / |
| 1008 | 0 | Manufacturer-defined device name | CONST | String | INVT CANopen |
| 1009 | 0 | Manufacturer-defined hardware version | CONST | String | V1.00 |
| 100A | 0 | Manufacturer-defined software version | CONST | String | V1.00 |
| 100C | 0 | Protection time | RW | Unsigned16 | 0 |
| 100D | 0 | Life cycle factor | RW | Unsigned16 | 0 |
| 1016 | **Consumer heartbeat time** | | | | |
| | 0 | Number of subindexes | RO | Unsigned8 | / |
| | 1 | Consumer heartbeat time | RW | Unsigned32 | / |
| 1017 | 0 | Producer heartbeat time | RW | Unsigned16 | 0 |
| 1018 | **Identifier objects** | | | | |
| | 0 | Number of subindexes | RO | Unsigned8 | 4 |
| | 1 | Supplier ID | RO | Unsigned32 | 0x0000 0000 |
| | 2 | Product code | RO | Unsigned32 | 0x0000 0000 |
| | 3 | Revision No. | RO | Unsigned32 | 0x0000 0000 |

| Index (hexadecimal) | Subindex | Description | Access permission | Data type | Default value |
|---|---|---|---|---|---|
| | 4 | Sequence No. | RO | Unsigned32 | 0x0000 0000 |
| 1200 | **Servo SDO** | | | | |
| | 0 | Number of subindexes | RO | Unsigned8 | / |
| | 1 | COB-ID Client -> server (Rx) | RO | Unsigned32 | 600H+Node ID |
| | 2 | COB-ID Server -> client (Tx) | RO | Unsigned32 | 580H+Node ID |
| 1280 | **SDO** | | | | |
| | 0 | Number of subindexes | RO | Unsigned8 | / |
| | 1 | COB-ID Client -> server (Rx) | RO | Unsigned32 | / |
| | 2 | COB-ID Server -> client (Tx) | RO | Unsigned32 | / |
| | 3 | Node ID of server SDO | RO | Unsigned8 | / |
| 1400 | **PDO1 Rx communication parameters** | | | | |
| | 0 | Supported Max. number of subindexes | RO | Unsigned8 | / |
| | 1 | COB-ID used by PDO | RW | Unsigned32 | / |
| | 2 | Transmission type | RW | Unsigned8 | / |
| | 3 | / | / | Unsigned16 | / |
| | 4 | / | / | Unsigned8 | / |
| | 5 | Event timer | RW | Unsigned16 | / |
| 1401 | **PDO2 Rx communication parameters** | | | | |
| | 0 | Supported Max. number of subindexes | RO | Unsigned8 | / |
| | 1 | COB-ID used by PDO | RW | Unsigned32 | / |
| | 2 | Transmission type | RW | Unsigned8 | / |
| | 3 | / | / | Unsigned16 | / |
| | 4 | / | / | Unsigned8 | / |
| | 5 | Event timer | RW | Unsigned16 | / |

| Index (hexadecimal) | Subindex | Description | Access permission | Data type | Default value |
|---|---|---|---|---|---|
| 1402 | PDO3 Rx communication parameters | | | | |
| | 0 | Supported Max. number of subindexes | RO | Unsigned8 | / |
| | 1 | COB-ID used by PDO | RW | Unsigned32 | / |
| | 2 | Transmission type | RW | Unsigned8 | / |
| | 3 | / | / | Unsigned16 | / |
| | 4 | / | / | Unsigned8 | / |
| | 5 | Event timer | RW | Unsigned16 | / |
| 1403 | PDO4 Rx communication parameters | | | | |
| | 0 | Supported Max. number of subindexes | RO | Unsigned8 | / |
| | 1 | COB-ID used by PDO | RW | Unsigned32 | / |
| | 2 | Transmission type | RW | Unsigned8 | / |
| | 3 | / | / | Unsigned16 | / |
| | 4 | / | / | Unsigned8 | / |
| | 5 | Event timer | RW | Unsigned16 | / |
| 1600 | PDO1 Rx mapping parameters | | | | |
| | 0 | Number of application program objects mapped in PDO | RW | Unsigned8 | 3 |
| | 1 | First mapped object | RW | Unsigned32 | 0x21000010 |
| | 2 | Second mapped object | RW | Unsigned32 | 0x21000110 |
| | 3 | Third mapped object | RW | Unsigned32 | 0x21000210 |
| 1601 | PDO2 Rx mapping parameters | | | | |
| | 0 | Number of application program objects mapped in PDO | RW | Unsigned8 | 4 |
| | 1 | First mapped | RW | Unsigned32 | 0x21010010 |

