

**MOVIDRIVE[®] CANopen Fieldbus Interface
DFO11A**

Edition

10/2000

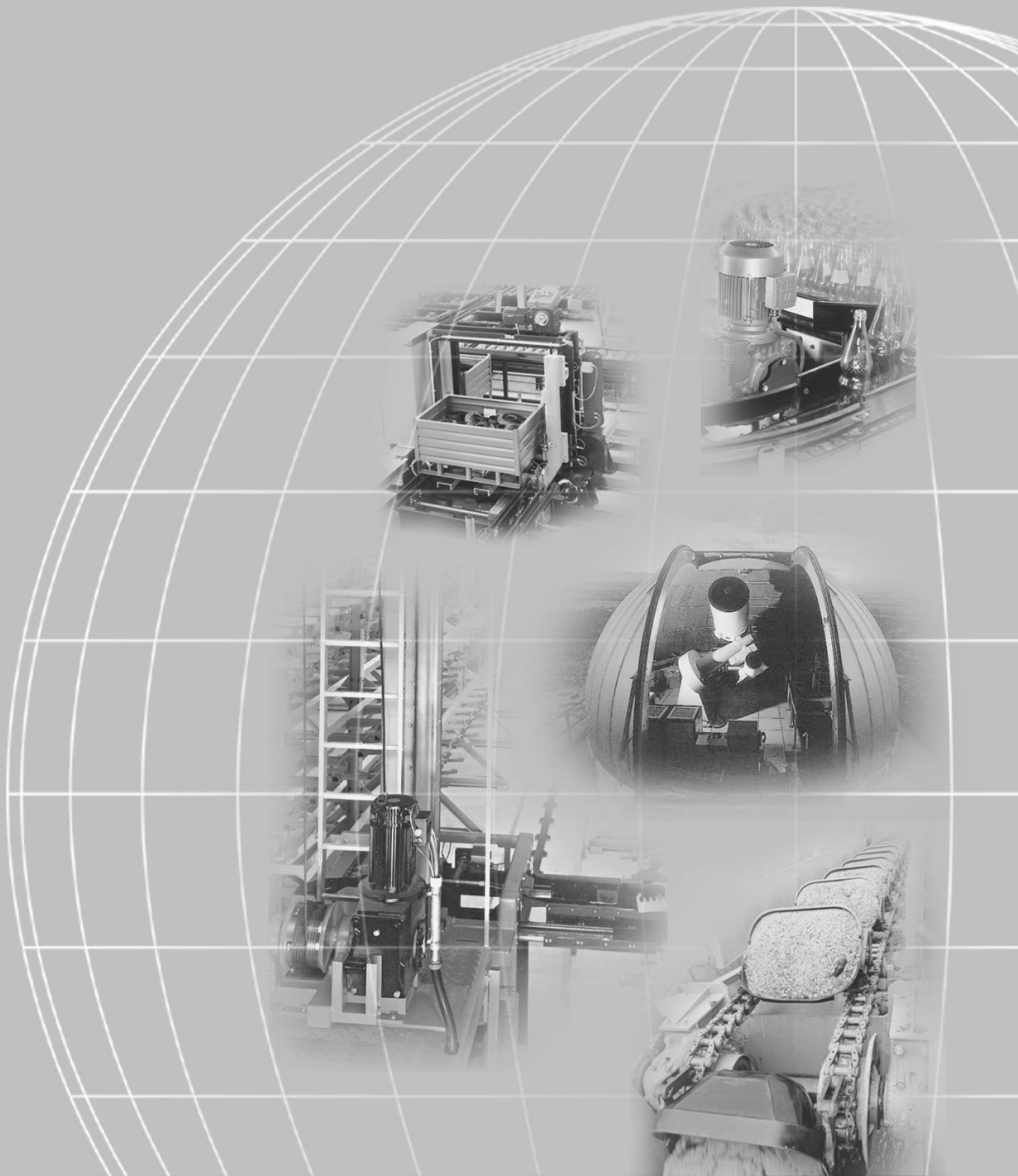


CANopen

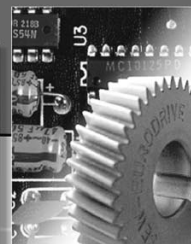


Manual

09187111 / EN



SEW-EURODRIVE





Contents

1 Introduction	4
2 Assembly/Installation	6
2.1 Types of Units Supported	6
2.2 Installation of Option PCB	6
2.3 Connector Wiring and Connector Pin Assignment.....	6
2.4 Shielding and Routing of Bus Cables.....	7
2.5 Setting the DIP Switches	8
2.6 Display Elements	10
3 Project Planning/Startup	11
3.1 Data Exchange with CANopen.....	11
3.2 Startup	18
4 Application Example	27
4.1 Positioning with Five Process Data.....	27
5 Appendix	31
6 Index	33



1 Introduction

Contents of this manual

This manual pertains to the CANopen fieldbus interface DFO11A and describes the installation of the CANopen option pcb DFO11A into the drive inverter and the startup of MOVIDRIVE[®] on the CANopen fieldbus .

It also contains an explanation of all settings for the fieldbus option and the different types of connections with CANopen in the form of small startup examples.

Documentation

For simple and effective connection of MOVIDRIVE[®] to the CANopen fieldbus system, the following documentation should be used in addition to this manual:

- MOVIDRIVE[®] Fieldbus Unit Profile Manual
- MOVIDRIVE[®] System Manual

The manual accompanying MOVIDRIVE[®] Fieldbus Unit Profile features a description of fieldbus parameters and related coding. It also features an explanation of the various control schemes and possible applications in form of small examples.

The MOVIDRIVE[®] system manual contains a list of all parameters of the drive inverter that can be read and written via various communication interfaces such as RS-485 and even via the fieldbus interface.

