



USER'S MANUAL

NX FREQUENCY CONVERTERS


CANOPEN OPTION BOARD OPT-C6

| | | |
|-----|--|----|
| 1. | GENERAL | 3 |
| 2. | CANOPEN OPTION BOARD TECHNICAL DATA | 4 |
| 2.1 | General | 4 |
| 2.2 | CanOpen cable | 5 |
| 3. | CANOPEN | 6 |
| 4. | CANOPEN OPTION BOARD LAYOUT AND CONNECTIONS | 7 |
| 4.1 | CanOpen option board..... | 7 |
| 4.2 | Bus terminal resistors | 7 |
| 4.3 | LED indications..... | 8 |
| 4.4 | Connection of CanOpen bus cable..... | 9 |
| 5. | INSTALLATION OF VACON NX CANOPEN BOARD | 14 |
| 5.1 | Board information sticker..... | 16 |
| 6. | COMMISSIONING | 17 |
| 7. | CANOPEN-VACON NX INTERFACE | 19 |
| 7.1 | CanOpen message frame | 19 |
| 7.2 | Network Management (NMT) | 21 |
| 7.3 | Process data (PDO) | 22 |
| 7.4 | Transmission types | 23 |
| 7.5 | Controlling the drive via PDO messages..... | 24 |
| 7.6 | Using manufacturer specific PDOs | 27 |
| 7.7 | Drive Monitoring..... | 29 |
| 7.8 | Anyparameter service..... | 30 |
| 8. | Service Data (SDO) | 31 |
| 9. | Node Guarding Protocol | 41 |
| 10. | Electronic Data Sheet, EDS-file | 42 |
| 11. | APPENDICES | 43 |
| | appendix a - device profile for drives..... | 43 |
| | appendix b – sdo messages | 44 |
| | appendix c – process data contents..... | 45 |
| | appendix d..... | 46 |

1. GENERAL

Vacon NX frequency converters can be connected to the CanOpen system using a fieldbus board. The converter can then be controlled, monitored and programmed from the Host system.

The CanOpen fieldbus board shall be installed in slot E on the control board of the frequency converter.

| | |
|--|---|
|  <p>DANGER</p> | <p>Internal components and circuit boards are at high potential when the frequency converter is connected to the power source. This voltage is extremely dangerous and may cause death or severe injury if you come into contact with it.</p> |
|--|---|

2. CANOPEN OPTION BOARD TECHNICAL DATA

2.1 General

| | | |
|---------------------|-------------------------------|--|
| CanOpen connections | Interface | Open Style Connector (Pluggable connector, 5.08mm) |
| | Data transfer method | CAN (ISO 11898) |
| | Transfer cable | 2 wire twisted shielded cable |
| | Electrical isolation | 500 VDC |
| Communications | CanOpen | CiA DS-301 |
| | | CiA DSP-402 |
| | Baud rate | 10 kBaud |
| | | 20 kBaud |
| 50 kBaud | | |
| 100 kBaud | | |
| | 125 kBaud | |
| | 250 kBaud | |
| | 500 kBaud | |
| | 1000 kBaud | |
| | Addresses | 1 – 127 |
| Environment | Ambient operating temperature | -10°C...55°C |
| | Storing temperature | -40°C...60°C |
| | Humidity | <95%, no condensation allowed |
| | Altitude | Max. 1000 m |
| | Vibration | 0.5 G at 9...200 Hz |
| Safety | | Fulfils EN50178 standard |

Table 2-1. CanOpen technical data

2.2 CanOpen cable

According to the ISO 11898 standard, cables to be chosen for CAN bus lines should have a nominal impedance of 120Ω , and a specific line delay of nominal 5 ns/m . Line termination has to be provided through termination resistors of 120Ω located at both ends of the line. The length related resistance should have $70\text{ m}\Omega/\text{m}$. All these mentioned AC and DC parameters are suitable for a 1 Mbit/s transmission rate.

The table below shows practical bus length for CANopen networks with less than 64 nodes:

| Baudrate [Kbit/s] | Max. Bus length [m] |
|-------------------|---------------------|
| 1000 | 30 |
| 800 | 50 |
| 500 | 100 |
| 250 | 250 |
| 125 | 500 |
| 50 | 1000 |
| 20 | 2500 |

3. CANOPEN

CANopen is a networking system based on the serial bus Controller Area Network (CAN). The CANopen Communication Profile (CiA DS-301) supports both direct access to device parameters and time-critical process data communication. CANopen device profiles (CiA DS-40x) define standards for basic device functionality while providing ample scope for additional vendor-specific device features. CANopen leashes the full power of CAN by allowing direct peer to peer data exchange between nodes in an organised and, if necessary, deterministic manner. The network management functions specified in CANopen simplify project design, implementation and diagnosis by providing standard mechanisms for network start-up and error management.

CANopen supports both-cyclic and event-driven communication. This makes it possible to reduce the bus load to a minimum but still maintaining extremely short reaction times. High communication performance can be achieved at relatively low baud rates, thus reducing EMC problems and minimising cable costs.

CANopen is the ideal networking system for all types of automated machinery. One of the distinguishing features of CANopen is its support for data exchange at the supervisory control level as well as accommodating the integration of very small sensors and actuators on the same physical network. This avoids the unnecessary expense of gateways linking sensor/actuator bus systems with higher communication networks and makes CANopen particularly attractive to original equipment manufacturers.

Device Profile Drives and Motion Control (CiA DSP-402) document represents the standardised CANopen Device Profile for digital controlled motion products like servo controllers, frequency converters or stepper motors. All the above-mentioned devices use communication techniques which conform to those described in the CANopen Application Layer and Communication Profile. The starting and stopping of the drive and several mode specific commands are executed by the statemachine. The operation mode defines the behaviour of the drive. The following modes are defined in this profile:

- Homing Mode
- Profile Position Mode
- Interpolated Position Mode
- Profile Velocity Mode
- Profile Torque Mode
- Velocity Mode

Vacon CanOpen Option Board supports the **Velocity Mode**

4. CANOPEN OPTION BOARD LAYOUT AND CONNECTIONS

Vacon CanOpen Board is connected to the fieldbus through a 5-pin pluggable bus connector (board NXOPTC6).

The communication with the control board of the frequency converter takes place through the standard Vacon Interface Board Connector.

4.1 CanOpen option board

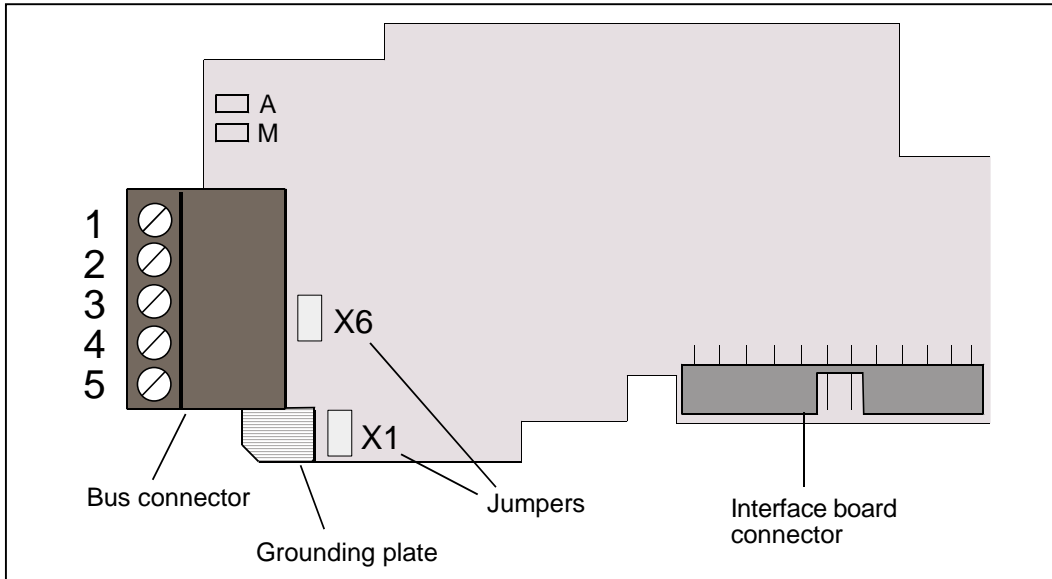


Figure 4-1. Vacon CanOpen option board OPT-C6

| Signal | Connector | Description |
|------------|-----------|--------------------------------|
| CAN_GND | 1 | Ground / 0V / V- |
| CAN_L | 2 | CAN_L bus line (dominant low) |
| (CAN_SHLD) | 3 | Optional CAN shield |
| CAN_H | 4 | CAN_H bus line (dominant high) |
| (CAN_V+) | 5 | - |

Table 4-1. OPT-C6 bus connector signals

4.2 Bus terminal resistors

If Vacon drive is the last device of the CanOpen line the bus termination must be set. Use jumper X6 (ON position see Figure 4-1), or an external resistor (120Ω) connected to terminals 2 and 4.



4.3 LED indications

The CanOpen Option Board includes two LED status indicators next to the connector: Fieldbus status (M), CanOpen(A). Led N is unused.

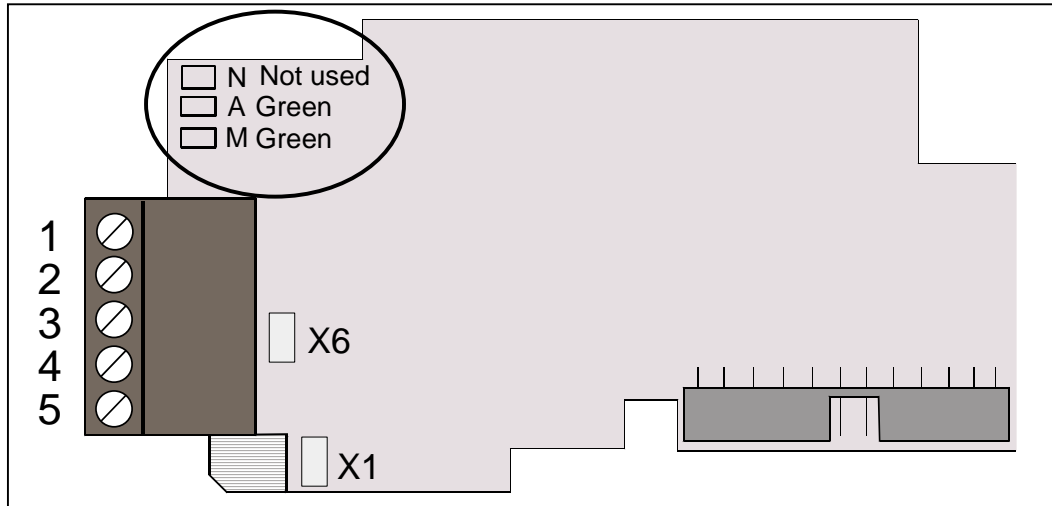


Figure 4-2. LED indications on the CANopen board

CanOpen board status LED (A) GREEN

| LED is: | Meaning: |
|-----------------------------|--|
| OFF | Option board not activated |
| ON | Option board in initialisation state waiting for activation command from the frequency converter |
| Blinking fast (1 blink / s) | Option board is activated and in RUN state Option board is ready for external communication |
| Blinking slow (1 blink/ 5s) | Option board is activated and in FAULT state Internal fault of option board |

Fieldbus status LED (M) GREEN

| LED is: | Meaning: |
|------------------------------|--|
| OFF | Fieldbus module is waiting for parameters from the frequency converter No external communication |
| ON | Fieldbus module is activated Parameters received and module activated Module is waiting for messages from the bus |
| Blinking fast (1 blink / s) | Module is activated and receiving messages from the bus |
| Blinking slow (1 blink / 5s) | Module is in FAULT state No messages from Master within the watchdog time Bus broken, cable loose or Master off line |

4.4 Connection of CanOpen bus cable

The bus cable shield can be grounded in three different ways:

- a) clamping the cable to the frequency converter frame
- b) to the frame of the frequency converter through an RC filter
- c) directly to the converter frame

Note: Normally, the option board has already been installed in slot E of the control board. It is not necessary to detach the whole board for the grounding of the bus cable shield. Just detach the terminal block.

4.4.1 Grounding by clamping the cable to the converter frame

This manner of grounding is the most effective and especially recommended when the distances between the devices are relatively short (see 4.4.2.1).