| Index (hexadecimal) | Subindex | Description | Access permission | Data type | Default value |
|---|---|---|---|---|---|
| | | object | | | |
| | 2 | Second mapped object | RW | Unsigned32 | 0x21000310 |
| | 3 | Third mapped object | RW | Unsigned32 | 0x21000410 |
| | 4 | Fourth mapped object | RW | Unsigned32 | 0x21000510 |
| 1602 | **PDO3 Rx mapping parameters** | | | | |
| | 0 | Number of application program objects mapped in PDO | RW | Unsigned8 | 4 |
| | 1 | First mapped object | RW | Unsigned32 | 0x21000610 |
| | 2 | Second mapped object | RW | Unsigned32 | 0x21000710 |
| | 3 | Third mapped object | RW | Unsigned32 | 0x21000810 |
| | 4 | Fourth mapped object | RW | Unsigned32 | 0x21000910 |
| 1603 | **PDO4 Rx mapping parameters** | | | | |
| | 0 | Number of application program objects mapped in PDO | RW | Unsigned8 | 4 |
| | 1 | First mapped object | RW | Unsigned32 | 0x21000a10 |
| | 2 | Second mapped object | RW | Unsigned32 | 0x21000b10 |
| | 3 | Third mapped object | RW | Unsigned32 | 0x21000c10 |
| | 4 | Fourth mapped object | RW | Unsigned32 | 0x21000d10 |
| 1800 | **PDO1 Tx communication parameters** | | | | |
| | 0 | Supported Max. number of subindexes | RO | Unsigned8 | / |

| Index (hexadecimal) | Subindex | Description | Access permission | Data type | Default value |
|---|---|---|---|---|---|
| | 1 | COB-ID used by PDO | RW | Unsigned32 | / |
| | 2 | Transmission type | RW | Unsigned8 | 255 |
| | 3 | Disabled time | RW | Unsigned16 | 500 |
| | 4 | Reserved | RW | Unsigned8 | / |
| | 5 | Event timer | RW | Unsigned16 | 0 |
| 1801 | **PDO2 Tx communication parameters** | | | | |
| | 0 | Supported Max. number of subindexes | RO | Unsigned8 | / |
| | 1 | COB-ID used by PDO | RW | Unsigned32 | / |
| | 2 | Transmission type | RW | Unsigned8 | 254 |
| | 3 | Disabled time | RW | Unsigned16 | 500 |
| | 4 | Reserved | RW | Unsigned8 | / |
| | 5 | Event timer | RW | Unsigned16 | 0 |
| 1802 | **PDO3 Tx communication parameters** | | | | |
| | 0 | Supported Max. number of subindexes | RO | Unsigned8 | / |
| | 1 | COB-ID used by PDO | RW | Unsigned32 | / |
| | 2 | Transmission type | RW | Unsigned8 | 254 |
| | 3 | Disabled time | RW | Unsigned16 | 500 |
| | 4 | Reserved | RW | Unsigned8 | / |
| | 5 | Event timer | RW | Unsigned16 | 0 |
| 1803 | **PDO4 Tx communication parameters** | | | | |
| | 0 | Supported Max. number of subindexes | RO | Unsigned8 | / |
| | 1 | COB-ID used by PDO | RW | Unsigned32 | / |
| | 2 | Transmission type | RW | Unsigned8 | 254 |
| | 3 | Disabled time | RW | Unsigned16 | 500 |
| | 4 | Reserved | RW | Unsigned8 | / |
| | 5 | Event timer | RW | Unsigned16 | 0 |
| 1A00 | **PDO1 Tx mapping parameters** | | | | |