MOVIDRIVE[®] and CANopen

The MOVIDRIVE[®] drive inverter together with the DFO11A option and its high-performance universal fieldbus interface allows connection to higher-level automation systems via CANopen.

Device profile

The underlying behavior of the inverter (also referred to as the unit profile) that forms the basis for CANopen operation, is fieldbus independent and, therefore, uniform. This allows the user to develop fieldbus-independent drive applications. As a result, changing over to other bus systems such as INTERBUS (option DFI11A), PROFIBUS (option DFP11A), CAN bus (DFC11A) or DeviceNet (DFD11A) is easily achieved.

Drive parameters

Use of the MOVIDRIVE[®] CANopen interface offers direct access to all drive parameters and functions. Control of the drive inverter is achieved through the fast, cyclical process data. Using this process data channel, the user can specify not only setpoint values (e.g., setpoint speed, integrator time for acceleration/deceleration, etc.), but also initiate various drive functions such as release, controller inhibit, normal stop, quick stop, etc. At the same time, this channel can also be used to read back actual values such as actual speed, current, unit status, fault code or even reference signals from the drive inverter.

PDO/SDO

Whereas the process data exchange is mapped to the CANopen PDOs¹, parameterization of the inverter is mapped solely via SDOs². This parameter data exchange allows implementation of applications in which all important drive parameters are stored in the higher-level automation system. As a result, it is not necessary to set parameters manually in the drive inverter.

1. Process Data Object
2. Service Data Object



Startup

The CANopen option is designed so that the fieldbus-specific settings for the Slave ID and Baud Rate can be adjusted by means of a hardware switch on the option pcb. Using this manual setting, the drive inverter can be quickly integrated into the CANopen environment and switched on. The parameters can be set automatically by the higher-level CANopen master (parameter download). This future-oriented system offers the advantages of reducing systems startup time and simplifying the documentation of the application program, since all important drive parameters can be stored directly in the control program.

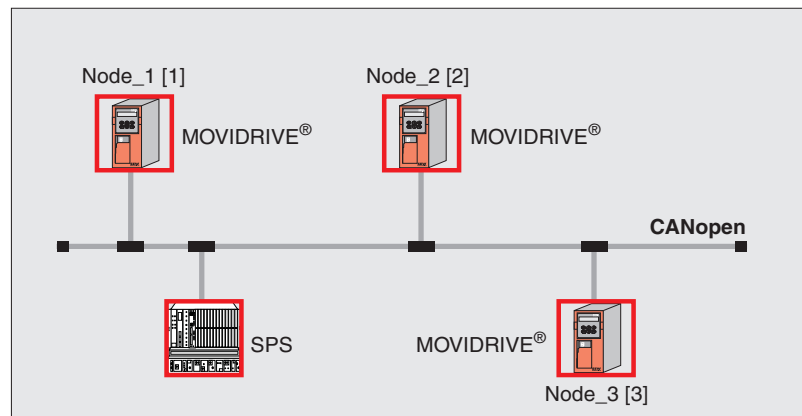


Figure 1: CANopen with MOVIDRIVE® and PLC

03678AXX

Monitoring functions

The use of a fieldbus system requires additional drive system monitoring such as time-monitoring of the fieldbuses (fieldbus timeout) or even quick stop concepts. The MOVIDRIVE® monitoring functions can be customized to your applications allowing you to determine which fault response the drive inverter will trigger in the case of a bus error. A quick stop is useful for many applications, but you can also freeze the last setpoint values so that the drive continuously runs using the last valid setpoint values (e.g., a conveyor belt). Since the functionality of the control terminals is ensured in fieldbus operation, the user can still implement fieldbus-independent quick stop schemes via the terminals of the drive inverter.

Diagnostics

The MOVIDRIVE® drive inverter offers numerous diagnostic options for startup and service. Using the integrated fieldbus monitor, users can control setpoint values sent from the higher-level controller as well as the actual values.

Users also receive additional information on the status of the fieldbus option pcb. The fieldbus monitor function offers a convenient diagnostic option that allows for a detailed display of fieldbus and unit status information in addition to the settings of all drive parameters (including fieldbus parameters).



2 Assembly/Installation

2.1 Types of units supported

The DFO11A option for connection to CANopen can be used with all drive inverters in the MOVIDRIVE[®] series.

2.2 Installation of option pcb

Before you begin

- Take appropriate ESD measures (anti-static band, conductive shoes, etc.) before touching the option pcb.
- Keep option pcb in its original packaging and remove it only when you are ready to install it.
- Avoid frequent touching of the option pcb and hold the pcb only at the edges. Do not touch any components.

Installation of option pcb

- Disconnect the supply voltage of the inverter. Switch off the supply voltage and the 24 V supply, if necessary.
- Remove lower cover from control unit.
- Unscrew electronics shield terminal.
- Remove black cover plate.
- Insert option pcb into guide rails of slot OPTION1 and push it into the slot.
- Apply moderate pressure to the front plate of the option pcb while pushing it into the slot. The pcb is seated correctly when it is level with the controller pcb.
- Replace and fasten electronics shield terminal.
- Replace cover on the control unit.
- The installation of DFO11A option pcb is now complete.

2.3 Connector wiring and connector pin assignment

Connector wiring

The DFO11A option contains a 9-pin Sub D connector. The CAN bus is connected via the 9-pin male Sub D connector. Since the spur line must be kept short, T-connectors may not be used. Instead, the bus lines must be directly looped through the connector (see Figure 2).