In this manner of grounding, the position of jumper X1 is of no importance

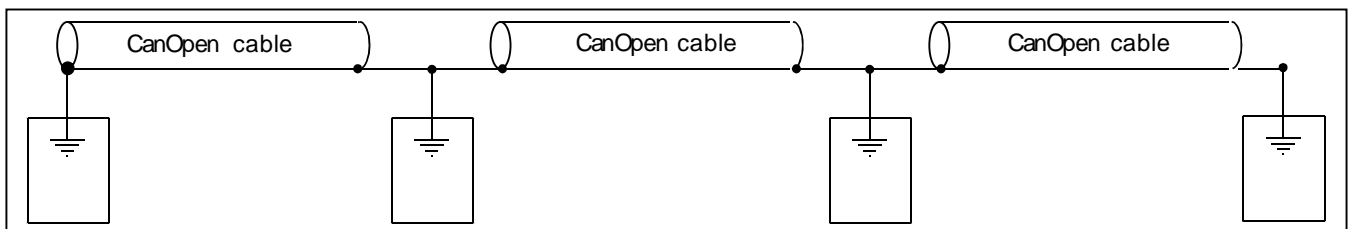


Figure 4-3. Grounding by clamping the cable to the converter frame

- 1 Strip about 5 cm of the CanOpen cable in the same way as shown in Figure 4-4 but **cut off the grey cable shield**. Remember to do this for both bus cables (except for the last device).
- 2 Leave no more than 1 cm of the data cable outside the terminal block and strip the data cables at about 0.5 cm to fit in the terminals. See Figures 4-5 and 4-6.
Note: Do this for both bus cables
- 3 Insert the data cables of **both CanOpen cables** into terminals #2 and #4. See Figure 4-7.
- 4 Strip the CanOpen cable at such a distance from the terminal that you can fix it to the frame with the grounding clamp. See Figure 4-4. **Error! Reference source not found.**

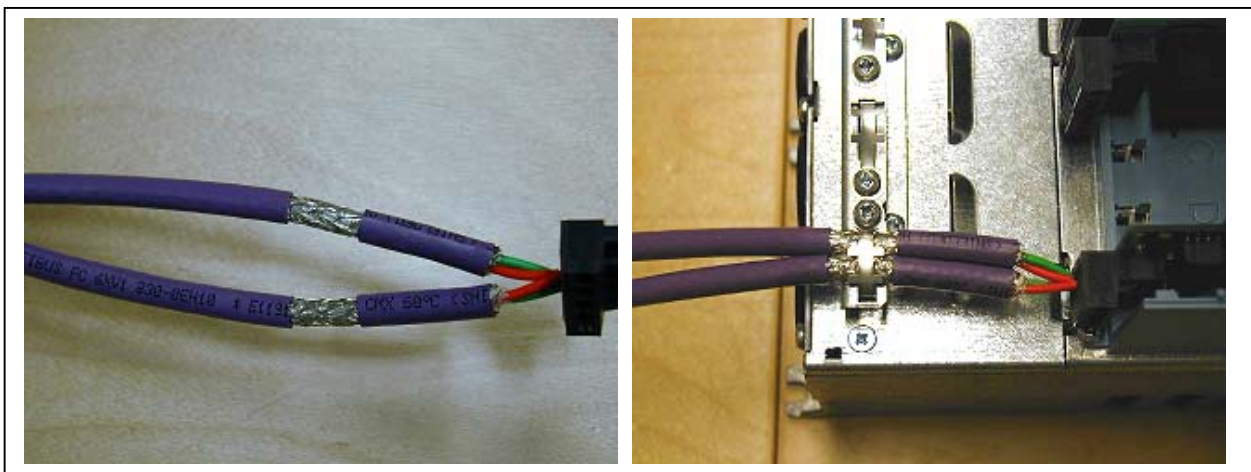


Figure 4-4.

4.4.2 Grounding the bus cable shield directly to the frequency converter frame using jumper X1

- 1 Set jumper X1 in ON position:

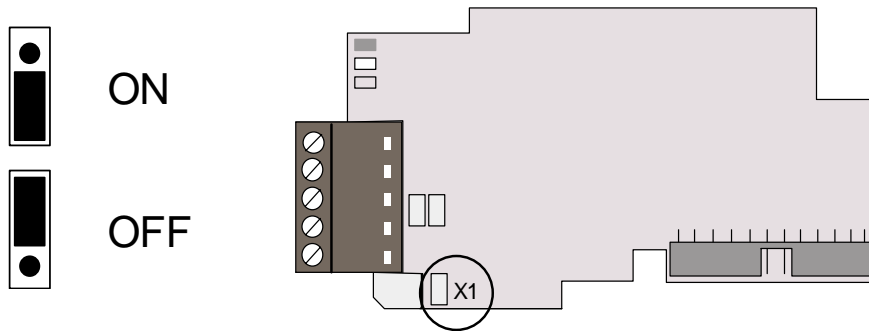


Figure 4-5. Jumper X1 positions

- 2 Strip about 5 cm of the CanOpen cable as shown in the picture.
Note: Do the same for both bus cables (except for the last device). However, since the grounding shall be done on one cable only cut off the exposed part of the other grounding cable.

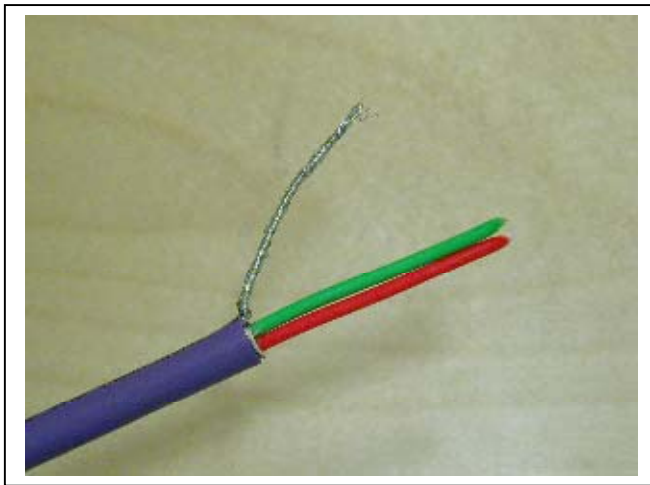


Figure 4-6.

- 3 Leave no more than 1 cm of the red and green data cable outside the terminal block and strip the data cables at about 0.5 cm to fit in the terminals. See pictures below.
Note: Do this for both bus cables.

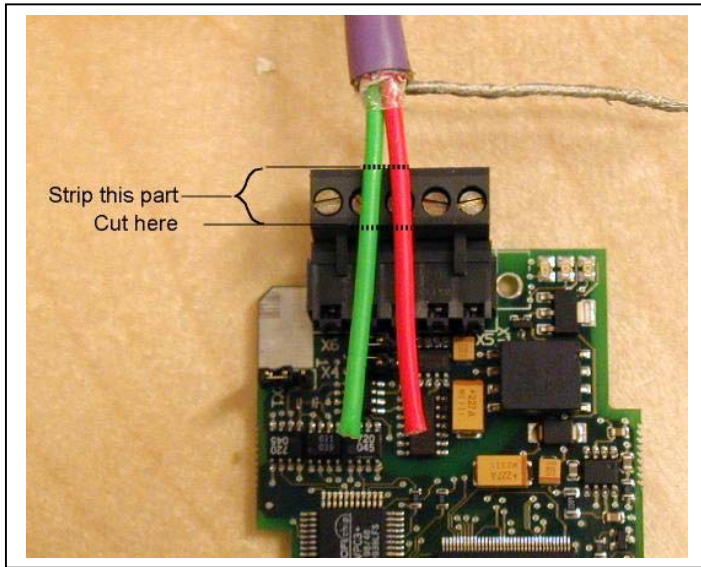


Figure 4-7.

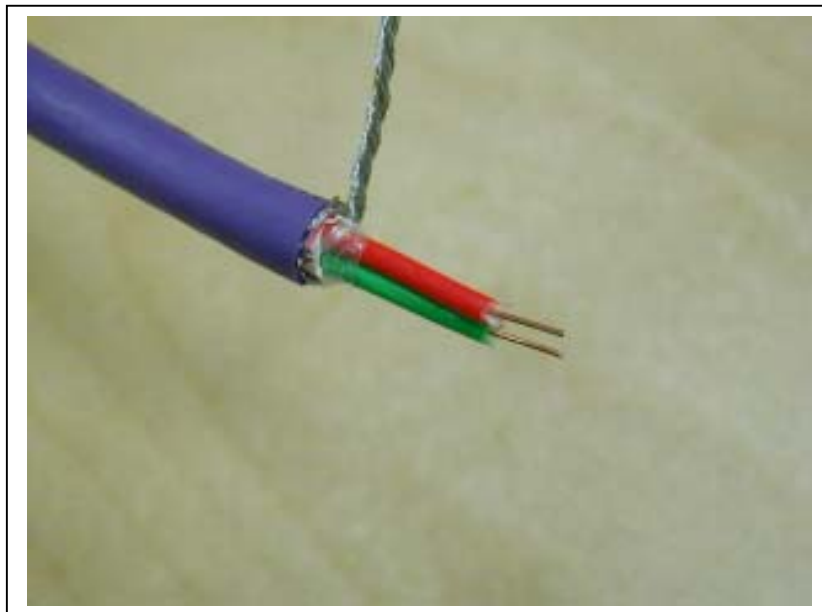


Figure 4-8.

- 4 We recommend you to use an Abico connector to fit the grounding cable into the grounding terminal (#3).
Insert the white and brown data cables of **both CanOpen cables** into terminals #2 (white) and #4 (brown).

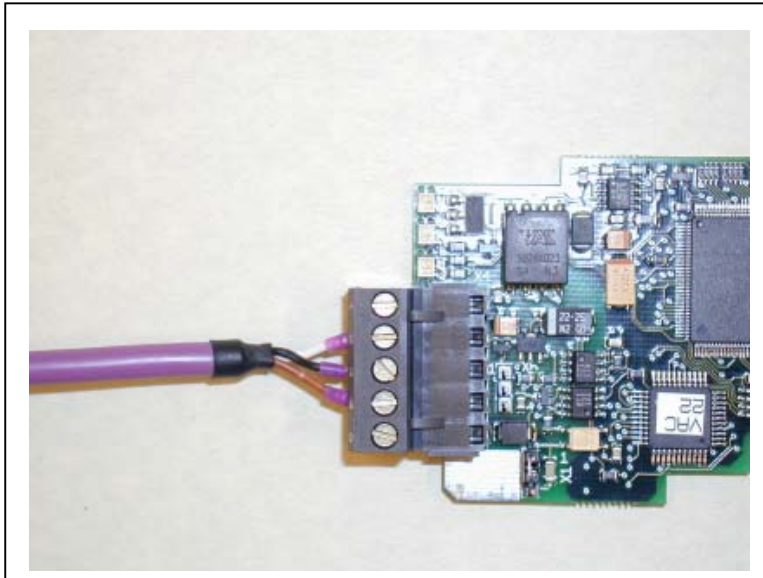
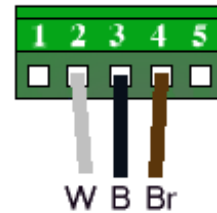


Figure 4-9.



- 5 Place the CanOpen board into slot E of the control board (see board installation on page 14) and fix both the CanOpen cables on the frame with the clamp.

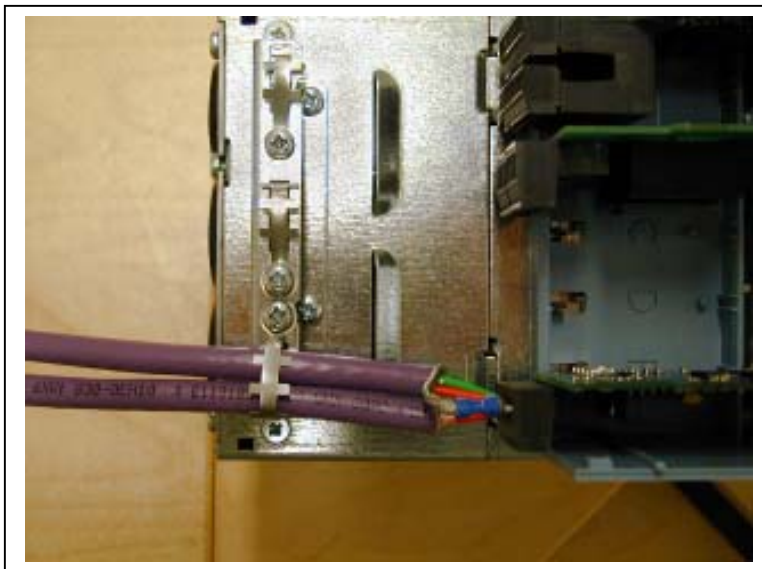


Figure 4-10.

4.4.2.1 *Grounding the bus cable shield directly to the frequency converter frame using an RC-filter*

We recommend you to do the grounding in this manner when the distance between the devices exceeds 50 meters (55 yds.). When the distance between the devices is long disturbances (e.g. voltage spikes) are more likely to appear. In this grounding method, the disturbances are filtered out. Even if the ground planes of A, B and C are different (which is very typical e.g. in construction) there is no current between them because the points do not have a ground connection.