| Index (hexadecimal) | Subindex | Description | Access permission | Data type | Default value |
|---|---|---|---|---|---|
| | 0 | Number of application program objects mapped in PDO | RW | Unsigned8 | 3 |
| | 1 | First mapped object | RW | Unsigned32 | 0x20000010 |
| | 2 | Second mapped object | RW | Unsigned32 | 0x20000110 |
| | 3 | Third mapped object | RW | Unsigned32 | 0x20000210 |
| 1A01 | **PDO2 Tx mapping parameters** | | | | |
| | 0 | Number of application program objects mapped in PDO | RW | Unsigned8 | 4 |
| | 1 | First mapped object | RW | Unsigned32 | 0x20010010 |
| | 2 | Second mapped object | RW | Unsigned32 | 0x20000310 |
| | 3 | Third mapped object | RW | Unsigned32 | 0x20000410 |
| | 4 | Fourth mapped object | RW | Unsigned32 | 0x20000510 |
| 1A02 | **PDO3 Tx mapping parameters** | | | | |
| | 0 | Number of application program objects mapped in PDO | RW | Unsigned8 | 4 |
| | 1 | First mapped object | RW | Unsigned32 | 0x20000610 |
| | 2 | Second mapped object | RW | Unsigned32 | 0x20000710 |
| | 3 | Third mapped object | RW | Unsigned32 | 0x20000810 |
| | 4 | Fourth mapped object | RW | Unsigned32 | 0x20000910 |
| 1A03 | **PDO4 Tx mapping parameters** | | | | |

| Index (hexadecimal) | Subindex | Description | Access permission | Data type | Default value |
|---|---|---|---|---|---|
| | 0 | Number of application program objects mapped in PDO | RW | Unsigned8 | 4 |
| | 1 | First mapped object | RW | Unsigned32 | 0x20000a10 |
| | 2 | Second mapped object | RW | Unsigned32 | 0x20000b10 |
| | 3 | Third mapped object | RW | Unsigned32 | 0x20000c10 |
| | 4 | Fourth mapped object | RW | Unsigned32 | 0x20000d10 |

# Appendix B Related function codes

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| P00.01 | Channel of running commands | 0: Keypad<br>1: Terminal<br>2: Communication | 0–2 | 0 |
| P00.02 | Communication channel of running commands | 0: Modbus communication<br>1: PROFIBUS/CANopen/ DeviceNet communication<br>2: Ethernet communication<br>3: EtherCAT/PROFINET communication<br>4: PLC programmable extension card<br>5: Wireless communication card<br>**Note:** Channels 1, 2, 3, 4, and 5 are extension functions that require corresponding extension cards. | 0–5 | 0 |
| P00.06 | Frequency A command setting mode | 0: Keypad<br>1–8: Reserved<br>9: PROFIBUS/CANopen/DeviceNet communication<br>10: Ethernet communication<br>11–12: Reserved<br>13: EtherCAT/PROFINET communication<br>14–15: Reserved | 0–15 | 0 |
| P00.07 | Frequency B command setting mode | | 0–15 | 2 |
| P03.11 | Torque setting mode | 0–1: Keypad<br>2–7: Reserved<br>8: PROFIBUS/CANopen/DeviceNet communication<br>9: Ethernet communication<br>10: Reserved<br>11: EtherCAT/PROFINET communication<br>12: Reserved | 0–12 | 0 |
| P03.14 | Setting mode of upper frequency limit of forward running in torque | 0: Keypad (P03.16)<br>1–6: Reserved<br>7: PROFIBUS/CANopen/DeviceNet communication | 0–12 | 0 |

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| | control | 8: Ethernet communication<br>9: Reserved<br>10: EtherCAT/PROFINET communication<br>11–12: Reserved | | |
| P03.15 | Setting mode of upper frequency limit of reverse running in torque control | 0: Keypad (P03.17)<br>1–6: Reserved<br>7: PROFIBUS/CANopen/DeviceNet communication<br>8: Ethernet communication<br>9: Reserved<br>10: EtherCAT/PROFINET communication<br>11–12: Reserved | 0–12 | 0 |
| P03.18 | Setting mode of upper limit of electromotive torque | 0: Keypad (P03.20)<br>1–5: Reserved<br>6: PROFIBUS/CANopen/DeviceNet communication<br>7: Ethernet communication<br>8: Reserved<br>9: EtherCAT/PROFINET communication<br>10–11: Reserved | 0–11 | 0 |
| P03.19 | Setting mode of upper limit of brake torque | 0: Keypad (P03.21)<br>1–5: Reserved<br>6: PROFIBUS/CANopen/DeviceNet communication<br>7: Ethernet communication<br>8: Reserved<br>9: EtherCAT/PROFINET communication<br>10–11: Reserved | 0–11 | 0 |
| P04.27 | Voltage setting channel | 0: Keypad (P04.28)<br>1–7: Reserved<br>8: PROFIBUS/CANopen/DeviceNet communication<br>9: Ethernet communication<br>10: Reserved | 0–13 | 0 |