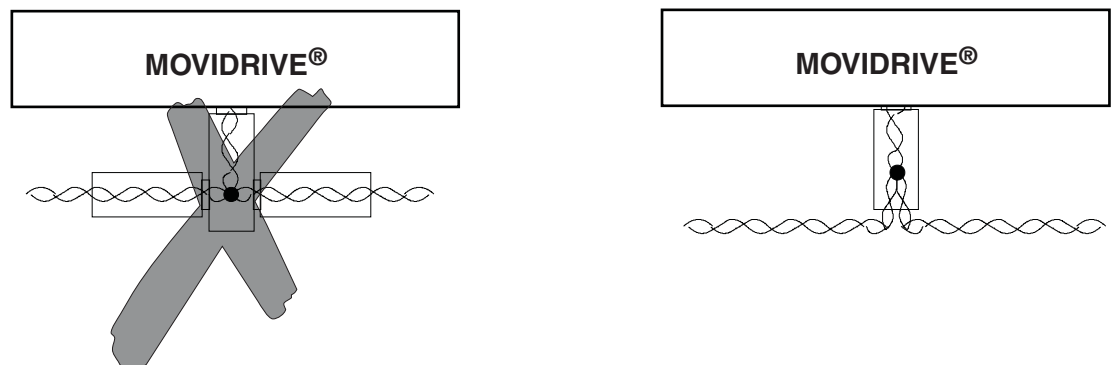


Figure 2: Cabling of CAN bus (top view)

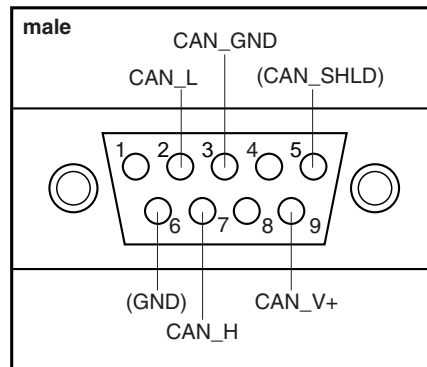
03679AXX



Connector pin assignment

Figure 3 shows the pin assignment of the 9-pin Sub-D coupling. This connector pin assignment corresponds to the CiA (CAN in Automation) recommendations. In order to avoid EMC problems, connectors with metallic connector housings or metallized housings must be used to connect the inverter to the CAN bus.

The assignment of terminals is described in "CANopen Specification CIA DR-303-1".



03680AXX

Figure 3: Connector pin assignment

2.4 Shielding and routing of bus cables

Professional shielding of the bus cables attenuates electrical interference that may occur in industrial environments.

Measures for good shielding properties

- Tighten the mounting screws on the connectors, modules, and potential-compensation cables by hand.
- Do not route signal and bus cables parallel to line cables (motor leads). They must be routed in separate cable conduits.
- Use metallic, grounded cable racks in industrial environments.
- Route the signal cable and the corresponding potential compensation in close proximity using the shortest way possible.
- Avoid extending bus lines by means of connectors.
- Route the bus cables closely alongside existing grounding areas.
- Use bus connectors with metallized or metallic housing.



CAUTION!

If ground potential fluctuations are present, a compensating current may flow via the shield that is bilaterally connected and connected with ground potential (PE). In this case, ensure that sufficient compensation potential is present in accordance with relevant VDE regulations.

Bus termination

In order to avoid disruptions in the bus system due to reflections, etc., every CANopen segment at the physically first and last participant must be terminated using 120 Ω bus terminating resistors. Optionally, a bus terminating resistor with a DIP switch can be connected to the circuit for a DFO11A.



2.5 Setting the DIP Switches

The CANopen DFO11A pcb has 16 DIP switches.

One DIP switch is used to connect the terminating resistor, two DIP switches are used to set the baud rate, and 7 DIP switches are used to set the CANopen slave ID (node address). An additional three switches set the number of process data words (Figure 4).

The default settings of the pcb at delivery are: Baud rate = 500 kbps, Module ID = 1, Number of process data = 2.



03766AXX

Figure 4: DIP switch

Designation	Meaning	Switch
PD(0) ... PD(2)	Number of process data	S1-0 ... S1-2
NA(0) ... NA(6)	Module ID	S1-3 ... S1-9
DR(0) ... DR(1)	Baud rate	S1-10 ... S1-11
nc	No function	S2-0 ... S2-2
R	Terminating resistor	S2-3



CAN bus terminating resistor

DIP switch "R" can be used to connect a terminating resistor for the CAN bus to the system. This terminating resistor must be switched on at the first and the last device on the CAN bus and must be switched off at all other devices.

CAN bus baud rate

DIP switches DR0 and DR1 are used to adjust the baud rate to the CAN bus. The baud rate is set using the following table:

Baud rate	DR0	DR1
125 kbps	off	off
250 kbps	on	off
500 kbps	off	on
1 Mbps	on	on

CANopen address

The DIP switches NA0 ... NA5 can be used to select a CANopen address in the range 1 ... 127.

The address is calculated as follows:

$$\text{Address} = \text{NA0} + 2 \cdot \text{NA1} + 4 \cdot \text{NA2} + 8 \cdot \text{NA3} + 16 \cdot \text{NA4} + 32 \cdot \text{NA5} + 64 \cdot \text{NA6}$$



Caution: The CANopen address 0 (all NA DIP switches set to off) is invalid!

Number of process data words

The DIP switches PD0 through PD2 determine the number of process data words to be transmitted (possible numbers are 1 through 8). Generally, the number of process input data words (PI) and process output data words (PO) is identical. Since a PDO can transmit no more than 8 bytes, a maximum of 4 process data words can be transferred with one PDO. If more than 4 process data words are to be transferred, a second PDO channel (PDO2) is used). However, the SEW MOVILINK® profile provides for only 3 process data words. The remaining 5 process data words can be used exclusively by IPOS programs.