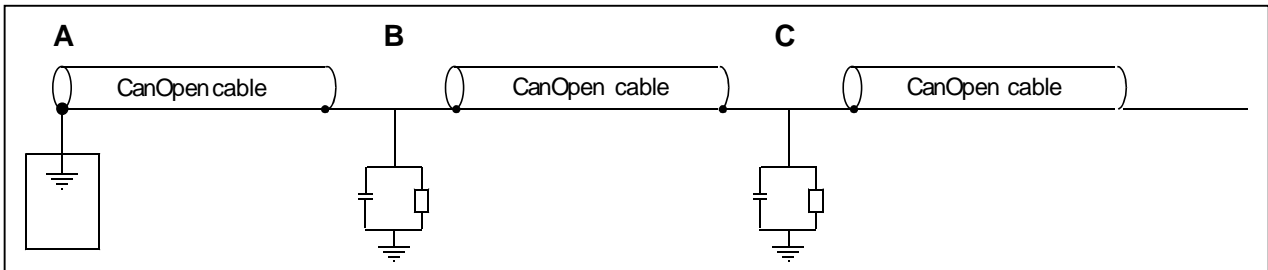


Figure 4-11. Grounding with RC filter

- 1 Set jumper X1 in OFF position

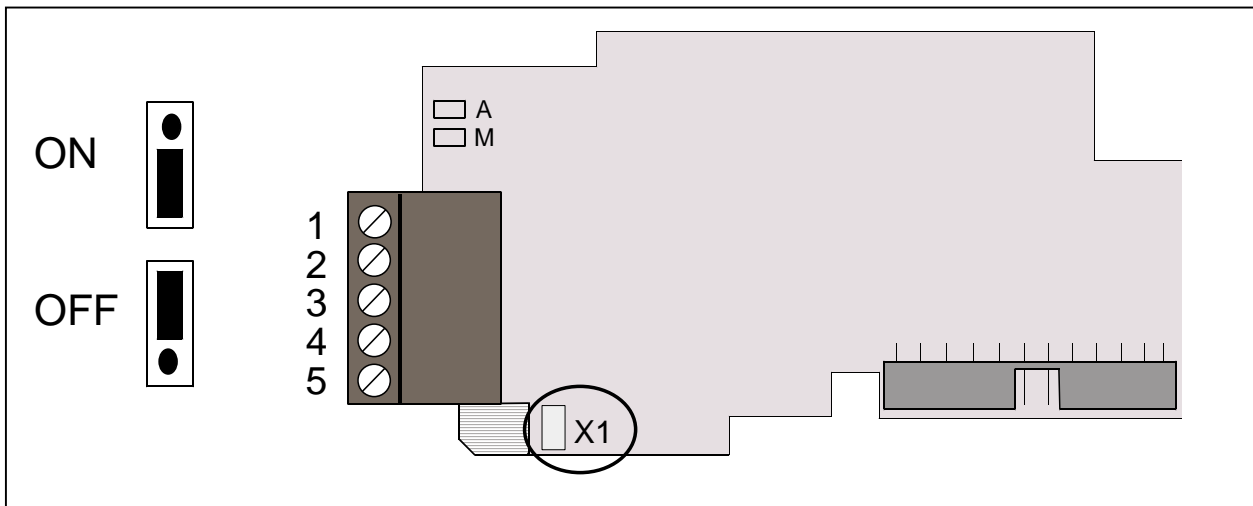



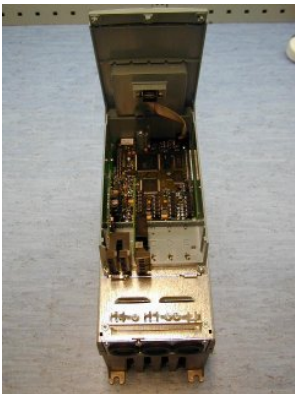





Figure 4-12. Jumper X1 positions

- 2 Carry out the grounding in the same way as advised in Chapter 4.4.1.

5. INSTALLATION OF VACON NX CANOPEN BOARD

| | |
|--|--|
|  NOTE | <p>MAKE SURE THAT THE FREQUENCY CONVERTER IS SWITCHED OFF BEFORE AN OPTION OR FIELDBUS BOARD IS CHANGED OR ADDED!</p> |
|--|--|

| | | |
|----------|--|---|
| A | <p>Vacon NX frequency converter</p> |  |
| B | <p>Remove the cable cover.</p> |  |
| C | <p>Open the cover of the control unit.</p> |  |

| | | |
|-----------------|--|--|
| <p>D</p> | <p>Install CanOpen option board in slot E on the control board of the frequency converter. Make sure that the grounding plate (see below) fits tightly in the clamp.</p> |  |
| <p>E</p> | <p>Make a sufficiently wide opening for your cable by cutting the grid as wide as necessary.</p> |  |
| <p>F</p> | <p>Close the cover of the control unit and the cable cover.</p> |  |

5.1 Board information sticker

The CanOpen option board package delivered by the factory includes a sticker (shown below). Please mark the board type (1), the slot into which the board is mounted (2) and the mounting date (3) on the sticker. Finally, attach the sticker on your drive.

Drive modified:

Option board: NXOPT..... Date:.....
 in slot: A B C D E

IP54 upgrade/Collar Date:.....

EMC level modified: H T / T H Date:.....

6. COMMISSIONING

READ FIRST CHAPTER 8 'COMMISSIONING' IN VACON NX USER'S MANUAL (Document nr. ud00701, please visit <http://www.vacon.com/support/documents.html>).

Note! You must select Fieldbus as the active control place, if you wish to control the frequency converter through fieldbus. See Vacon NX User's Manual, Chapter 7.3.3.1

The Vacon CanOpen board is commissioned with the control keypad by giving values to appropriate parameters in menu **M7** (for locating the expander board menu see Vacon NX User's Manual, Chapter 7).

Expander board menu (M7)

The *Expander board menu* makes it possible for the user 1) to see what expander boards are connected to the control board and 2) to reach and edit the parameters associated with the expander board.

Enter the following menu level (**G#**) with the *Menu button right*. At this level, you can browse through slots A to E with the *Browser buttons* to see what expander boards are connected. On the lowermost line of the display you also see the number of parameter groups associated with the board.

If you still press the *Menu button right* once you will reach the parameter group level where there are two groups: Editable parameters and Monitored values. A further press on the *Menu button right* takes you to either of these groups.

CanOpen parameters

To commission the CanOpen board, enter the level P7.5.1.# from the *Parameters* group (G7.5.1). Give desired values to all CanOpen parameters (see xx and xx).

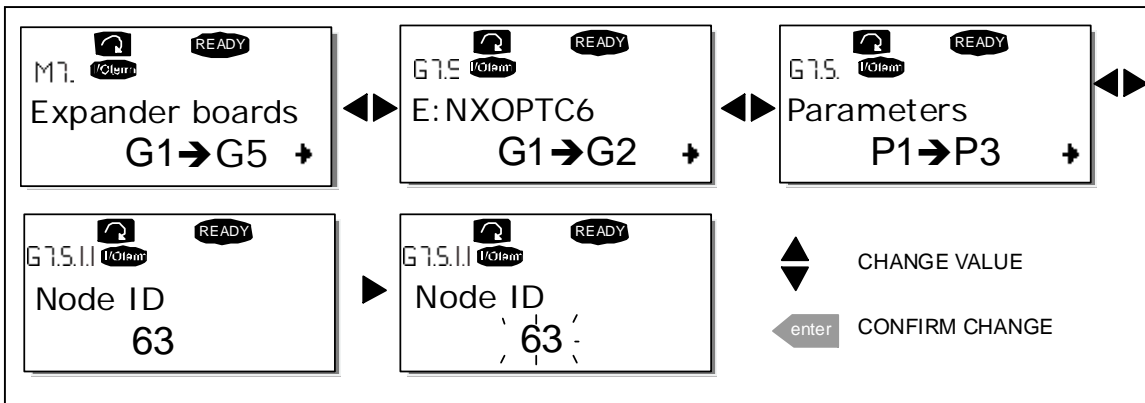


Figure 6-1. Changing the CANopen option board parameters

| # | Name | Default | Range | Description |
|---|--------------|---------|--|--|
| 1 | Node ID | 1 | 1 – 127 | |
| 2 | BAUD RATE | 6 | 1 – 10 kBaud 2 – 20 kBaud 3 – 50 kBaud 4 – 100 kBaud 5 – 125 kBaud 6 – 250 kBaud 7 – 500 kBaud 8 – 1000 kBaud | Communication speed |
| 3 | Operate mode | 1 | 1 – Drive Profile 2 – ByPass | Communication set selection 1 = xPDO1, xPDO6 2 = xPDO1, xPDO21, xPDO22 |

Table 6-1. CANopen parameters

The parameters of every device must be set before connecting to the bus. Especially the parameters “NODE ID” and “BAUD RATE” must be the same as in the master configuration.

CanOpen status

To see the present status of the CanOpen Fieldbus, enter the *CanOpen Status* page from *Monitor menu (G7.5.2)*. See picture and table below.

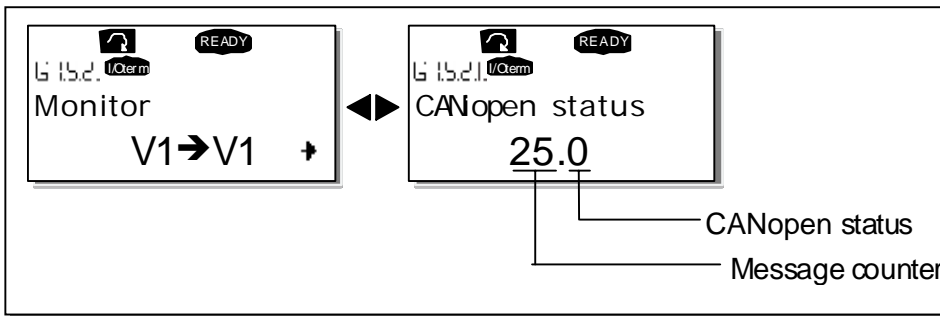


Figure 6-2. DeviceNet status

| CanOpen status | |
|----------------|-------------------|
| 0 | INITIALISING |
| 4 | STOPPED |
| 5 | OPERATIONAL |
| 6 | PRE_OPERATIONAL |
| 7 | RESET_APPLICATION |
| 8 | RESET_COMM |
| 9 | UNKNOWN |

Table 6-2. CANopen status indications

7. CANOPEN-VACON NX INTERFACE

CanOpen communication objects transmitted via the CAN network are described by services and protocols. They are classified as follows:

- The real-time data transfer is performed by the Process Data Objects (PDOs) protocol
→ REAL-TIME CONTROL OF THE DRIVE
- Service Data Objects (SDO) protocols provide the read and write access to entries of a device object dictionary
→ CONFIGURATION OF THE DRIVE, READ/WRITE DRIVE PARAMETERS
- The Network Management (NMT) protocols provide services for network initialization, error control and device status control
→ START/STOP CANOPEN COMMUNICATION

7.1 CanOpen message frame

| | | | | | | | |
|------|---------------|------|------|---------------------|--------|-------|-------|
| SOF | COB-ID | RTR | CTRL | Data Segment | CRC | ACK | EOF |
| 1bit | 11bit | 1bit | 5bit | 0-8bytes | 16bits | 2bits | 7bits |

| | | | |
|------|----------------------------------|-----|-------------------------|
| SOF | Start of Frame | CRC | Cyclic Redundancy Check |
| RTR | Remote Transmission Request | ACK | Acknowledge |
| CTRL | Control Field (i.e. Data Length) | EOF | End of Frame |

COB-ID

The identification field of the CANopen-message is 11 bits.

| | | | | | | | | | | | |
|--------|---------------|---|---|---|-----------|---|---|---|---|---|---|
| ID-Bit | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| COB-ID | Function Code | | | | Module-ID | | | | | | |

The default identification field consists of a functional part and a module-ID part. The functional part determines the object priority. This kind of identification field allows communication between a master and 127 slaves. Broadcasting is indicated by a module-id of zero. Function codes are determined with object dictionaries in device profiles.

Predefined connection sets

CanOpen option board has two different communication parameter sets. These sets can be selected via Operate Mode –parameter from panel. Both communication sets fullfils 'Drives And Motor Control'-profile described in CiA DSP-402.

| Drive Profile - mode | | | |
|---------------------------------|------------------------|--------------|--------------------------|
| Object | Function Code (binary) | COB-ID | Comm. parameter at Index |
| NMT message | 0000 | 0x0000 | - |
| Sync Message | 0001 | 0x0080 | 0x1005 |
| Time-Stamp-Message | 0001 | 0x0100 | - |
| PDO1, Process Data Objects (tx) | 0011 | 0x0180 +Node | 0x1800 |
| PDO1, Process Data Objects (rx) | 0100 | 0x0200 +Node | 0x1400 |
| PDO6, Process Data Objects (tx) | 0101 | 0x0280 +Node | 0x1801 |
| PDO6, Process Data Objects (rx) | 0110 | 0x0300 +Node | 0x1401 |
| SDO, Service Data Objects (tx) | 1011 | 0x0580 +Node | |
| SDO, Service Data Objects (rx) | 1100 | 0x0600 +Node | |
| Node Guarding | 1110 | 0x0700 +Node | (0x100E) |

Table 7-1. Drive Profile -mode

| ByPass - mode | | | |
|----------------------------------|------------------------|--------------|--------------------------|
| Object | Function Code (binary) | COB-ID | Comm. parameter at Index |
| NMT message | 0000 | 0x0000 | - |
| Sync Message | 0001 | 0x0080 | 0x1005 |
| Time-Stamp-Message | 0010 | 0x0100 | - |
| PDO1, Process Data Objects (tx) | 0011 | 0x0180 +Node | 0x1800 |
| PDO1, Process Data Objects (rx) | 0100 | 0x0200 +Node | 0x1400 |
| PDO21, Process Data Objects (tx) | 0111 | 0x0380 +Node | 0x1814 |
| PDO21, Process Data Objects (rx) | 1000 | 0x0400 +Node | 0x1414 |
| PDO22, Process Data Objects (tx) | 1001 | 0x0480 +Node | 0x1815 |
| PDO22, Process Data Objects (rx) | 1010 | 0x0500 +Node | 0x1415 |
| SDO, Service Data Objects (tx) | 1011 | 0x0580 +Node | |
| SDO, Service Data Objects (rx) | 1100 | 0x0600 +Node | |
| Node Guarding | 1110 | 0x0700 +Node | (0x100E) |

Table 7-2. Bypass -mode

7.2 Network Management (NMT)

The CanOpen network management is node-oriented and follows a master/slave structure. It requires one device in the network, which fulfills the function of the NMT master. The other nodes are NMT slaves.

The CanOpen NMT slave devices implement a state machine, see picture below. After power on a node initialises and transits to the "Pre-operational State". In this state communication across SDO channels is possible for node configuration, but not yet across PDOs. With the NMT message "Start Remote Node", a selected or any nodes on the network can be set into the "Operational State". In this state, also the exchange of data by means of PDOs is possible. With enabling the operation of all nodes of a network at the same time, a co-ordinated operation of the communicating system is secured (DS301).