<table>
<tr><th>Function code</th><th>Name</th><th>Parameter description</th><th>Setting range</th><th>Default value</th></tr>
<tr><td></td><td></td><td>11: EtherCAT/PROFINET communication<br>12–13: Reserved</td><td></td><td></td></tr>
<tr><td>P06.01</td><td>Y1 output</td><td rowspan="4">0: Invalid<br>1–23: Reserved<br>24: PROFIBUS/CANopen/DeviceNet communication virtual terminal output<br>25: Ethernet communication virtual terminal output<br>26–33: Reserved<br>34: EtherCAT/PROFINET communication virtual terminal output<br>35–63: Reserved</td><td>0–63</td><td>0</td></tr>
<tr><td>P06.02</td><td>HDO output</td><td>0–63</td><td>0</td></tr>
<tr><td>P06.03</td><td>Relay output RO1</td><td>0–63</td><td>1</td></tr>
<tr><td>P06.04</td><td>Relay output RO2</td><td>0–63</td><td>5</td></tr>
<tr><td>P06.14</td><td>Analog output AO1</td><td rowspan="2">0: Running frequency<br>1–15: Reserved<br>16: PROFIBUS/CANopen/DeviceNet communication setting 1<br>17: PROFIBUS/CANopen/DeviceNet communication setting 2<br>18: Ethernet communication setting 1<br>19: Ethernet communication setting 2<br>20: Reserved<br>21: EtherCAT/PROFINET communication setting 1<br>22–26: Reserved<br>27: EtherCAT/PROFINET communication setting 2<br>28–47: Reserved</td><td>0–47</td><td>0</td></tr>
<tr><td>P06.16</td><td>HDO high-speed pulse output</td><td>0–47</td><td>0</td></tr>
<tr><td>P07.27</td><td>Type of current fault</td><td>0: No fault<br>29: PROFIBUS communication fault (E-DP)<br>30: Ethernet communication fault (E-NET)<br>31: CANopen communication fault (E-CAN)<br>57: PROFINET communication timeout fault (E-PN)</td><td>/</td><td>/</td></tr>
</table>

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| | | 58: CAN communication timeout fault (ESCAN)<br>60: Card identification failure in slot 1 (F1-Er)<br>61: Card identification failure in slot 2 (F2-Er)<br>62: Card identification failure in slot 3 (F3-Er)<br>63: Card communication failure in slot 1 (C1-Er)<br>64: Card communication failure in slot 2 (C2-Er)<br>65: Card communication failure in slot 3 (C3-Er)<br>66: EtherCAT communication fault (E-CAT)<br>67: BACnet communication fault (E-BAC)<br>68: DeviceNet communication fault (E-DEV)<br>69: CAN slave fault in master/slave synchronous communication (S-Err) | | |
| P07.28 | Type of last fault | / | / | / |
| P07.29 | Type of last but one fault | / | / | / |
| P07.30 | Type of last but two fault | / | / | / |
| P07.31 | Type of last but three fault | / | / | / |
| P07.32 | Type of last but four fault | / | / | / |
| P08.31 | Motor 1 and motor 2 switching channel | 0x00–0x14<br>LED ones place: Switching channel<br>0: Terminal<br>1: Modbus communication<br>2: PROFIBUS/CANopen/DeviceNet communication | 00–14 | 0x00 |