The number of process data words, the resulting process data channels (PDOs), and the position of the DIP switches are determined using the following table:

Number of process data words (PI and PO)	Number of IPOS process data words	Total number of process data words	Number of PDOs	PD0	PD1	PD2
1	0	1	1	off	off	off
2	0	2	1	on	off	off
3	0	3	1	off	on	off
3	1	4	1	on	on	off
3	2	5	2	off	off	on
3	3	6	2	on	off	on
3	4	7	2	off	on	on
3	5	8	2	on	on	on



2.6 Display elements

The display elements consist of 4 two-colored LEDs.

Function	Abbreviation
CAN bus status	BUSOFF
CANopen status	STATE
Node guarding	GUARD
Send or receive	COMM

BUSOFF LED

Status	LED	Meaning
NO ERROR	Off	The number of bus errors is within the normal range (Error-Active-State).
BUS WARNING	Flashes red (approx. 1 s cycle)	The number of physical bus errors is too high. No more error messages are actively written to the bus (Error-Passive-State).
OVERFLOW	Flashes red (approx. 250 ms cycle)	The cycle time on the bus is too small. Therefore, the CAN controller has rejected at least one message.
	Red	BusOFF state The number of physical bus errors has continued to grow despite a switch to the Error-Passive-State. Access to the bus is deactivated.
CONFIG. ERROR	Red, the other three LEDs flash green (approx. 1 s cycle)	The invalid value 0 was entered as node ID via the DIP switches. However, operation requires that the node ID be set higher than 0 (see Section [Assembly/Installation]) / [Setting of the DIP Switches] / [CANopen Address]).

STATE LED

Status	LED	Meaning
Pre-operational	Flashes green	See Section [Project Planning/Startup] / [Data Exchange with CANopen] / [Unit Status and NMT Services]
Operational	Green	
Stopped	Off	

GUARD LED

Status	LED	Meaning
Node guarding Timeout	Flashes green	See Section [Project Planning/Startup] / [Data Exchange with CANopen] / [Node Guarding]
Node guarding active and ok	Green	
Node guarding not active	Off	

COMM LED

LED	Meaning
Flashes green	Message received for this node
Flashes green	Message sent



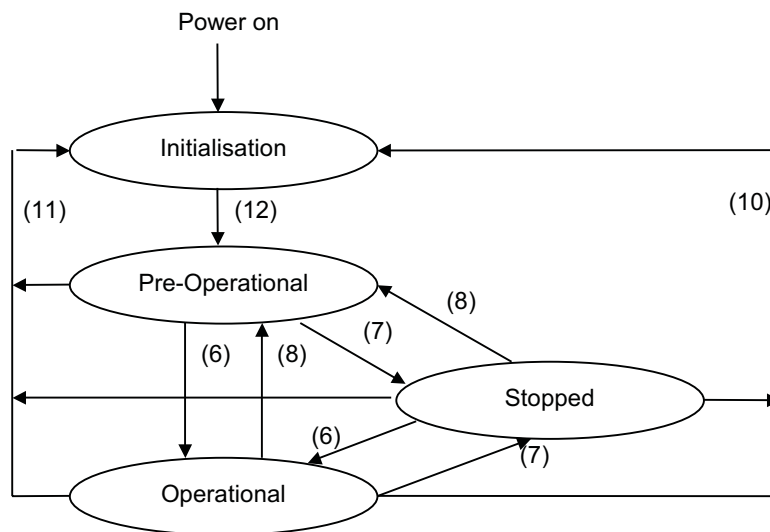
3 Project Planning/Startup

3.1 Data exchange with CANopen

Unit status and NMT services

The CANopen option pcb supports the "minimum capability device", that is, it supports "pre-operational", "operational," and "stopped" statuses. In the "pre-operational" status, the unit can communicate via SDOs only (see Section [Project Planning/Startup] / [Data Exchange with CANopen] / [SDOs and Their Services]). In "operational" status, PDOs and SDOs can be exchanged, and in "stopped" status, neither SDOs nor PDOs can be exchanged.

After power on, the CANopen option pcb is automatically in "pre-operational" status.



03686AXX

Figure 5: Status diagram for CANopen devices

Changing status

The status can be changed at any time using NMT¹ services.

The following commands can be used:

- (6) Node_Start indication
- (7) Node_Stop indication
- (8) Enter_Pre-Operational_State indication
- (10) Reset_Node indication

This command resets the complete inverter, and the CANopen option pcb sets its default settings in the object directory.

- (11) Reset_Communication

This command initiates a reset of communication parameters in the object directory.

- (12) "Initialisation" finished – changes automatically to "Pre-Operational."

1. Network Management



Structure of CAN messages

CAN messages are structured as follows:

NMT service	ID	Byte 1	Byte 2
Node_Start	0x0000	0x01	Node ID
Node_Stop	0x0000	0x02	Node ID
Enter_Pre-Operational_State	0x0000	0x80	Node ID
Reset_Node	0x0000	0x81	Node ID
Reset_Communication	0x0000	0x82	Node ID

Node ID corresponds to the address set with the DIP switches (see *Assembly/Installation / Setting of the DIP Switches*). In addition, the 0 value can be used for the node ID; in this case, all CANopen devices are addressed.

NMT services are not confirmed by the slave.