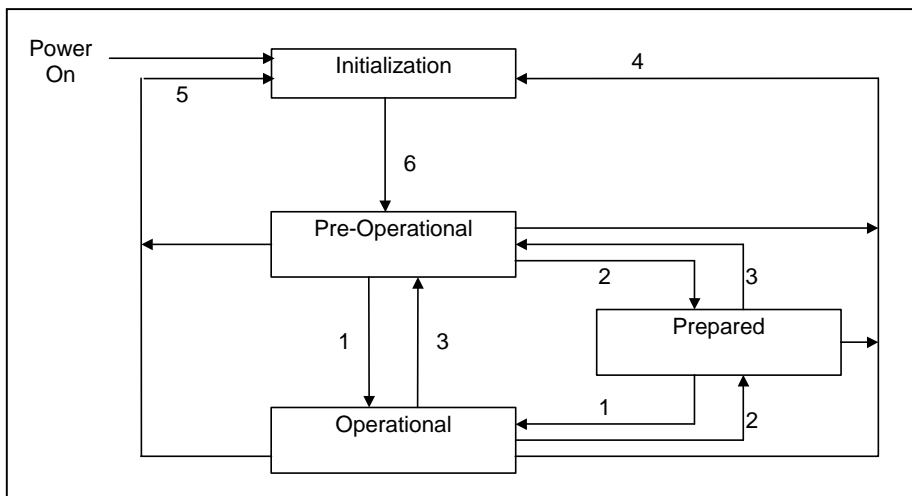
To Set Vacon CanOpen Option board to the "Operational State" the following message must be sent:

Message: Start Remote Node

Master to Slave (1)

| Header | | | Data | | | | | | | |
|--------|-----|-----|--------|-------------|---|---|---|---|---|---|
| ID | RTR | Len | 1 = CS | 2 = Node ID | 3 | 4 | 5 | 6 | 7 | 8 |
| 0000 | 0 | 2 | 01 | 01 | | | | | | |

Function of the internal state machine:



| Change | Message / Event | Command Specifier (CS) |
|--------|-----------------------------|------------------------|
| 1 | Start Remote Node | CS = 1 |
| 2 | Stop Remote Node | CS = 2 |
| 3 | Enter Pre-operational State | CS = 128 |
| 4 | Reset Node | CS = 129 |
| 5 | Reset Communication | CS = 130 |
| 6 | Initialisation finished | Automatic |

7.3 Process data (PDO)

The real-time data transfer is performed by means of "Process Data Objects" (PDO). The transfer of PDOs is performed with no protocol overhead. Process Data is time-critical data used for control of the drive and monitor status of the drive. Transmit PDOs support several transmission modes which are cyclic, acyclic, synchronous, asynchronous and RTR only mode. Most PDOs support also Event Timer for transmitting PDOs. TPD01 is exception which is defined for asynchronous only operation. Receive PDOs support only asynchronous transmission mode (event driven). Vacon CanOpen option board uses 8 types of PDOs as follows:

| PDO Type | Mapped Data | Mapped Data | Mapped Data | Mapped Data | Cyclic | Acyclic | Synchronous | Asynchronous (default) | RTR Only mode | Event Timer |
|----------|-------------------|--------------------|--------------------|--------------------|--------|---------|-------------|------------------------|---------------|-------------|
| TPD01 | statusword | - | - | - | | | | x | | |
| TPD06 | statusword | vl_control_effort | - | - | X | X | X | x | X | X |
| TPD021 | nx_status_word | nx_actual_speed | process_data_out 1 | process_data_out 2 | X | X | X | x | X | X |
| TPD022 | process_data_out3 | process_data_out4 | process_data_out5 | process_data_out6 | X | X | X | x | X | X |
| RPD01 | controlword | - | - | - | | | | x | | |
| RPD06 | controlword | vl_target_velocity | - | - | | | | x | | |
| RPD021 | nx_control_word | nx_speed_reference | process_data_in1 | process_data_in2 | | | | x | | |
| RPD022 | process_data_in3 | process_data_in4 | process_data_in5 | process_data_in6 | | | | x | | |

Note: manufacturer specific PDOs (TPD021/RPD021/ TPD022/RPD022) consist of application specific process data. See appendix for contents of these processdatas in different applications. See Chapter 7.6 (Using manufacturer specific PDOs) or a specific application manual.

Note! all Tx objects are event-driven by default. (Message is sent out if one or several values within the message change)

7.4 Transmission types

| transmission type | PDO transmission | | | | |
|-------------------|------------------|---------|-----------------|------------------|----------|
| | cyclic | acyclic | synchronou s | asynchronou s | RTR only |
| 0 | | X | X | | |
| 1-240 | X | | X | | |
| 241-251 | - Reserved - | | | | |
| 252 | | | X | | X |
| 253 | | | | X | X |
| 254 | | | | X | |
| 255 | | | | X | |

Table 7-3. Description of transmission type

Synchronous (transmission types 0-240 and 252) means that the transmission of the PDO shall be related to the SYNC object. Preferably the devices use the SYNC as a trigger to output or actuate based on the previous synchronous Receive PDO respectively to update the data transmitted at the following synchronous Transmit PDO. Asynchronous means that the transmission of the PDO is not related to the SYNC object. A transmission type of zero means that the message shall be transmitted synchronously with the SYNC object but not periodically. A value between 1 and 240 means that the PDO is transferred synchronously and cyclically. The transmission type indicating the number of SYNC which are necessary to trigger PDO transmissions. Receive PDOs are always triggered by the following SYNC upon reception of data independent of the transmission types 0 - 240.

The transmission types 252 and 253 mean that the PDO is only transmitted on remote transmission request. At transmission type 252, the data is updated (but not sent) immediately after reception of the SYNC object. At transmission type 253 the data is updated at the reception of the remote transmission request (hardware and software restrictions may apply). These value are only possible for TPDOs. For TPDOs transmission type 254 means, the application event is manufacturer specific (manufacturer specific part of the Object Dictionary), transmission type 255 means, that the application event is defined in the device profile. RPDOs with that type trigger the update of the mapped data with the reception. PDOs Sub-index 3h contains the inhibit time. This time is a minimum interval for PDO transmission. The value is defined as multiple of 100ms. It is not allowed to change the value while the PDO exists (Bit 31 of sub-index 1 is 0).

In mode 254/255 additionally an event time can be used for TPDO. If an event timer exists for a TPDO (value not equal to 0) the elapsed timer is considered to be an event. The event timer elapses as multiple of 1 ms of the entry in sub-index 5h of the TPDO. This event will cause the transmission of this TPDO in addition to otherwise defined events. The occurrence of the events set the timer. Independent of the transmission type the RPDO event timer is used recognize the expiration of the RPDO.

7.5 Controlling the drive via PDO messages

PDO1 Rx

Master to Slave (1)

| Header | | | Data | | | | | | | |
|--------|-----|-----|--------------------|---|---|---|---|---|---|---|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x201 | 0 | 2 | controlword | | - | - | - | - | - | - |

PDO1 Tx

Slave (1) to Master

| Header | | | Data | | | | | | | |
|--------|-----|-----|-------------------|---|---|---|---|---|---|---|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x181 | 0 | 2 | statusword | | - | - | - | - | - | - |

PDO6 Rx

Master to Slave (1)

| Header | | | Data | | | | | | | |
|--------|-----|-----|--------------------|---|---------------------------|---|---|---|---|---|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x301 | 0 | 4 | controlword | | vl_target_velocity | | - | - | - | - |

PDO6 Tx

Slave (1) to Master

| Header | | | Data | | | | | | | |
|--------|-----|-----|-------------------|---|--------------------------|---|---|---|---|---|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x281 | 0 | 4 | statusword | | vl_control_effort | | - | - | - | - |

The state of the drive can be controlled by the *controlword*
The state of the drive is shown in the *statusword*.

The *Statemachine* describes the device status and the possible control sequence of the drive.

controlword

| bit | Name |
|-----|-------------------------|
| 0 | Switch ON |
| 1 | Disable Voltage |
| 2 | Quick Stop |
| 3 | Enable Operation |
| 4 | Operation Mode Specific |
| 5 | Operation Mode Specific |
| 6 | Operation Mode Specific |
| 7 | Reset Fault |
| 8 | Halt |
| 9 | Reserved |
| 10 | Reserved |
| 11 | Manufacturer Specific |
| 12 | Manufacturer Specific |
| 13 | Manufacturer Specific |
| 14 | Manufacturer Specific |
| 15 | Manufacturer Specific |

statusword

| bit | Name |
|-----|-------------------------|
| 0 | Ready to Switch ON |
| 1 | Switched ON |
| 2 | Operation Enable |
| 3 | Fault |
| 4 | Voltage Disable |
| 5 | Quick Stop |
| 6 | Swich ON Disable |
| 7 | Warning |
| 8 | Manufacturer Specific |
| 9 | Remote |
| 10 | Target Reached |
| 11 | Internal Limit Active |
| 12 | Operation Mode Specific |
| 13 | Operation Mode Specific |
| 14 | Manufacturer Specific |
| 15 | Manufacturer Specific |

By using a controlword the drive can be controlled as follows:

| Command | ControlWord | Description |
|----------------------|-------------|---|
| Prepare READY | 0006hex | Set Statemachine to "Ready To Switch ON" state |
| Prepare RUN | 0007hex | Set Statemachine to "Switched ON" state |
| RUN | 000Fhex | Start motor if "Fieldbus" is the active control place |
| STOP | 0007hex | Stop motor |
| FAULT RESET (step 1) | bit 7 = 0 | Rising edge to bit 7 |
| FAULT RESET (step 2) | bit 7 = 1 | |

vl_target_velocity

The vl_target_velocity is the required speed reference to the frequency converter.
The unit is RPM.

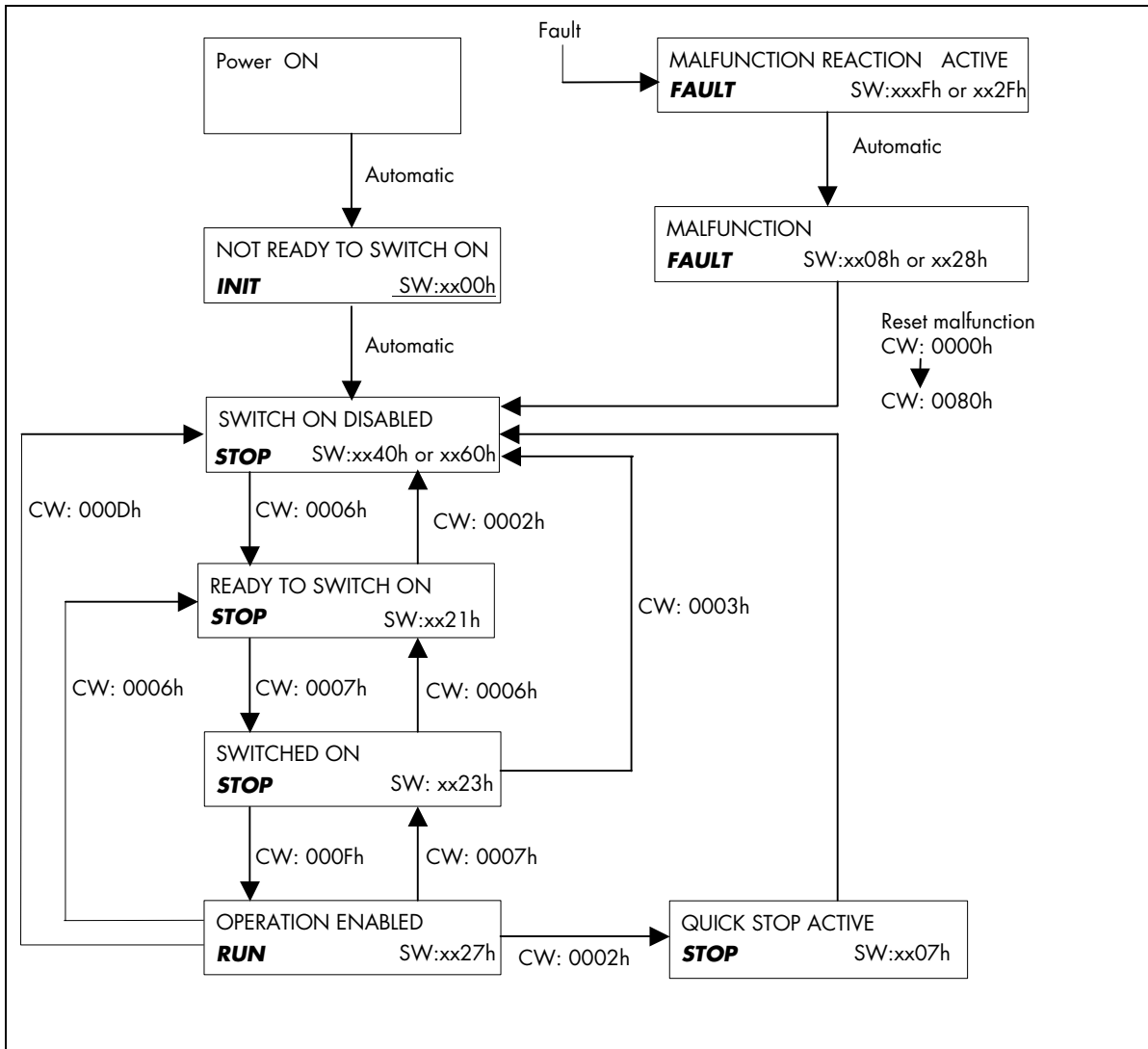
vl_control_effort

The vl_control_effort is the actual speed of the motor.
The unit is RPM.