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| | | 3: Ethernet communication<br>4: EtherCAT/PROFINET communication<br>LED tens place: Switching in running<br>0: Disabled<br>1: Enabled | | |
| P09.00 | PID reference source | 0: Keypad (P09.01)<br>1–6: Reserved<br>7: PROFIBUS/CANopen/DeviceNet communication<br>8: Ethernet communication<br>9: Reserved<br>10: EtherCAT/PROFINET communication<br>11–12: Reserved | 0–12 | 0 |
| P09.02 | PID feedback source | 0: AI1<br>1–4: Reserved<br>5: PROFIBUS/CANopen/DeviceNet communication<br>6: Ethernet communication<br>7: Reserved<br>8: EtherCAT/PROFINET communication<br>9–10: Reserved | 0–10 | 0 |
| P15.01 | Module address | 0–127 | 0–127 | 2 |
| P15.02 | Received PZD2 | 0: Invalid<br>1: Set frequency (0–Fmax, unit: 0.01 Hz)<br>2: PID reference (0–1000, in which 1000 corresponds to 100.0%)<br>3: PID feedback (0–1000, in which 1000 corresponds to 100.0%)<br>4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01 Hz) | 0–31 | 0 |
| P15.03 | Received PZD3 | | 0–31 | 0 |
| P15.04 | Received PZD4 | | 0–31 | 0 |
| P15.05 | Received PZD5 | | 0–31 | 0 |
| P15.06 | Received PZD6 | | 0–31 | 0 |
| P15.07 | Received PZD7 | | 0–31 | 0 |
| P15.08 | Received PZD8 | | 0–31 | 0 |
| P15.09 | Received PZD9 | | 0–31 | 0 |
| P15.10 | Received PZD10 | | 0–31 | 0 |
| P15.11 | Received PZD11 | | 0–31 | 0 |
| P15.12 | Received PZD12 | | 0–31 | 0 |

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| | | 6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01 Hz)<br>7: Upper limit of the electromotive torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>8: Upper limit of the brake torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>9: Virtual input terminal command, 0x000–0x3FF<br>(corresponding to S8, S7, S6, S5, HDIB, HDIA, S4, S3, S2, and S1 in sequence)<br>10: Virtual output terminal command, 0x00–0x0F<br>(corresponding to RO2, RO1, HDO, and Y1 in sequence)<br>11: Voltage setting (for V/F separation) (0–1000, in which 1000 corresponds to 100.0% of the rated voltage of the motor)<br>12: AO output setting 1 (-1000–+1000, in which 1000 corresponds to 100.0%)<br>13: AO output setting 2 (-1000–+1000, in which 1000 corresponds to 100.0%)<br>14: MSB of position reference (signed number)<br>15: LSB of position reference (unsigned number)<br>16: MSB of position feedback (signed number)<br>17: LSB of position feedback (unsigned number)<br>18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0) | | |

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| | | 19–31: Reserved | | |
| P15.13 | Transmitted PZD2 | 0: Invalid<br>1: Running frequency (×100, Hz)<br>2: Set frequency (×100, Hz)<br>3: Bus voltage (×10, V)<br>4: Output voltage (×1, V)<br>5: Output current (×10, A)<br>6: Actual output torque (×10, %)<br>7: Actual output power (×10, %)<br>8: Rotating speed of the running (×1, RPM)<br>9: Linear speed of the running (×1, m/s)<br>10: Ramp frequency reference<br>11: Fault code<br>12: AI1 value (×100, V)<br>13: AI2 value (×100, V)<br>14: AI3 value (×100, V)<br>15: HDIA frequency (×100, kHz)<br>16: Terminal input state<br>17: Terminal output state<br>18: PID reference (×100, %)<br>19: PID feedback (×100, %)<br>20: Rated torque of the motor<br>21: MSB of position reference (signed number)<br>22: LSB of position reference (unsigned number)<br>23: MSB of position feedback (signed number)<br>24: LSB of position feedbackl (unsigned number)<br>25: State word<br>26: HDIB frequency value (×100, kHz)<br>27–31: Reserved | 0–31 | 0 |
| P15.14 | Transmitted PZD3 | | 0–31 | 0 |
| P15.15 | Transmitted PZD4 | | 0–31 | 0 |
| P15.16 | Transmitted PZD5 | | 0–31 | 0 |
| P15.17 | Transmitted PZD6 | | 0–31 | 0 |
| P15.18 | Transmitted PZD7 | | 0–31 | 0 |
| P15.19 | Transmitted PZD8 | | 0–31 | 0 |
| P15.20 | Transmitted PZD9 | | 0–31 | 0 |
| P15.21 | Transmitted PZD10 | | 0–31 | 0 |
| P15.22 | Transmitted PZD11 | | 0–31 | 0 |
| P15.23 | Transmitted PZD12 | | 0–31 | 0 |
| P15.24 | Temporary variable 1 used for transmitted PZD | 0–65535 | 0–65535 | 0 |