SDOs and their services

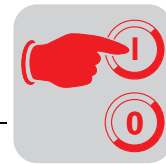
SDOs (Service Data Objects) are used to set parameters and read out the configuration of the inverter. SDOs are capable of loading large data amounts into the inverter, but are relatively slow. In order to send SDOs from or to the inverter, the following services can be applied to SDOs:

- Multiplexed Download Domain (write access)
 - Initiate Domain Download Protocol
 - Expedited Transfer
 - Normal Transfer (net data length > 4 bytes)
- Multiplexed Upload Domain (read access)
 - Initiate Domain Upload Protocol
 - Expedited Transfer
 - Normal Transfer (net data length > 4 bytes)
- Abort Domain Transfer (if protocol errors occur)

The *Normal Transfer* is implemented only for uploads of objects 0x1008 - 0x100A, all others are transferred using *Expedited Transfer* during upload since their maximum length measures 4 bytes.

The response time for download or upload is not specified; among other things, it is dependent upon the system load. That is: the bus load and the number of incoming PDOs.

The parameters, their indexes and subindexes can be found in Section [Project Planning/Startup] / [Data Exchange with CANopen] / [Object Directory].



Example: Read access "Device Type"

The following example shows a read access to the entry "Device Type" in the inverter with the CANopen address = 3 as set by the DIP switches:

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x600+Node ID=0x603	0x40	0x00	0x10	0x00	0x00	0x00	0x00	0x00
ID for SDO	Expedited Upload	Index low	Index high	Subindex	Value has no effect			

For read access, the controller must send the protocol "Initiate Multiplexed Upload Domain."

This protocol ensures that CAN messages are always 8 bytes long.

The option pcb sends the following as an answer:

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x580+Node ID=0x583	0x43 ¹⁾	0x00	0x10	0x00	0x2D	0x01	0x00	0x00
ID for SDO	Expedited Upload	Index low	Index high	Subindex	Response, low	Response	Response	Response, high

1) This value applies only to the listed message. This value may change for other indexes depending on data type; see CANopen DS301.

Example: Write access

Analogous to this, a write access to index 0x100C, subindex 0x00 (Guard Time). This entry is set to 10000 ms (0x2710).

First, the controller sends an "Expedited Download" request.

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x600+Node ID=0x603	0x23 ¹⁾	0x0C	0x10	0x00	0x10	0x27	0x00	0x00
ID for SDO	Expedited Download	Index low	Index high	Subindex	Value, low	Value, high	Filler byte	Filler byte

1) This value applies only to the listed message. This value may change for other indexes depending on data type; see CANopen DS301.

Upon successful execution (the CANopen option pcb checks index, subindex, write permission, data type and, if necessary, whether the transmitted value is permitted), the CANopen option pcb responds.

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x580+Node ID=0x583	0x60 ¹⁾	0x00	0x10	0x00	0x00	0x00	0x00	0x00
ID for SDO	Expedited Download	Index low	Index high	Subindex	Value has no effect			

1) This value applies only to the listed message. This value may change for other indexes depending on data type; see CANopen DS301.



Faulty service operation

If an error occurs, the DFO11A transmits an Abort message. The Abort message contains an error code that describes the cause of the error.

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x580+Node ID=0x583	0x80	0x00	0x10	0x00				
ID for SDO	SDO abort	Index low	Index high	Subindex	Add. code low	Add. code high	Error code	Error class

PDOs

CANopen allows the quick exchange of process data between controller and inverter by means of PDOs (Process Data Objects). In order to enable a CANopen unit to exchange PDOs, the unit must have been set to the "operational" status via NMT commands (Section [Project Planning/Startup] / [Data Exchange with CANopen] / [Unit Status and NMT Services]). The process data (optionally 1, 2, or 3) of the MOVILINK[®] protocol can be exchanged with the PDOs. If the 3 process data are not sufficient for control, up to 5 additional IPOS control words can be transmitted for further processing by an IPOS program in the inverter. The corresponding mapping is determined automatically by the setting of the DIP switches (Section [Assembly/Installation] / [Setting of the DIP Switches]).

The transmission rate of a PDO, however, must always be kept at or below 1 ms per PDO in order to ensure the transfer of all data.

RX-PDOs

The RX-PDO is defined as the PDO in the CANopen profile that was sent from the controller to the inverter. The process data are accepted asynchronously by the CANopen option pcb.

RX-PDOs correspond to the following scheme:

RX-PDO1:

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x200+Node ID =0x203	PO1 (low)	PO1 (high)	PO2 (low)	PO2 (high)	PO3 (low)	PO3 (high)	IPOS-PO1 (low)	IPOS-PO1 (high)

RX-PDO2:

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x300+Node ID =0x303	IPOS-PO2 (low)	IPOS-PO2 (high)	IPOS-PO3 (low)	IPOS-PO3 (high)	IPOS-PO4 (low)	IPOS-PO4 (high)	IPOS-PO5 (low)	IPOS-PO5 (high)

The length and number of RX-PDOs can be set by adjusting the setting of the DIP switches (Section [Assembly/Startup] / [Setting of the DIP Switches]).

**TX-PDO**

TX-PDOs are transmitted by the inverter. The TX-PDO is transmitted only upon reception of an RX-PDO or reception of an RTR-TX-PDO. An RTR-TX-PDO causes the inverter to place its process input data on the CAN bus without requiring new process output data to the inverter. For this purpose, the RTR bit must be set in the CAN message of the TX-PDO.

TX-PDOs are structured similar to RX-PDOs, but they contain process input data words (PI) instead of process output data words (PO). Like RX-PDOs, the length and number of TX-PDOs can be set by adjusting the setting of the DIP switches (Section [Assembly/Startup] / [Setting of the DIP Switches]).