State Machine

The state machine describes the device status and the possible control sequence of the drive. The state transitions can be generated by using "controlword". The "status word" parameter indicates the current status of state machine. The modes *INIT*, *STOP*, *RUN* and *FAULT* correspond to the actual mode of the Drive.

SW = StatusWord
 CW = ControlWord word



7.6 Using manufacturer specific PDOs

Manufacturer specific PDOs are in use when 'Operate Mode' parameter in panel is set to ByPass. For communication set used in ByPass -mode, see Table 7-2. Manufacturer specific PDOs are TPD021, TPD022, RPD021 and RPD022.

PDO21 Rx

Master to Slave (1)

| Header | | | Data | | | | | | | |
|--------|-----|-----|-------------------------------|---|----------------------------------|---|--------------------------------|---|--------------------------------|---|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x401 | 0 | 8 | <i>nx_control_word</i> | | <i>nx_speed_referense</i> | | <i>process_data_in1</i> | | <i>process_data_in2</i> | |

PDO21 Tx

Slave (1) to Master

| Header | | | Data | | | | | | | |
|--------|-----|-----|------------------------------|---|-------------------------------|---|---------------------------------|---|---------------------------------|---|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x381 | 0 | 8 | <i>nx_status_word</i> | | <i>nx_actual_speed</i> | | <i>process_data_out1</i> | | <i>process_data_out2</i> | |

PDO22 Rx

Master to Slave (1)

| Header | | | Data | | | | | | | |
|--------|-----|-----|--------------------------------|---|--------------------------------|---|--------------------------------|---|--------------------------------|---|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x501 | 0 | 8 | <i>process_data_in3</i> | | <i>process_data_in4</i> | | <i>process_data_in5</i> | | <i>process_data_in6</i> | |

PDO22 Tx

Slave (1) to Master

| Header | | | Data | | | | | | | |
|--------|-----|-----|---------------------------------|---|---------------------------------|---|---------------------------------|---|---------------------------------|---|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x481 | 0 | 8 | <i>process_data_out3</i> | | <i>process_data_out4</i> | | <i>process_data_out5</i> | | <i>process_data_out6</i> | |

Controlling drive

NOTE: some Vacon applications may use "Bypass mode" for an enhanced implementation of "Velocity mode" rather than for a manufacturer specific interface. Please refer to the specific application manual for details

The Reference to the drive can be set also via manufacturer specific Process Data Object 21 (rx) when option board is set to ByPass -mode. In the application, the value is scaled in percentage of frequency area between the set minimum and maximum frequencies.

nx_control_word

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|-----|-----|-----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | - | - | - | - | - | - | - | - | - | - | - | - | RST | DIR | RUN |

In Vacon applications, the three first bits of the control word are used to control the frequency converter. However, you can customise the content of the control word for your own applications because the control word is sent to the frequency converter as such.

| Bit | Description | |
|---------|---|------------------|
| | Value = 0 | Value = 1 |
| 0 | Run | Stop |
| 1 | Clockwise | Counterclockwise |
| 2 | Rising edge of this bit will reset active fault | |
| 3....15 | Not in use | Not in use |

nx_speed_reference

| | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MSB | | | | | | | | | | | | | | | LSB |

This is the Reference 1 to the frequency converter. Used normally as Speed reference. The allowed scaling is -10000...10000. In the application, the value is scaled in percentage of the frequency area between the set minimum and maximum frequencies.

Processdata_in1 ... Processdata_in6

| | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MSB | | | | | | | | | | | | | | | LSB |

These are application specific process data. See APPENDIX C for content of these process data in different applications.

7.7 Drive Monitoring

Several drive actual values/parameters can be monitored by using manufacturer specific PDOs.

nx_speed_reference

| | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MSB | | | | | | | | | | | | | | | LSB |

This is the reference 1 to the frequency converter. Used normally as Speed reference. The allowed scaling is 0..10000. In the application, the value is scaled in percentage of the frequency area between set minimum and maximum frequency.

nx_status_word

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|------|------|--------|----|---|------|---|-----|-----|-----|-----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| - | - | - | - | - | UVFS | DDIR | TCSPDL | FR | Z | AREF | W | FLT | DIR | RUN | RDY |

Information about the status of the device and messages is indicated in the *Status word*. The *Status word* is composed of 16 bits that have the following meanings:

| Bit | Description | |
|---------|--------------------------------------|------------------------------------|
| | Value = 0 | Value = 1 |
| 0 | Not Ready | Ready |
| 1 | STOP | RUN |
| 2 | Clockwise | Counterclockwise |
| 3 | - | Faulted |
| 4 | - | Warning |
| 5 | Ref. frequency not reached | Ref. Frequency reached |
| 6 | - | Motor is running at zero speed |
| 7 | Flux Ready | Flux Not Ready |
| 8 | TC Speed Limit Active | TC Speed Limit Not Active |
| 9 | Detected Encoder Direction Clockwise | Encoder Direction Counterclockwise |
| 10 | UV Fast Stop Active | UV Fast Stop Not Active |
| 11...15 | Not In use | Not In use |

Table 7-4. Status word bit descriptions

Processdata_out1 ... Processdata_out6

| | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| MSB | | | | | | | | | | | | | | | LSB |

These are application specific process data. See APPENDIX C for content of these process data in different applications.

7.8 Anyparameter service

SDO protocol can be used to read any parameter or actual value and write any parameter from drive. These parameters are read from drive with it's ID number specified in user manual. There are three index in object dictionary as follow for anyparameter service.

| Index | Description | Size | Type | Hi 16b | Low 16b |
|-------|-----------------------|------|------|--------|-------------|
| 2000 | AnyparameterReadID | U16 | RW | - | Read ID |
| 2001 | AnyparameterReadValue | U32 | RO | Status | Value |
| 2002 | AnyparameterWrite | U32 | RW | ID | Write Value |

Reading anyparameter

Writing new value to index 2000 will trigger read event, while read is in process index 2001 is zero. Read event will return value to index 2001. If read is succes status will get value of ID and Value is value of ID. If read fails Status will get value 0xFFFF (dec 65535).

Writing anyparameter

When new ID and value is written to index 2002 a write event will be triggered. Index 2002 value will remain as long as writing is processed (normal sdo/pdo operation during this time). If write is succes index 2002 ID and value will be cleared and new write is possible. If write fails ID will clamp to 0xFFFF and value zero.

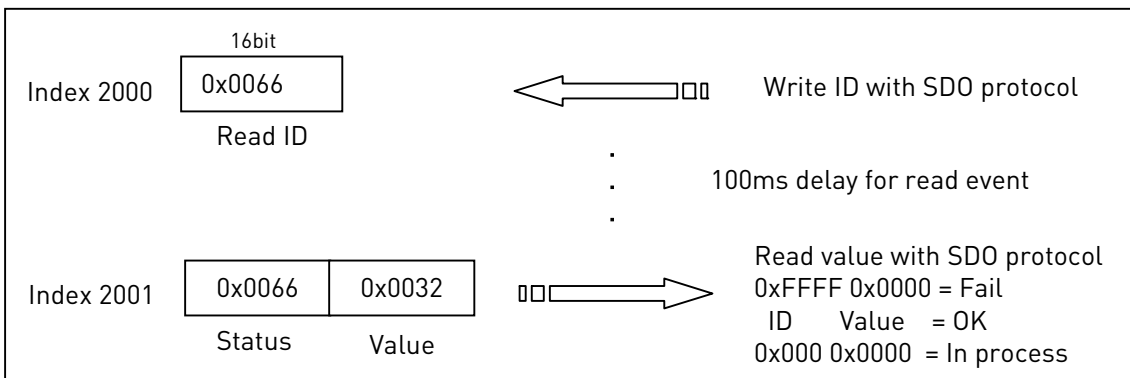


Figure 7-1. Reading Any Parameter

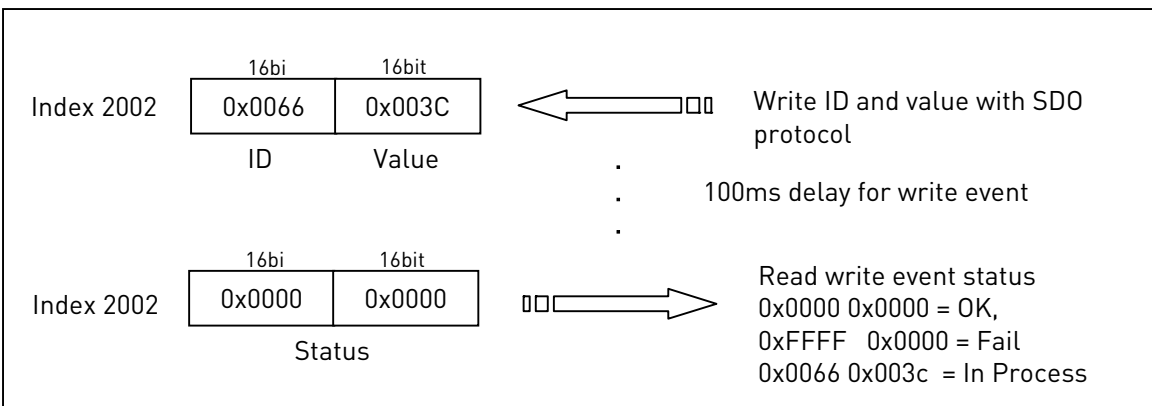


Figure 7-2. Writing Any Parameter

8. SERVICE DATA (SDO)

With service Data Objects (SDOs) the access to entries of a device Object Dictionary is provided. Via SDO all items from object dictionary can be read/written.

Service Data Objects (SDOs) are normally used for device configuration such as setting device parameters. They are also used to define the type and format of information communicated using the Process Data Objects. CanOpen Configuration tools with EDS-files can be used for that purpose. The construction and the method of operation of the SDOs can be found in the CanOpen (DS301) Communication Profile document. Appendix B consist of short description of the SDOs used with Vacon CanOpen option Board.

Object Dictionary

| Index | | Name | Type | Attr. |
|-------|-------|--|---------------------|-----------|
| hex | dec | | | |
| 1000 | 4096 | device_type | <i>Unsigned32</i> | <i>RO</i> |
| 1001 | 4097 | error_register | <i>Unsigned8</i> | <i>RO</i> |
| 1003 | 4099 | pre defined error field | <i>Unsigned32</i> | <i>RO</i> |
| 1005 | 4101 | cob-id sync message | <i>Unsigned32</i> | <i>RO</i> |
| 100C | 4108 | guard_time | <i>Unsigned32</i> | <i>RW</i> |
| 100D | 4109 | life_time_factor | <i>Unsigned32</i> | <i>RW</i> |
| 1018 | 4120 | Identity Object | <i>Identity</i> | |
| 1200 | 4608 | 1 st _server_SDO_parameter | <i>SDOParameter</i> | |
| 1400 | 5120 | 1 st _receive_PDO_parameter | <i>PDOCommPar</i> | |
| 1405 | 5125 | 6 st _receive_PDO_parameter | <i>PDOCommPar</i> | |
| 1414 | 5140 | 21 st _receive_PDO_parameter | <i>PDOCommPar</i> | |
| 1415 | 5141 | 22 st _receive_PDO_parameter | <i>PDOCommPar</i> | |
| 1600 | 5632 | 1 st _receive_PDO_mapping | <i>PDOMapping</i> | |
| 1605 | 5637 | 6 st _receive_PDO_mapping | <i>PDOMapping</i> | |
| 1614 | 5652 | 21 st _receive_PDO_mapping | <i>PDOMapping</i> | |
| 1615 | 5653 | 22 st _receive_PDO_mapping | <i>PDOMapping</i> | |
| 1800 | 6144 | 1 st _transmit_PDO_parameter | <i>PDOCommPar</i> | |
| 1805 | 6149 | 6 nd _transmit_PDO_parameter | <i>PDOCommPar</i> | |
| 1814 | 6164 | 21 nd _transmit_PDO_parameter | <i>PDOCommPar</i> | |
| 1815 | 6165 | 22 nd _transmit_PDO_parameter | <i>PDOCommPar</i> | |
| 1A00 | 6656 | 1 st _transmit_PDO_mapping | <i>PDOMapping</i> | |
| 1A05 | 6661 | 6 st _transmit_PDO_mapping | <i>PDOMapping</i> | |
| 1A14 | 6676 | 21 st _transmit_PDO_mapping | <i>PDOMapping</i> | |
| 1A15 | 6677 | 22 st _transmit_PDO_mapping | <i>PDOMapping</i> | |
| 2000 | 8192 | AnyParameterReadID | <i>Integer16</i> | <i>RW</i> |
| 2001 | 8193 | AnyParameterReadValue | <i>Integer32</i> | <i>RO</i> |
| 2002 | 8194 | AnyParameterWrite | <i>Integer32</i> | <i>RW</i> |
| 2003 | 8195 | nx_current_percentage | <i>Integer16</i> | <i>RO</i> |
| 2004 | 8196 | nx_torque_percentage | <i>Integer16</i> | <i>RO</i> |
| 2063 | 8291 | nx_fault_code | <i>Integer16</i> | <i>RO</i> |
| 27D1 | 10193 | NX control word | <i>Integer16</i> | <i>RW</i> |
| 27D3 | 10195 | NX speed reference | <i>Integer16</i> | <i>RW</i> |
| 27D4 | 10196 | Process data in1 | <i>Integer16</i> | <i>RW</i> |
| 27D5 | 10197 | Process data in2 | <i>Integer16</i> | <i>RW</i> |
| 27D6 | 10198 | Process data in3 | <i>Integer16</i> | <i>RW</i> |
| 27D7 | 10199 | Process data in4 | <i>Integer16</i> | <i>RW</i> |
| 27D8 | 10200 | Process data in5 | <i>Integer16</i> | <i>RW</i> |
| 27D9 | 10201 | Process data in6 | <i>Integer16</i> | <i>RW</i> |
| 2836 | 10294 | NX status word | <i>Integer16</i> | <i>RO</i> |
| 2838 | 10296 | NX actual speed | <i>Integer16</i> | <i>RO</i> |
| 2839 | 10297 | Process data out1 | <i>Integer16</i> | <i>RO</i> |