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| P15.25 | DP communication timeout time | 0.0 (invalid)–300.0s | 0.0–300.0 | 0.0s |
| P15.26 | CANopen communication timeout time | 0.0 (invalid)–300.0s | 0.0–300.0 | 0.0s |
| P15.27 | CANopen communication baud rate | 0: 1000 kbps<br>1: 800 kbps<br>2: 500 kbps<br>3: 250 kbps<br>4: 125 kbps<br>5: 100 kbps<br>6: 50 kbps<br>7: 20 kbps | 0–7 | 0 |
| P15.28 | CAN communication address | 0–127 | 0–127 | 1 |
| P15.29 | CAN baud rate setting | 0: 50 kbps<br>1: 125 kbps<br>2: 250 kbps<br>3: 500 kbps<br>4: 1 Mbps | 0–4 | 1 |
| P15.30 | CAN communication timeout time | 0.0 (invalid)–300.0s | 0.0–300.0 | 0.0s |
| P15.31 | DeviceNet communication timeout time | 0.0 (invalid)–300.0s | 0.0–300.0 | 0.0s |
| P15.32 | Displayed node baud rate | 0 | 0 | 0 |
| P15.33 | Enable polling | 0–1 | 0–1 | 1 |
| P15.34 | Output instance in polling | 19: INVT inverter output<br>20: ODVA basic speed control output<br>21: ODVA extended speed control output<br>22: ODVA speed and torque control output<br>23: ODVA extended speed and torque | 19–27 | 19 |

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| | | control output<br>24: INVT basic speed control output<br>25: INVT extended speed control output<br>26: INVT speed and torque control output<br>27: INVT extended speed and torque control output | | |
| P15.35 | Input instance in polling | 69: INVT inverter input<br>70: ODVA basic speed control input<br>71: ODVA extended speed control input<br>72: ODVA speed and torque control input<br>73: ODVA extended speed and torque control input<br>74: INVT basic speed control input<br>75: INVT extended speed control input<br>76: INVT speed and torque control input<br>77: INVT extended speed and torque control input | 69–77 | 69 |
| P15.36 | Enable state change/period | 0–1 | 0–1 | 0 |
| P15.37 | Output instance in state change/period | 19: INVT inverter output<br>20: ODVA basic speed control output<br>21: ODVA extended speed control output<br>22: ODVA speed and torque control output<br>23: ODVA extended speed and torque control output<br>24: INVT basic speed control output<br>25: INVT extended speed control output<br>26: INVT speed and torque control output<br>27: INVT extended speed and torque control output | 19–27 | 19 |
| P15.38 | Input instance in state change/period | 69: INVT inverter input<br>70: ODVA basic speed control input<br>71: ODVA extended speed control input | 69–77 | 69 |

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| | | 72: ODVA speed and torque control input<br>73: ODVA extended speed and torque control input<br>74: INVT basic speed control input<br>75: INVT extended speed control input<br>76: INVT speed and torque control input<br>77: INVT extended speed and torque control input | | |
| P15.39 | Output length of component 19 | 8–32 | 8–32 | 32 |
| P15.40 | Input length of component 19 | 8–32 | 8–32 | 32 |
| P15.41 | BACnet communication mode setting | 0: Enable P16.22 (I_Am service)<br>1: Enable P15.42 (Baud rate of BACnet_MSTP) | 0–1 | 0 |
| P15.42 | Baud rate of BACnet_MSTP | 0–5 | 0–5 | 0 |
| P15.43–P15.69 | Reserved | | | |
| P16.01 | Ethernet communication rate setting | 0: Self-adaption<br>1: 100M full duplex<br>2: 100M half duplex<br>3: 10M full duplex<br>4: 10M half duplex | 0–4 | 0 |
| P16.02 | IP address 1 | 0–255 | 0–255 | 192 |
| P16.03 | IP address 2 | 0–255 | 0–255 | 168 |
| P16.04 | IP address 3 | 0–255 | 0–255 | 0 |
| P16.05 | IP address 4 | 0–255 | 0–255 | 1 |
| P16.06 | Subnet mask 1 | 0–255 | 0–255 | 255 |
| P16.07 | Subnet mask 2 | 0–255 | 0–255 | 255 |
| P16.08 | Subnet mask 3 | 0–255 | 0–255 | 255 |
| P16.09 | Subnet mask 4 | 0–255 | 0–255 | 0 |
| P16.10 | Gateway 1 | 0–255 | 0–255 | 192 |
| P16.11 | Gateway 2 | 0–255 | 0–255 | 168 |
| P16.12 | Gateway 3 | 0–255 | 0–255 | 1 |
| P16.13 | Gateway 4 | 0–255 | 0–255 | 1 |
| P16.14 | Ethernet | 0x0000–0xFFFF | 0000–FF | 0x0000 |