TX-PDO1:

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x180+Node ID =0x183	PO1 (low)	PO1 (high)	PO2 (low)	PO2 (high)	PO3 (low)	PO3 (high)	IPOS-PO1 (low)	IPOS-PO1 (high)

TX-PDO2:

ID	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
0x280+Node ID =0x283	IPOS-PO2 (low)	IPOS-PO2 (high)	IPOS-PO3 (low)	IPOS-PO3 (high)	IPOS-PO4 (low)	IPOS-PO4 (high)	IPOS-PO5 (low)	IPOS-PO5 (high)

PDO mapping

The CANopen option pcb supports only one default mapping because of speed compatibility. Using this type of mapping, the option pcb is capable of sending PDOs with the shortest possible latency.

COB-IDs

Object	Default COB-ID
TX-PDO1	0x180 + Node ID
TX-PDO2	0x280 + Node ID
RX-PDO1	0x200 + Node ID
RX-PDO2	0x300 + Node ID
TX-SDO	0x580 + Node ID
RX-SDO	0x600 + Node ID
Node guarding	0x700 + Node ID
NMT	0x0000

**Node-guarding**

The CANopen option pcb supports two types of node guarding.

For one, the network master can check whether the individual nodes are still operational. To accomplish this, a node guarding object with set RTR bit must be sent to the nodes (example for node ID "3"):

ID	RTR	Message length
0x700+Node ID= 0x703	Yes	0 bytes

At standby, the node responds with a corresponding node guarding object that returns the current operating status (see Section [Project Planning/Startup] / [Data Exchange with CANopen] / [Unit Status and NMT Services]) and a toggle bit.

ID	RTR	Byte 1
0x700+Node ID= 0x703	No	Bit 7: Toggle bit Bit 0 ... 6: Operating status Operating status: operational = 5 pre-operational = 127 stopped = 4

The toggle bit changes with every message between 0 and 1. Using the response, the network master can determine whether the CANopen participants are still in their original status or whether the status has changed due to an error. For example: the CANopen option pcb always changes to the "pre-operational" state if a reset of the inverter occurs.

Lifeguarding

In the second type of node-guarding, the CANopen slaves check their NMT master. For this purpose, a timeout period in milliseconds can be set with indexes 0x100C ("guard time") and 0x100D ("life time factor"). This timeout period is calculated by multiplying life time factor * guard time. Timeout periods smaller than 10 ms are rejected by the CANopen option pcb.

The second type of node guarding is active only if a timeout period is set to a value other than 0 (that is, life time factor \neq 0 and guard time \neq 0). If the master does not trigger a "Node Event" within the timeout period, the inverter responds using the fieldbus timeout programmed response (P831).

Exception: The timeout response (P831) is programmed with "No Response" or "Display Error," and timeout monitoring is activated via CANopen – which is not appropriate.

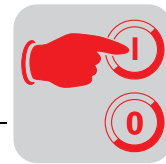
Activation of node guarding is signalled by continuous illumination of GUARD-LED.

Caution!

The timeout period set by the controller can be read with P819 via MOVITOOLS, but it may not be changed using MOVITOOLS; instead, it may only be changed by the controller via CANopen objects 0x100C ("guard time") and 0x100D ("life time factor").



For example, if a timeout error occurs due to wire breakage, the RX setpoint values in the inverter are set to zero as a precaution. This prevents an immediate restart of the machine in case the timeout response P831 was programmed only to warn and a node-guarding object is received after a timeout while the controller has not yet sent new process output data.

**Node Event**

Node guarding is active in all operating states following the first reception of a "Node Event" from the master.

With this CANopen option pcb, the Node Event can be chosen freely from the following events:

- reception of an RTR Guard message from the master
- reception of an RX-PDO1
- reception of an RX-PDO2

Node Event can be programmed with object index 0x2010, subindex 0, whereby the object (an UNSIGNED32) is structured as follows:

Bit 2	Bit 1	Bit 0
Every RX PDO 2 represents a Node Event	Every RX PDO 1 represents a Node Event	Every incoming RTR Guard message represents a Node Event

For the time being, bits 3 through 31 are not defined and must always be 0.

The default value following a CANopen reset is 1; that is, only RTR Guard messages are accepted as Node Event.

Sync object

The optional Sync Object presented by CANopen is not analyzed by the CANopen option pcb.

Error detection

Errors in the inverter can be detected by checking the index 0x1001 (Error Register) or 0x1002 (Manufacturer Status Register). But in order to keep the bus load as low as possible, it is practical to place the status word after a process input data word. This allows for quick error detection without creating an additional bus load. Additional signalling options, especially the emergency object, were not implemented.

"Error History" includes a maximum of the last five errors. These errors can be queried via the index area of the inverter or in accordance with CANopen profile via object 0x1003. If a "0" is written into this object at subindex 0, the "Error History" will be deleted.

Object directory

All CANopen objects relevant for DFO11A are entered in the CANopen object directory. The object directory is divided into two areas:

- Communication-specific profile area (index 0x1000 - 0x1BFF)
- Manufacturer-specific profile area (index 0x2000 - 0x5FFF)

All communication-specific objects and some objects especially relevant for DFO11A are listed in their entirety in the appendix.

The manufacturer-specific objects of the MOVIDRIVE[®] inverter are described in the fieldbus unit profile.

However, CANopen allows only Read and Write services to the manufacturer-specific objects via SDO. If the SEW-specific services of the MOVILINK[®] fieldbus unit profile should be used (e.g., read minimum, read maximum, read default, write volatile, ...), this can be accomplished using a detour via objects 0x2066 and 0x2067. Object 0x2067 (a SIGNED32) contains the data intended to list the next MOVILINK[®] or the result of the last MOVILINK[®] service if it was successful.



Writing to object 0x2066 triggers the MOVILINK[®] service.

Objekt 0x2066 (UNSIGNED32) is structured as follows:

Bit 31 - Bit 24	Bit 23 - Bit 16	Bit 15 - Bit 8	Bit 7 - Bit 0
Management	Reserved	Index high	Index low

The exact meaning of the management byte is described in the fieldbus device profile, Section [Structure of the MOVILINK[®] Parameter Channel].