| | | | | |
|------|-------|----------------------------|-------------------|-----------|
| 283A | 10298 | Process data out2 | <i>Integer16</i> | <i>RO</i> |
| 283B | 10299 | Process data out3 | <i>Integer16</i> | <i>RO</i> |
| 283C | 10300 | Process data out4 | <i>Integer16</i> | <i>RO</i> |
| 283D | 10301 | Process data out5 | <i>Integer16</i> | <i>RO</i> |
| 283E | 10302 | Process data out6 | <i>Integer16</i> | <i>RO</i> |
| 6040 | 24640 | controlword | <i>Unsigned16</i> | <i>RW</i> |
| 6041 | 24641 | statusword | <i>Unsigned16</i> | <i>RO</i> |
| 6042 | 24642 | vl_target_velocity | <i>Integer16</i> | <i>RW</i> |
| 6043 | 24643 | vl_velocity_demand | <i>Integer16</i> | <i>RO</i> |
| 6044 | 24644 | vl_control_effort | <i>Integer16</i> | <i>RO</i> |
| 6046 | 24646 | vl_velocity_min_max_amount | <i>Unsigned32</i> | <i>RW</i> |
| 6048 | 24648 | vl_velocity_acceleration | <i>Ramp</i> | <i>RW</i> |
| 6049 | 24649 | vl_velocity_deceleration | <i>Ramp</i> | <i>RW</i> |
| 604A | 24650 | vl_velocity_quick_stop | <i>Ramp</i> | <i>RW</i> |
| 604E | 24654 | vl_velocity_reference | <i>Unsigned32</i> | <i>RW</i> |
| 6060 | 24672 | modes_of_operation | <i>Integer8</i> | <i>RW</i> |
| 6061 | 24673 | modes_of_operation_display | <i>Integer8</i> | <i>RO</i> |

Description of the Object Dictionary

Abbreviations

| | | | |
|------|--------------|-------|------------------|
| ro | - read only | i16 | - Integer8 |
| wo | - write only | i32 | - Integer8 |
| rw | - read write | u8 | - Unsigned8 |
| co | - constant | u16 | - Unsigned16 |
| bool | - Boolean | u32 | - Unsigned32 |
| i8 | - Integer8 | float | - Floating Point |

| Index (HEX) | Sub-Index | Name | Default Min Max | Type Attr. | Descriptions |
|---------------------------|-----------|-------------------------|--|------------|--|
| General Parameters | | | | | |
| 1000 | 00 | Device Type | 0x00001192 0x00000000 0xFFFFFFFF | u32 co | The device type specifies the kind of device. The lower 16 bits contain the device profile number and the upper 16 bits an additional information. |
| 1001 | 00 | Error Register | 0x00 0x00 0xFF | u8 ro | The error register is a field of 8 bits, each for a certain error type. If an error occurs the bit has to be set. Bit Meaning 0 generic error 1 current 2 voltage 3 temperature 4 communication error (overrun, error state) 5 device profile specific 6 reserved 7 manufacturer specific |
| 1003 | | Pre-defined Error Field | | | This object holds errors that have occurred on the device and have been signalled via Emergency Object. It is an error history. Writing value 0 to sub index 0 deletes the entire error history. |
| | 00 | Number of Errors | 0x00000000 0x00000000 0x0000000A | u8 rw | |
| | 01 | Standard Error Field | 0x00000000 0x00000000 0xFFFFFFFF | u32 ro | |
| | 02 | Standard Error Field | 0x00000000 0x00000000 0xFFFFFFFF | u32 ro | |
| | 03 | Standard Error Field | 0x00000000 0x00000000 0xFFFFFFFF | u32 ro | |
| | 04 | Standard Error Field | 0x00000000 0x00000000 0xFFFFFFFF | u32 ro | |
| | 05 | Standard Error Field | 0x00000000 0x00000000 0xFFFFFFFF | u32 ro | |
| | 06 | Standard Error Field | 0x00000000 0x00000000 0xFFFFFFFF | u32 ro | |
| | 07 | Standard Error Field | 0x00000000 0x00000000 0xFFFFFFFF | u32 ro | |
| | 08 | Standard Error Field | 0x00000000 0x00000000 0xFFFFFFFF | u32 ro | |
| | 09 | Standard Error Field | 0x00000000 0x00000000 0xFFFFFFFF | u32 ro | |

| | | | | | |
|---|----|--|--|--------|--|
| | 0A | Standard Error Field | 0x00000000 0x00000000 0xFFFFFFFF | u32 ro | |
| 1005 | 00 | cob-id sync message | 0x00000080 0x00000000 0xFFFFFFFF | u32 ro | cob-id used for synchronize PDO messages |
| 100C | 00 | Guard Time | 1000 0 65535 | u16 rw | This entry contains the guard time in milliseconds. It is 0, if not used. Unit: ms |
| 100D | 00 | Life Time Factor | 2 0 255 | u8 rw | The life time factor multiplied with the guard time gives the life time for the device. It is 0, if not used. |
| 1018 | | Identity Object | | | This object contains general information about the device. |
| | 00 | Number of entries | 0x4 0x1 0x4 | u8 ro | |
| | 01 | Vendor Id | 0x00000090 0x0 0xFFFFFFFF | u32 ro | Sub-Index 1 contains a unique value allocated each manufacturer. |
| | 02 | Product Code | 0x00000119 0x0 0xFFFFFFFF | u32 ro | Sub-Index 2 identifies the manufacturer specific product code (device version). |
| | 03 | Revision number | 0x00000001 0x0 0xFFFFFFFF | u32 ro | Sub-Index 3 contains the revision number. Bit 31-16 is the major revision number and Bit 15-0 the minor revision number. |
| | 04 | Serial number | 0x0 0x0 0xFFFFFFFF | u32 ro | Sub-Index 4 identified a manufacturer specific serial number. |
| Server SDO Parameters | | | | | |
| 1200 | | Server SDO Parameter | | | The object contains the parameters for the SDOs for which the device is the server. |
| | 00 | Number of Entries | 0x02 0x02 0x02 | u8 ro | |
| | 01 | COB-ID Client -> Server | 0x600+NodeID 0x600 0xFFFF | u32 ro | |
| | 02 | COB-ID Server -> Client | 0x580+NodeID 0x581 0xFFFF | u32 ro | |
| Receive PDO Communication Parameters | | | | | |
| 1400 | | Receive PDO 1 Communication Parameter | | | It contains the communication parameters of the first PDO the device is able to receive. |
| | 00 | Number of Entries | 0x02 0x02 0x05 | U8 ro | Sub-index 0 contains the number of PDO-parameters implemented. |
| | 01 | COB-ID | 0x200+NodeID 0x00000001 0xFFFFFFFF | U32 ro | Sub index 1 describes the COB-ID. If bit 31 is set the PDO is disabled. |
| | 02 | Transmission Type | 0xFF 0x00000000 0x000000FF | u8 ro | The transmission mode is defined by sub-index 2. Value 0xFF (255) = asynchronous transmission mode (=event driven) |
| 1405 | | Receive PDO 6 Communication Parameter | | | It contains the communication parameters of the second PDO the device is able to receive. |
| | 00 | Number of Entries | 0x02 0x02 0x05 | u8 ro | Sub-index 0 contains the number of PDO-parameters implemented. |
| | 01 | COB-ID | 0x300+NodeID 0x00000001 0xFFFFFFFF | u32 ro | Sub index 1 describes the COB-ID. If bit 31 is set the PDO is disabled. |
| | 02 | Transmission Type | 0xFF 0x00 0xFF | u8 ro | The transmission mode is defined by sub-index 2. Value 0xFF (255) = asynchronous transmission mode (=event driven) |
| 1414 | | Receive PDO 21 Communication Parameter | | | It contains the communication parameters of the manufacturer specific PDO21 the device is able to receive. |
| | 00 | Number of Entries | 0x02 0x02 | u8 ro | Sub-index 0 contains the number of PDO-parameters |

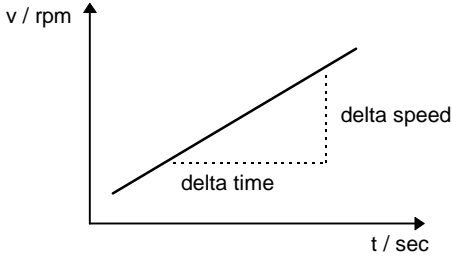
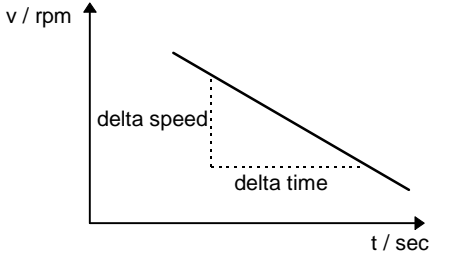
| | | | | | |
|---------------------------------------|----|--|--|--------|---|
| | | | 0x05 | | implemented. |
| | 01 | COB-ID | 0x400+NodeID 0x00000001 0xFFFFFFFF | u32 ro | Sub index 1 describes the COB-ID. If bit 31 is set the PDO is disabled. |
| | 02 | Transmission Type | 0xFF 0x00 0xFF | u8 ro | The transmission mode is defined by sub-index 2. Value 0xFF (255) = asynchronous transmission mode (=event driven) |
| 1415 | | Receive PDO 22 Communication Parameter | | | It contains the communication parameters of the manufacturer specific PDO22 the device is able to receive. |
| | 00 | Number of Entries | 0x02 0x02 0x05 | u8 ro | Sub-index 0 contains the number of PDO-parameters implemented. |
| | 01 | COB-ID | 0x500+NodeID 0x00000001 0xFFFFFFFF | u32 ro | Sub index 1 describes the COB-ID. If bit 31 is set the PDO is disabled. |
| | 02 | Transmission Type | 0xFF 0x00 0xFF | u8 ro | The transmission mode is defined by sub-index 2. Value 0xFF (255) = asynchronous transmission mode (=event driven) |
| Receive PDO Mapping Parameters | | | | | |
| 1600 | | Receive PDO 1 Mapping Parameter | | | It contains the mapping parameters of the first PDO the device is able to receive. Sub-index 0 contains the number of the mapped data objects. All further entries define the data by it's index, sub-index and length. |
| | 00 | Number of Entries | 0x01 0x00 0x40 | u8 ro | |
| | 01 | PDO Mapping Entry | 0x60400010 0x00000000 0xFFFFFFFF | u32 ro | |
| 1605 | | Receive PDO 6 Mapping Parameter | | | It contains the mapping parameters of the PDO6 the device is able to receive. Sub-index 0 contains the number of the mapped data objects. All further entries define the data by it's index, sub-index and length. |
| | 00 | Number of Entries | 0x02 0x0 0x40 | u8 ro | |
| | 01 | PDO Mapping Entry | 0x60400010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 02 | PDO Mapping Entry | 0x60420010 0x00000000 0xFFFFFFFF | u32 ro | |
| 1614 | | Receive PDO 21 Mapping Parameter | | | It contains the mapping parameters of the PDO6 the device is able to receive. Sub-index 0 contains the number of the mapped data objects. All further entries define the data by it's index, sub-index and length. |
| | 00 | Number of Entries | 0x04 0x00 0x40 | u8 ro | |
| | 01 | PDO Mapping Entry | 0x27D10010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 02 | PDO Mapping Entry | 0x27D30010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 03 | PDO Mapping Entry | 0x27D40010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 04 | PDO Mapping Entry | 0x27D50010 0x00000000 0xFFFFFFFF | u32 ro | |
| 1615 | | Receive PDO 22 Mapping Parameter | | | It contains the mapping parameters of the PDO6 the device is able to receive. Sub-index 0 contains the number of the mapped data objects. All further entries define the data by it's index, sub-index and length. |
| | 00 | Number of Entries | 0x04 0x00 0x40 | u8 ro | |