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| | monitoring variable address 1 | | FF | |
| P16.15 | Ethernet monitoring variable address 2 | 0x0000–0xFFFF | 0000–FFFF | 0x0000 |
| P16.16 | Ethernet monitoring variable address 3 | 0x0000–0xFFFF | 0000–FFFF | 0x0000 |
| P16.17 | Ethernet monitoring variable address 4 | 0x0000–0xFFFF | 0000–FFFF | 0x0000 |
| P16.18 | Reserved | | | |
| P16.19 | EtherCAT synchronization period | 0–4 (0: 250 μs; 1: 500 μs; 2: 1 ms; 3: 2 ms) | 0–4 | 2 |
| P16.20 | MSD of BACnet device number | Independent code of BACnet device (0–4194303) | 0–4194 | 0 |
| P16.21 | LSD of BACnet device number | | 0–999 | 1 |
| P16.22 | BACnet "I-Am" service setting | 0: Transmission at power-on<br>1: Continuous transmission | 0–1 | 0 |
| P16.23 | BACnet communication timeout time | 0.0 (invalid)–300.0s | 0.0–300.0 | 0.0s |
| P16.24 | Extension card identification time of slot 1 | 0.0–600.0s<br>When this parameter is set to 0.0, identification fault detection is not performed. | 0.0–600.00 | 0.0 |
| P16.25 | Extension card identification time of slot 2 | 0.0–600.0s<br>When this parameter is set to 0.0, identification fault detection is not performed. | 0.0–600.00 | 0.0 |
| P16.26 | Extension card identification | 0.0–600.0s<br>When this parameter is set to 0.0, | 0.0–600.00 | 0.0 |

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| | time of slot 3 | identification fault detection is not performed. | | |
| P16.27 | Extension card communication timeout time of slot 1 | 0.0–600.0s<br>When this parameter is set to 0.0, disconnection fault detection is not performed. | 0.0–600.00 | 0.0 |
| P16.28 | Extension card communication timeout time of slot 2 | 0.0–600.0s<br>When this parameter is set to 0.0, disconnection fault detection is not performed. | 0.0–600.00 | 0.0 |
| P16.29 | Extension card communication timeout time of slot 3 | 0.0–600.0s<br>When this parameter is set to 0.0, disconnection fault detection is not performed. | 0.0–600.00 | 0.0 |
| P16.30 | EtherCAT communication timeout time | 0.0 (invalid)–300.0s | 0.0–300.0 | 0.0s |
| P16.31 | PROFINET communication timeout time | 0.0 (invalid)–300.0s | 0.0–300.0 | 0.0s |
| P16.32 | Received PZD2 | 0: Invalid<br>1: Set frequency (0–Fmax, unit: 0.01 Hz)<br>2: PID reference (0–1000, in which 1000 corresponds to 100.0%)<br>3: PID feedback (0–1000, in which 1000 corresponds to 100.0%)<br>4: Torque setting (-3000–+3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>5: Setting of the upper limit of forward running frequency (0–Fmax, unit: 0.01 Hz)<br>6: Setting of the upper limit of reverse running frequency (0–Fmax, unit: 0.01 Hz)<br>7: Upper limit of the electromotive torque (0–3000, in which 1000 | 0–31 | 0 |
| P16.33 | Received PZD3 | | 0–31 | 0 |
| P16.34 | Received PZD4 | | 0–31 | 0 |
| P16.35 | Received PZD5 | | 0–31 | 0 |
| P16.36 | Received PZD6 | | 0–31 | 0 |
| P16.37 | Received PZD7 | | 0–31 | 0 |
| P16.38 | Received PZD8 | | 0–31 | 0 |
| P16.39 | Received PZD9 | | 0–31 | 0 |
| P16.40 | Received PZD10 | | 0–31 | 0 |
| P16.41 | Received PZD11 | | 0–31 | 0 |
| P16.42 | Received PZD12 | | 0–31 | 0 |