If an error occurs, the MOVILINK[®] error codes are returned with Abort SDO.

Example 1

Determine the possible maximum (service "Read Maximum") for index 0x2116 (ramp on right) from the inverter:

The management byte must accept the value 0x35, the reserved byte is set to 0.

Therefore, object 0x2066 is written with the value 0x35002116 via an SDO.

The possible maximum can then be read from object 0x2067 via an SDO.

Example 2

Write index 0x2116 (ramp on right) volatile with value 0x1234 (service "write volatile"):

Before this service is carried out, the MOVILINK[®] service data (index 0x2067) must be set to value 0x1234. This is accomplished by a writing SDO to index 0x2067.

Then carry out the MOVILINK[®] service by writing the value 0x33002116 to index 0x2066.

Residential Storing of Operating Parameters in No-Voltage Condition

The communications parameters with indexes in the range 0x1000 ... 0x2000 are not saved when power is shut down. After power-up, the default values are reloaded from ROM. All other parameters are saved even in the no-voltage condition, except if they were changed using the "write volatile" command.

3.2 Startup

This section describes the startup of the MOVIDRIVE[®] inverter with the DFO11A option by means of the operator software or the DBG11A keypad, Version .13 or higher.



- Inverter control mode fieldbus*** After installing the CANopen option pcb and setting the baud rate, node ID, and number of process data (via DIP switches), the MOVIDRIVE® inverter can be parameterized without further settings via the fieldbus system. For example, after power-up, all drive parameters from the higher-level programmable controller can be downloaded directly using CANopen.
- Control mode*** Before controlling the inverter via CANopen, it must be switched to the corresponding control mode. This can be achieved with parameter P100 (index 8461) setpoint source and P101 (index 8462) control source. Based on factory settings, these parameters are set to the values bipolar/fixe setpoint or terminals (setpoint processing via analog setpoint value and control via input terminals).
- Setpoint source*** With the setting P100 setpoint source = FIELDBUS, the inverter takes over the setpoint values from the fieldbus; and with P101 = FIELDBUS, the inverter is controlled via the fieldbus (that is, enable, quick stop, normal stop, controller inhibit, etc.). The MOVIDRIVE® inverter now responds to the process output data sent from the higher-level programmable controller.
- The control source "Fieldbus" (P101) is signalled to the higher-level controller using the bit "fieldbus mode active" in the status word.
- Safety*** For safety reasons, the inverter must also be enabled on the terminal side for control via the fieldbus system. Subsequently, the terminals must be connected or programmed so that the inverter is enabled via the input terminals. The simplest way of enabling the inverter at the terminal side consists of connecting the input terminal DI00 (Function/CONTROLLER INHIBIT) with a +24 V signal and programming the input terminals DI01 through DI05 to WITHOUT FUNCTION.



Procedure for startup

Procedure for startup of the inverter **MOVIDRIVE®** with fieldbus interface:

1. Switch drive inverter to ENABLE on the terminal side.

Connect input terminal DI00 (Function/CONTROLLER INHIBIT) to a +24 V signal (via wire jumper).

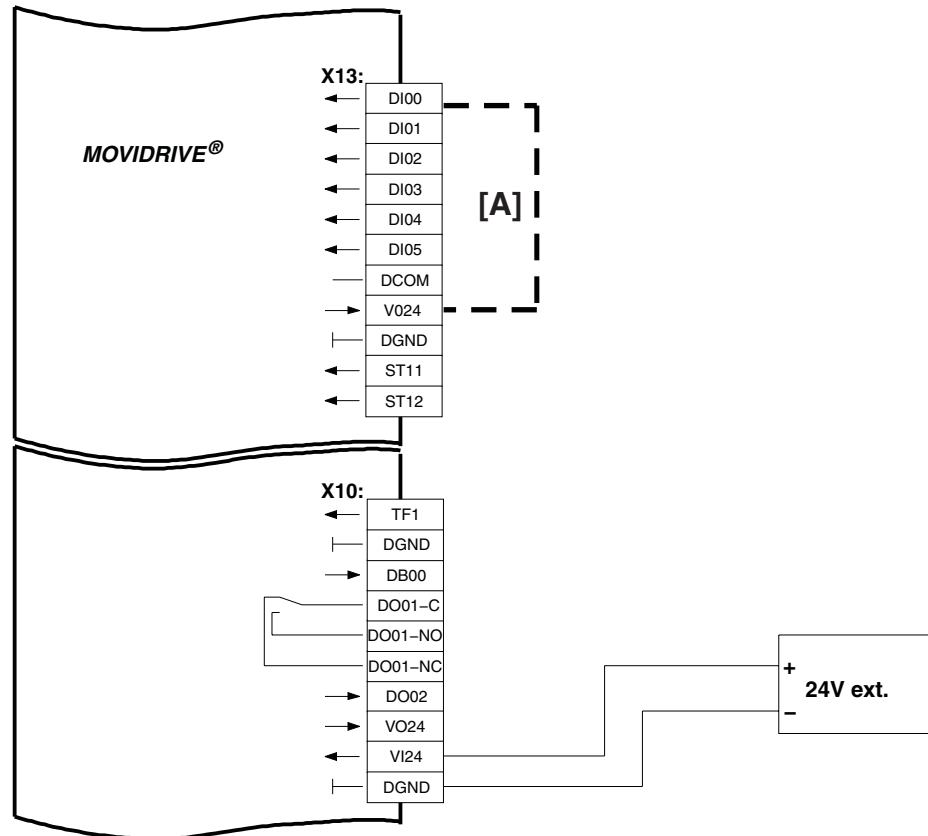


Figure 6: Connection for enable

03692BXX

[A] The drive inverter can be enabled on the terminal side via this wire jumper!