| | | | | | |
|--|----|---|--|--------|---|
| | 01 | PDO Mapping Entry | 0x27D60010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 02 | PDO Mapping Entry | 0x27D70010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 03 | PDO Mapping Entry | 0x27D80010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 04 | PDO Mapping Entry | 0x27D90010 0x00000000 0xFFFFFFFF | u32 ro | |
| Transmit PDO Communication Parameters | | | | | |
| 1800 | | Transmit PDO 1 Communication Parameter | | | It contains the communication parameters of the first PDO the device is able to transmit. |
| | 00 | Number of Entries | 0x04 0x01 0x05 | u8 ro | Sub-index 0 contains the number of PDO-parameters implemented. |
| | 01 | COB-ID | 0x180+NodeID 0x00000001 0xFFFFFFFF | u32 ro | Sub index 1 describes the COB-ID. If bit 31 is set the PDO is disabled. |
| | 02 | Transmission Type | 0xFF 0x00 0xFF | u8 ro | The transmission mode is defined by sub-index 2. Value 0xFF (255) = asynchronous transmission mode (=event driven) |
| | 03 | Inhibit Time | 1000 0x0000 0xFFFF | u16 rw | An inhibit time can be defined on sub-index 3 in 100 µs. This time is minimum interval for PD transmission. Default 100 ms |
| | 04 | CMS Priority Group | 0x03 0x00 0xFF | u8 ro | |
| 1805 | | Transmit PDO 6 Communication Parameter | | | It contains the communication parameters of the PDO 6 the device is able to transmit. |
| | 00 | Number of Entries | 0x05 0x02 0x05 | u8 ro | Sub-index 0 contains the number of PDO-parameters implemented. |
| | 01 | COB-ID | 0x280+NodeID 0x00000001 0xFFFFFFFF | u32 ro | Sub index 1 describes the COB-ID. If bit 31 is set the PDO is disabled. |
| | 02 | Transmission Type | 0xFF 0x00 0xFF | u8 rw | The transmission mode is defined by sub-index 2. Value 0xFF (255) = asynchronous transmission mode (=event driven) |
| | 03 | Inhibit Time | 1000 0x0000 0xFFFF | u16 rw | An inhibit time can be defined on sub-index 3 in 100 µs. This time is minimum interval for PD transmission. Default 100 ms |
| | 04 | CMS Priority Group | 0x03 0x00 0xFF | u8 ro | |
| | 05 | Event Timer | 0x0000 0x0000 0xFFFF | u16 rw | Event Time can be define on subindex 5 in 1ms resolution. This is the time interval PDO will be transmitted. Value 0 = Disable Event Timer |
| 1814 | | Transmit PDO 21 Communication Parameter | | | It contains the communication parameters of the PDO 21 the device is able to transmit. |
| | 00 | Number of Entries | 0x05 0x02 0x05 | u8 ro | Sub-index 0 contains the number of PDO-parameters implemented. |
| | 01 | COB-ID | 0x380+NodeID 0x00000001 0xFFFFFFFF | u32 ro | Sub index 1 describes the COB-ID. If bit 31 is set the PDO is disabled. |
| | 02 | Transmission Type | 0xFF 0x00 0xFF | u8 rw | The transmission mode is defined by sub-index 2. Value 0xFF (255) = asynchronous transmission mode (=event driven) |
| | 03 | Inhibit Time | 1000 0x0000 0xFFFF | u16 rw | An inhibit time can be defined on sub-index 3 in 100 µs. This time is minimum interval for PD transmission. Default 100 ms |
| | 04 | CMS Priority Group | 0x03 | u8 ro | |

| | | | | | |
|--|----|---|--|--------|--|
| | | | 0x00 0xFF | | |
| | 05 | Event Timer | 0x0000 0x0000 0xFFFF | u16 rw | Event Time can be define on subindex 5 in 1ms resolution. This is the time interval PDO will be transmitted. Value 0 = Disable Event Timer |
| 1815 | | Transmit PDO 22 Communication Parameter | | | It contains the communication parameters of the PDO 22 the device is able to transmit. |
| | 00 | Number of Entries | 0x05 0x02 0x05 | u8 ro | Sub-index 0 contains the number of PDO-parameters implemented. |
| | 01 | COB-ID | 0x380+NodeID 0x00000001 0xFFFFFFFF | u32 ro | Sub index 1 describes the COB-ID. If bit 31 is set the PDO is disabled. |
| | 02 | Transmission Type | 0xFF 0x00 0xFF | u8 rw | The transmission mode is defined by sub-index 2. Value 0xFF (255) = asynchronous transmission mode (=event driven) |
| | 03 | Inhibit Time | 1000 0x0000 0xFFFF | u16 rw | An inhibit time can be defined on sub-index 3 in 100 µs. This time is minimum interval for PD transmission. Default 100 ms |
| | 04 | CMS Priority Group | 0x03 0x00 0xFF | u8 ro | |
| | 05 | Event Timer | 0x0000 0x0000 0xFFFF | u16 rw | Event Time can be define on subindex 5 in 1ms resolution. This is the time interval PDO will be transmitted. Value 0 = Disable Event Timer |
| Transmit PDO Mapping Parameters | | | | | |
| 1A00 | | Transmit PDO 1 Mapping Parameter | | | It contains the mapping parameter for the PDOs the device is able to transmit. Sub-index 0 contains the number of the mapped data objects. All further entries define the data by it's index, sub-index and length. The structure of a mapping entry is: index , subindex ,length |
| | 00 | Number of Entries | 0x01 0x00 0x40 | u8 ro | |
| | 01 | PDO Mapping Entry | 0x60410010 0x00000000 0xFFFFFFFF | u32 ro | |
| 1A05 | | Transmit PDO 6 Mapping Parameter | | | It contains the mapping parameter for the PDOs the device is able to transmit. Sub-index 0 contains the number of the mapped data objects. All further entries define the data by it's index, sub-index and length. The structure of a mapping entry is: index, subindex, length |
| | 00 | Number of Entries | 0x02 0x0 0xFF | u8 ro | |
| | 01 | PDO Mapping Entry | 0x60410010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 02 | PDO Mapping Entry | 0x60440010 0x00000000 0xFFFFFFFF | u32 ro | |
| 1A14 | | Transmit PDO 21 Mapping Parameter | | | It contains the mapping parameter for the PDOs the device is able to transmit. Sub-index 0 contains the number of the mapped data objects. All further entries define the data by it's index, sub-index and length. The structure of a mapping entry is: index, subindex, length |
| | 00 | Number of Entries | 0x04 0x00 0xFF | u8 ro | |
| | 01 | PDO Mapping Entry | 0x28360010 0x00000000 0xFFFFFFFF | u32 ro | |

| | | | | | |
|---|----|-----------------------------------|--|--------|--|
| | 02 | PDO Mapping Entry | 0x28380010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 03 | PDO Mapping Entry | 0x28390010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 04 | PDO Mapping Entry | 0x283A0010 0x00000000 0xFFFFFFFF | u32 ro | |
| 1A15 | | Transmit PDO 22 Mapping Parameter | | | It contains the mapping parameter for the PDOs the device is able to transmit. Sub-index 0 contains the number of the mapped data objects. All further entries define the data by it's index, sub-index and length. The structure of a mapping entry is: index,subindex,length |
| | 00 | Number of Entries | 0x04 0x00 0xFF | u8 ro | |
| | 01 | PDO Mapping Entry | 0x283B0010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 02 | PDO Mapping Entry | 0x283C0010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 03 | PDO Mapping Entry | 0x283D0010 0x00000000 0xFFFFFFFF | u32 ro | |
| | 04 | PDO Mapping Entry | 0x283E0010 0x00000000 0xFFFFFFFF | u32 ro | |
| Manufacturer Specific Parameters | | | | | |
| 2000 | | AnyParameterReadID | 0x0000 0x0000 0xFFFF | u16 ro | |
| 2001 | | AnyParameterReadValue | 0x0000 0x0000 0xFFFF | u32 ro | |
| 2002 | | AnyParameterWrite | 0x0000 0x0000 0xFFFF | u32 ro | |
| 2003 | | NX current percentage | 0x0000 0x0000 0xFFFF | u16 ro | Measured motor current. (1 = 0,01A) |
| 2004 | | NX torque percentage | 0x0000 0x0000 0xFFFF | u16 ro | Calculated torque. Scaled in 0,0%...100,0% (0...1000) |
| 2063 | | NX fault code | 0x0000 0x0000 0xFFFF | i16 ro | Shows the drive fault code (=0, if no fault active) |
| 27D1 | | NX control word | 0x0000 0x0000 0xFFFF | i16 ro | |
| 27D3 | | NX speed reference | 0x0000 0x0000 0xFFFF | i16 ro | |
| 27D4 | | Process Data In1 | 0x0000 0x0000 0xFFFF | i16 ro | |
| 27D5 | | Process Data In2 | 0x0000 0x0000 0xFFFF | i16 ro | |
| 27D6 | | Process Data In3 | 0x0000 | i16 ro | |

| | | | | | |
|----------------------------------|----|----------------------------|--|--------|--|
| | | | 0x0000 0xFFFF | | |
| 27D7 | | Process Data In4 | 0x0000 0x0000 0xFFFF | i16 ro | |
| 27D8 | | Process Data In5 | 0x0000 0x0000 0xFFFF | i16 ro | |
| 27D9 | | Process Data In6 | 0x0000 0x0000 0xFFFF | i16 ro | |
| 2836 | | NX status word | 0x0000 0x0000 0xFFFF | i16 ro | |
| 2837 | | NX actual speed | 0x0000 0x0000 0xFFFF | i16 ro | |
| 2839 | | Process data out1 | 0x0000 0x0000 0xFFFF | i16 ro | |
| 283A | | Process data out2 | 0x0000 0x0000 0xFFFF | i16 ro | |
| 283B | | Process data out3 | 0x0000 0x0000 0xFFFF | i16 ro | |
| 283C | | Process data out4 | 0x0000 0x0000 0xFFFF | i16 ro | |
| 283D | | Process data out5 | 0x0000 0x0000 0xFFFF | i16 ro | |
| 283E | | Process data out6 | 0x0000 0x0000 0xFFFF | i16 ro | |
| Device Profile Parameters | | | | | |
| 6040 | | controlword | 0x0000 0x0000 0xFFFF | u16 rw | The control command for the state machine. The state machine describes the device status and possible control sequence of the drive. |
| 6041 | | statusword | 0x0000 0x0000 0xFFFF | u16 ro | The statusword indicates the current status of the drive. |
| 6042 | | vl target velocity | 0x0000 0x8000 0x7FFF | i16 rw | Speed reference of the drive. Unit: RPM |
| 6043 | | vl velocity demand | 0x0000 0x8000 0x7FFF | i16 ro | Speed reference after ramp function. Unit: RPM |
| 6044 | | vl control effort | 0x0000 0x8000 0x7FFF | i16 ro | Actual speed of the motor. Unit: RPM |
| 6046 | | vl velocity min max amount | | | Defines speed limits of the drive in rpm. The parameter consist of a minimum and a maximum speed. |
| | 00 | Number of Entries | 0x02 0x00 0x02 | u32 ro | |
| | 01 | Minimum Speed | 0x000000000x 00000000xFF FFFFFFF | u32 rw | |
| | 02 | Maximum Speed | 0x00000000 0x00000000x FFFFFFF | u32 rw | |

| | | | | | |
|------|----|----------------------------|----------------------------|--------|---|
| 6048 | | vl velocity acceleration | | | <p>This parameter specifies the slope of the acceleration ramp. The parameter consist of two parts: the delta speed and the delta time.</p>  |
| | 00 | Number of Entries | 0x02 0x00 0x02 | u8 ro | |
| | 01 | delta_speed | 0x0 0x0 0xFFFFFFFF | u32 rw | |
| | 02 | delta_time | 0x01 0x0 0xFFFF | u16 rw | |
| 6049 | | vl velocity deceleration | | | <p>This parameter specifies the slope of the deceleration ramp. The parameter consist of two parts: the delta speed and the delta time.</p>  |
| | 00 | Number of Entries | 0x00 0x00 0x02 | u8 ro | |
| | 01 | delta_speed | 0x00 0x0 0xFFFFFFFF | u32 rw | |
| | 02 | delta_time | 0x01 0x0 0xFFFF | u16 rw | |
| 604A | | vl velocity quick stop | | | Not Used |
| | 00 | Number of Entries | 0x2 0x0 0x2 | u8 ro | |
| | 01 | delta_speed | 0x00 0x0 0xFFFFFFFF | u32 rw | |
| | 02 | delta_time | 0x0001 0x0000 0xFFFF | u16 rw | |
| 6060 | | modes of operation | 0x02 0x80 0x7F | i8 ro | This parameter switches the actually chosen operation mode. |
| 6061 | | modes of operation display | 0x02 0x80 0x7F | i8 ro | This parameter shows current mode of operation. |

9. NODE GUARDING PROTOCOL

This protocol is used to detect remote errors in the network. Master can update status of slaves and slaves can guard is master active.

Each NMT slave uses one remote COB for node guarding protocol. The NMT Master polls each NMT slaves at regular time interval. This time interval is called guard time and may be different for each NMT slave. The response of the NMT slave contains the state of that NMT slave. Slave start guarding after one Node Guarding message was received from MNT master. After that if slave (=drive) don't receive any Node Guarding messages during "life time" it will generate "Fieldbus Fault".

Vacon CanOpen option board has following items for guarding purposes:

| Index (HEX) | Sub-Index | Name | Default Min Max | Type Attr. | Descriptions |
|-------------|-----------|------------------|--------------------|------------|---|
| 100C | 00 | Guard Time | 1000 0 65535 | u16 rw | This entry contains the guard time in milli-seconds. It is 0, if not used. Unit: ms |
| 100D | 00 | Life Time Factor | 2 0 255 | u8 rw | The life time factor multiplied with the guard time gives the life time for the device. It is 0, if not used. |

Default "life time" is 1000ms (Guard Time) * 2 (Life Time Factor) = 2 s

Example node quarding messages:

MASTER -> SLAVE (node id = 1)

| Message | ID | Len | RTR |
|---------------------|-------|-----|-----|
| NMT (Node Guarding) | 0x701 | 0 | 1 |

SLAVE RESPONSE (node id = 1)

| Message | ID | Len | 1 |
|---------------------|-------|-----|---|
| NMT (Node Guarding) | 0x701 | 1 | X |

X = bit7 0 / 1 (Toggle bit)
 bit6...0 Slave state (esim. 5 operational)

10. ELECTRONIC DATA SHEET, EDS-FILE

The usage of devices in a communication network requires configuration of the device parameters and communication facilities. CANopen defines a standardised way to access these parameters via the object dictionary.

For handling of the complexity of CANopen systems Software Tools are required. This reduces the complexity of the planning, configuration and analysis process and significantly increases the security of the system.