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| | | corresponds to 100.0% of the rated current of the motor)<br>8: Upper limit of the brake torque (0–3000, in which 1000 corresponds to 100.0% of the rated current of the motor)<br>9: Virtual input terminal command, 0x000–0x3FF (corresponding to S8, S7, S6, S5, HDIB, HDIA, S4, S3, S2, and S1 in sequence)<br>10: Virtual output terminal command, 0x00–0x0F (corresponding to RO2, RO1, HDO, and Y1 in sequence)<br>11: Voltage setting (for V/F separation) (0–1000, in which 1000 corresponds to 100.0% of the rated voltage of the motor)<br>12: AO output setting 1 (-1000–+1000, in which 1000 corresponds to 100.0%)<br>13: AO output setting 2 (-1000–+1000, in which 1000 corresponds to 100.0%)<br>14: MSB of position reference (signed number)<br>15: LSB of position reference (unsigned number)<br>16: MSB of position feedback (signed number)<br>17: LSB of position feedback (unsigned number)<br>18: Position feedback setting flag (position feedback can be set only after this flag is set to 1 and then to 0)<br>19–31: Reserved | | |
| P16.43 | Transmitted PZD2 | 0: Invalid<br>1: Running frequency (×100, Hz)<br>2: Set frequency (×100, Hz)<br>3: Bus voltage (×10, V)<br>4: Output voltage (×1, V) | 0–31 | 0 |
| P16.44 | Transmitted PZD3 | | 0–31 | 0 |
| P16.45 | Transmitted | | 0–31 | 0 |

| Function code | Name | Parameter description | Setting range | Default value |
|---|---|---|---|---|
| | PZD4 | 5: Output current (×10, A) | | |
| P16.46 | Transmitted PZD5 | 6: Actual output torque (×10, %)<br>7: Actual output power (×10, %) | 0–31 | 0 |
| P16.47 | Transmitted PZD6 | 8: Rotating speed of the running (×1, RPM) | 0–31 | 0 |
| P16.48 | Transmitted PZD7 | 9: Linear speed of the running (×1, m/s)<br>10: Ramp frequency reference | 0–31 | 0 |
| P16.49 | Transmitted PZD8 | 11: Fault code<br>12: AI1 value (×100, V) | 0–31 | 0 |
| P16.50 | Transmitted PZD9 | 13: AI2 value (×100, V)<br>14: AI3 value (×100, V) | 0–31 | 0 |
| P16.51 | Transmitted PZD10 | 15: HDIA frequency (×100, kHz)<br>16: Terminal input state | 0–31 | 0 |
| P16.52 | Transmitted PZD11 | 17: Terminal output state<br>18: PID reference (×100, %) | 0–31 | 0 |
| P16.53 | Transmitted PZD12 | 19: PID feedback (×100, %)<br>20: Rated torque of the motor<br>21: MSB of position reference (signed number)<br>22: LSB of position reference (unsigned number)<br>23: MSB of position feedback (signed number)<br>24: LSB of position feedback (unsigned number)<br>25: State word<br>26: HDIB frequency value (×100, kHz)<br>27–31: Reserved | 0–31 | 0 |

Service line:86-755-86312859 Website:www.invt.com

The products are owned by **Shenzhen INVT Electric Co.,Ltd.**
Two companies are commissioned to manufacture: (For product code, refer to the 2nd/3rd place of S/N on the name plate.)

**Shenzhen INVT Electric Co., Ltd.** (origin code: 01)
Address: INVT Guangming Technology Building, Songbai Road, Matian, Guangming District,Shenzhen, China

**INVT Power Electronics (Suzhou) Co., Ltd.** (origin code: 06)
Address: 1# Kunlun Mountain Road, Science&Technology Town, Gaoxin District, Suzhou, Jiangsu, China

**Industrial Automation:** ■Inverter ■Servo & Motion Control ■Motor & Electric Spindle ■PLC ■HMI ■Intelligent Elevator Control System ■Traction Drive

**Electric Power:** ■SVG ■Solar Inverter ■UPS ■Online Energy Management System

66001-00600



201906 (V1.1)