For this purpose Software Tools need an electronic description of the CANopen devices. To allow the usage of manufacturer independent Tools, this document defines a standardised file format – called Electronic Data Sheet EDS.

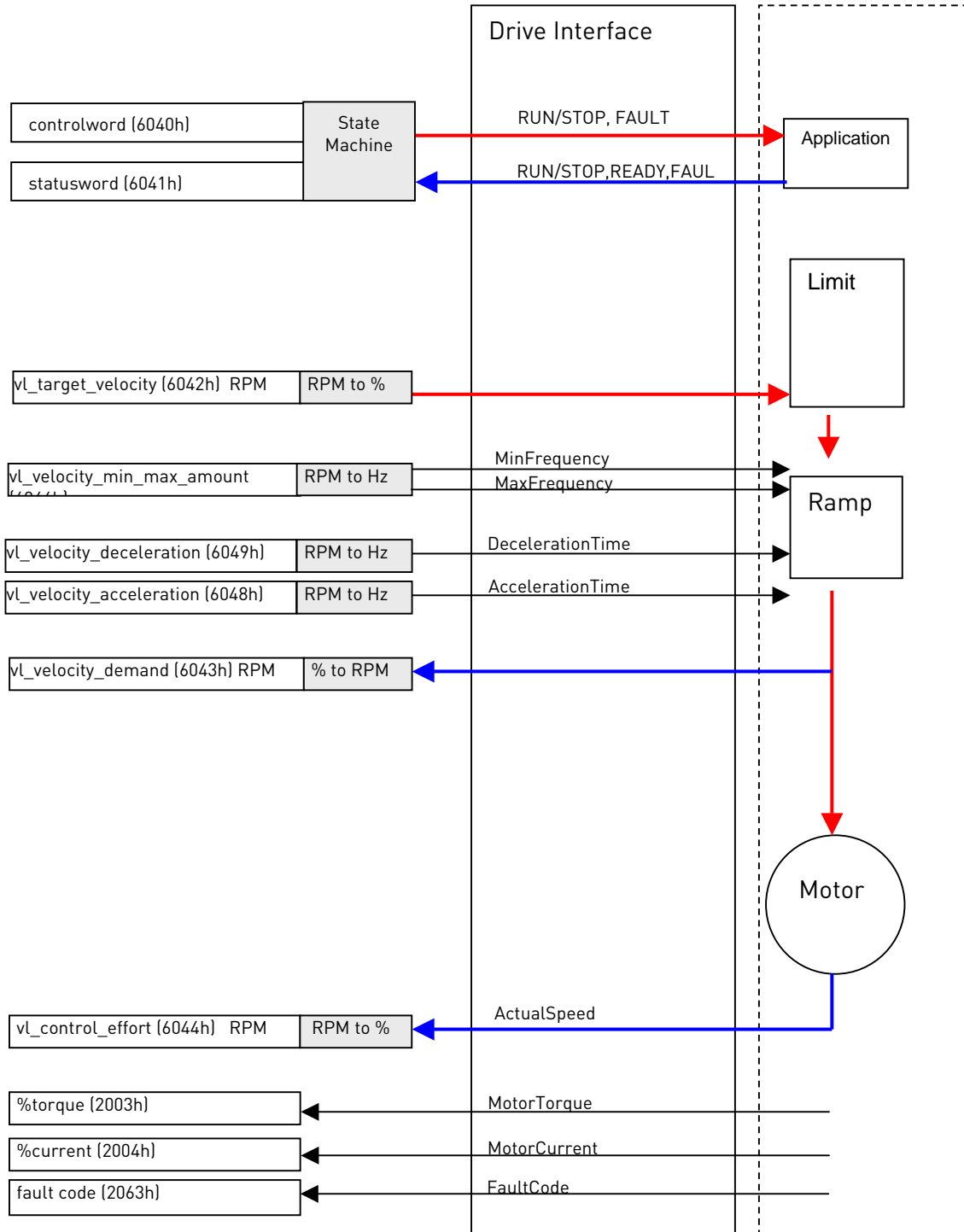
The EDS-file for Vacon CanOpen Option Board can be found from [vacon web-site](http://vacon.com)

11. APPENDICES

APPENDIX A - DEVICE PROFILE FOR DRIVES

Vacon CanOpen option board follows the Drive device profile DSP-402. The Velocity mode is supported.

Basic Device Control and Device Data Interface



APPENDIX B – SDO MESSAGES

Read Dictionary Object: SD01 Rx

Master to Slave (1)

| Header | | | Data | | | | | | | |
|--------|-----|-----|---------|--------------|---|------------|----|----|----|----|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x601 | 0 | 4 | Command | Object Index | | Sub- Index | D0 | D1 | D2 | D3 |

Command = 0x40
 Command aaabbbbb (= 0100 0000)
 aaa = 010, Initiate upload request

Response: SD01 Tx

Slave (1) to Master

| Header | | | Data | | | | | | | |
|--------|-----|-----|--------------|---------------------|---|-------------------|----|----|----|----|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x581 | 0 | 4 | <i>Reply</i> | <i>Object Index</i> | | <i>Sub- Index</i> | D0 | D1 | D2 | D3 |

Reply = e.g. 0x43
 Reply aaabcces (= 0100 0011)
 aaa = 010 Initiate upload response
 b = 0 not used
 cc = 00 number of empty bytes (that case answer consist of 4 bytes)
 e = 1 expedited transfer
 s = 1 data set size is indicated
 Data D0...D4

Write Dictionary Object: SD01 Rx

Master to Slave (1)

| Header | | | Data | | | | | | | |
|--------|-----|-----|----------------|---------------------|---|-------------------|----|----|----|----|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x601 | 0 | 4 | <i>Command</i> | <i>Object Index</i> | | <i>Sub- Index</i> | D0 | D1 | D2 | D3 |

Command = e.g. 0x2B
 Command aaabcces (= 0010 1011)
 aaa = 010 Initiate download request
 b = 0 not used
 cc = 00 number of empty bytes (that case answer consist of 4 bytes)
 e = 1 expedited transfer
 s = 1 data set size is indicated
 Data D0...D4

Response: SD01 Tx

Slave (1) to Master

| Header | | | Data | | | | | | | |
|--------|-----|-----|--------------|---------------------|---|-------------------|----|----|----|----|
| ID | RTR | Len | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0x581 | 0 | 4 | <i>Reply</i> | <i>Object Index</i> | | <i>Sub- Index</i> | D0 | D1 | D2 | D3 |

Reply = 0x60
 Reply aaabbbbb (= 0110 0000)
 aaa = 010 Initiate download response
 bbbbb = 0 0000 not used

APPENDIX C – PROCESS DATA CONTENTS

Process Data OUT (Slave → Master)

The fieldbus master can read the frequency converter's actual values using process data variables. *Basic, Standard, Local/Remote, Multi-Step, PID control and Pump and fan control* applications use process data as follows:

| Data | Value | Unit | Scale |
|--------------------|------------------|------|---------|
| Process data OUT 1 | Output Frequency | Hz | 0,01 Hz |
| Process data OUT 2 | Motor Speed | rpm | 1 rpm |
| Process data OUT 3 | Motor Current | A | 0,1 A |
| Process data OUT 4 | Motor Torque | % | 0,1 % |
| Process data OUT 5 | Motor Power | % | 0,1 % |
| Process data OUT 6 | Motor Voltage | V | 0,1 V |

The *Multipurpose* application has a selector parameter for every Process Data. The monitoring values and drive parameters can be selected using the ID number (see NX All in One Application Manual, Tables for monitoring values and parameters). Default selections are as in the table above.

Process Data IN (Master -> Slave)

ControlWord, Reference and Process Data are used with All-in One applications as follows:

Basic, Standard, Local/Remote, Multi-Step applications

| Data | Value | Unit | Scale |
|-------------|---|------|-------|
| Reference | Speed Reference | % | 0.01% |
| ControlWord | Start/Stop Command Fault reset Command | - | - |
| PD1 – PD6 | Not used | - | - |

Multipurpose control application

| Data | Value | Unit | Scale |
|------------------|---|------|-------|
| Reference | Speed Reference | % | 0.01% |
| ControlWord | Start/Stop Command Fault reset Command | - | - |
| Process Data IN1 | Torque Reference | % | 0.1% |
| Process Data IN2 | Free Analog INPUT | % | 0.01% |
| PD3 – PD6 | Not Used | - | - |

PID control and Pump and fan control applications

| Data | Value | Unit | Scale |
|------------------|---|------|-------|
| Reference | Speed Reference | % | 0.01% |
| ControlWord | Start/Stop Command Fault reset Command | - | - |
| Process Data IN1 | Reference for PID controller | % | 0.01% |
| Process Data IN2 | Actual Value 1 to PID controller | % | 0.01% |
| Process Data IN3 | Actual Value 2 to PID controller | % | 0.01% |
| PD4–PD6 | Not Used | - | - |

APPENDIX D

Version 8 of the C6 Option board doesn't work the same way than the others. Some changes are required at end of the PLC.

HEAD OFFICE AND PRODUCTION:**Vaasa**

Vacon Plc
 Rursorintie 7
 65380 Vaasa
 firstname.lastname@vacon.com
 telephone: +358 (0)201 2121
 fax: +358 (0)201 212 205

PRODUCTION:**Suzhou, China**

Vacon Suzhou Drives Co. Ltd.
 Building 13CD
 428 Xinglong Street
 Suchun Industrial Square
 Suzhou 215126
 telephone: +86 512 6283 6630
 fax: +86 512 6283 6618

Vacon Traction Oy

Vehnämyllynkatu 18
 33580 Tampere
 telephone: +358 (0)201 2121
 fax: +358 (0)201 212 710

SALES COMPANIES AND REPRESENTATIVE OFFICES:**FINLAND****Helsinki**

Vacon Plc
 Äyritie 12
 01510 Vantaa
 telephone: +358 (0)201 212 600
 fax: +358 (0)201 212 699

Tampere

Vacon Plc
 Vehnämyllynkatu 18
 33580 Tampere
 telephone: +358 (0)201 2121
 fax: +358 (0)201 212 750

AUSTRIA

Vacon AT Antriebssysteme GmbH
 Aumühlweg 21
 2544 Leobersdorf
 telephone: +43 2256 651 66
 fax: +43 2256 651 66 66

BELGIUM

Vacon Benelux NV/SA
 Interleuvenlaan 62
 3001 Heverlee (Leuven)
 telephone: +32 (0)16 394 825
 fax: +32 (0)16 394 827

CHINA

Vacon Suzhou Drives Co. Ltd.
 Beijing Office
 A205, Grand Pacific Garden Mansion
 8A Guanhua Road
 Beijing 100026
 telephone: +86 10 6581 3734
 fax: +86 10 6581 3754

FRANCE

Vacon France
 ZAC du Fresne
 1 Rue Jacquard – BP72
 91280 Saint Pierre du Perray CDIS
 telephone: +33 (0)1 69 89 60 30
 fax: +33 (0)1 69 89 60 40

GERMANY

Vacon GmbH
 Gladbecker Strasse 425
 45329 Essen
 telephone: +49 (0)201 806 700
 fax: +49 (0)201 806 7099

INDIA

Vacon India
 Flat no T1, 3rd floor
 VNS Ashok Apartment
 Plot no. 9A, New Beach Road
 Thiruvanniyur
 Chennai-600041
 Tel. +91 44 245 150 18

ITALY

Vacon S.p.A.
 Via F.lli Guerra, 35
 42100 Reggio Emilia
 telephone: +39 0522 276811
 fax: +39 0522 276890

THE NETHERLANDS

Vacon Benelux BV
 Weide 40
 4206 CJ Gorinchem
 telephone: +31 (0)183 642 970
 fax: +31 (0)183 642 971

NORWAY

Vacon AS
 Langgata 2
 3080 Holmestrand
 telephone: +47 330 96120
 fax: +47 330 96130

RUSSIA

ZAO Vacon Drives
 Bolshaja Jakimanka 31,
 109180 Moscow
 telephone: +7 (095) 974 14 47
 fax: +7 (095) 974 15 54

ZAO Vacon Drives

2ya Sovetskaya 7, office 210A
 191036 St. Petersburg
 telephone: +7 (812) 332 1114
 fax: +7 (812) 279 9053

SPAIN

Vacon Drives Ibérica S.A.
 Miquel Servet, 2. P.I. Bufalvent
 08243 Manresa
 telephone: +34 93 877 45 06
 fax: +34 93 877 00 09

SWEDEN

Vacon AB
 Torget 1
 172 67 Sundbyberg
 telephone: +46 (0)8 293 055
 fax: +46 (0)8 290 755

THAILAND

Vacon South East Asia
 335/32 5th-6th floor
 Srinakarin Road, Prawet
 Bangkok 10250
 Tel. +66 (0)85 100 7090

UNITED ARAB EMIRATES

Vacon Middle East and Africa
 Block A, Office 4A 226
 P.O.Box 54763
 Dubai Airport Free Zone
 Dubai
 Tel. +971 (0)4 204 5200
 Fax: +971 (0)4 204 5203

UNITED KINGDOM

Vacon Drives (UK) Ltd.
 18, Maizefield
 Hinckley Fields Industrial Estate
 Hinckley
 LE10 1YF Leicestershire
 telephone: +44 (0)1455 611 515
 fax: +44 (0)1455 611 517

Vacon distributor:

ООО "Вакон-Бизнес-Центр"
 199034, СПб, В.О., 17 линия, д. 4-6
 e-mail: vacon@vacon.spb.ru
 тел.: (812)3260952, факс: (812)3260951