X13	DI00: /Controller inhibit	X10	TF1: TF input
	DI01 ... X13:DI05: No function		DGND: Reference potential binary signals
	DCOM: Reference DI00 ... DI05		DB00: /Brake
	VO24: + 24 V		DO01-C: Relay contact
	DGND: Reference potential binary signals		DO01-NO: Relay NO contact
	ST11: RS-485 +		DO01-NC: Relay NC contact
	ST12: RS-485 -		DO02: /Malfunction
			VO24: + 24 V
			VI24: + 24 V (external)



2. Control mode = FIELDBUS

Switch over control and setpoint value processing of the inverter with parameter P100 and P101 to FIELDBUS.

3. Input terminal DI01... DI05 = WITHOUT FUNCTION

Program the functionality of input terminals DI01 through DI05 with parameters P600 through P604 to WITHOUT FUNCTION.

**Installation of
EDS file**

Eight EDS files, **MDX1.eds** ... **MDX8.eds**, are available for the DFO11A option pcb. Depending on the number of set process data words, the corresponding file must be installed using the CANopen manager software.

Example

Three process data words are selected at the DIP switches PD0 ... PD2. The MDX3.eds file must then be loaded using the CANopen manager software.



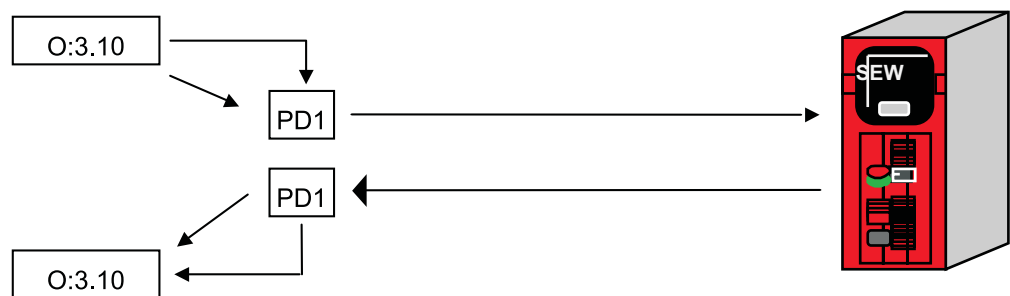
Project planning for one process data word

The DIP switches for the number of process data words of DFO11A must be set in accordance with Section [Assembly/Installation] / [Setting of the DIP Switches] / [Number of Process Data] to PD0=off, PD1=off, PD2=off.

This setting is displayed in the operating software or the DBG via parameter "PD Configuration" = 1 PD or 1PD + Param. This causes one process data word (2 bytes) to be processed in the inverter and transmission of one process input data word to the controller.

If the controller sends more than one process output data word, only the first process data is processed or sent back. With less than one process data, the PDO message is ignored.

If the controller sends one process output data word, it is processed and one process input data word is sent from the inverter to the controller.



03765AXX

Figure 7: Illustration of the process data word in the PLC memory area

The output file of the PLC contains the process output data and the input file contains the process input data of the PLC. In the above example, the output data word O:3.10 is copied onto the process output data word 1 and processed by the inverter. The inverter returns one process input data word that is copied into the input data word I:3.10 of the PLC.



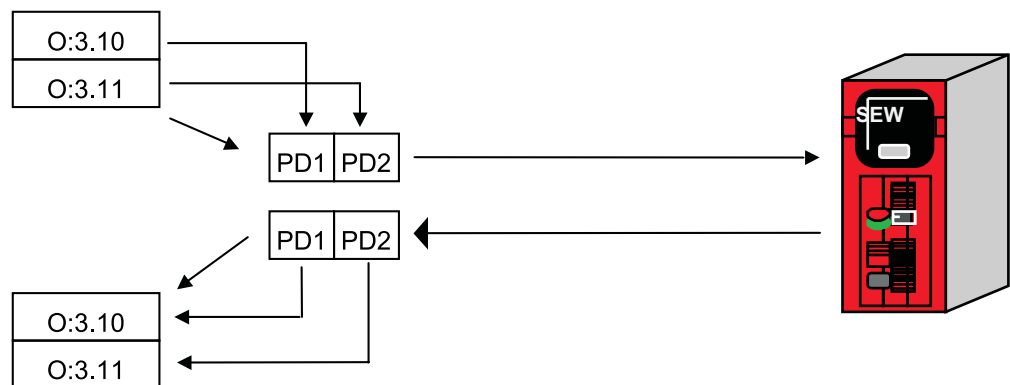
Project planning for two process data words

The DIP switches for the number of process data words of DFO11A must be set in accordance with Section [Assembly/Installation] / [Setting of the DIP Switches] / [Number of Process Data] to PD0=on, PD1=off, PD2=off.

This setting is displayed in the operating software or the DBG via parameter "PD Configuration" = 2 PD or 2PD + Param. This causes the processing of two process data words (4 bytes) in the inverter and transmission of two process input data words to the controller.

If the controller sends more than two process output data words only the first two process data are processed or sent back. With fewer than two process data, the PDO message is ignored.

If the controller sends two process output data words, two process output data words are processed and two process input data words are sent from the inverter to the controller.



03764AXX

Figure 8: Illustration of the 2 process data words in the PLC memory area

The output file of the PLC contains the process output data and the input file contains the process input data of the PLC. In the above example, the output data words O:3.10 and O:3.11 are copied onto the process output data words 1 and 2 and processed by the inverter. The inverter returns two process input data words that are copied into the input data words I:3.10 and I:3.11 of the PLC.



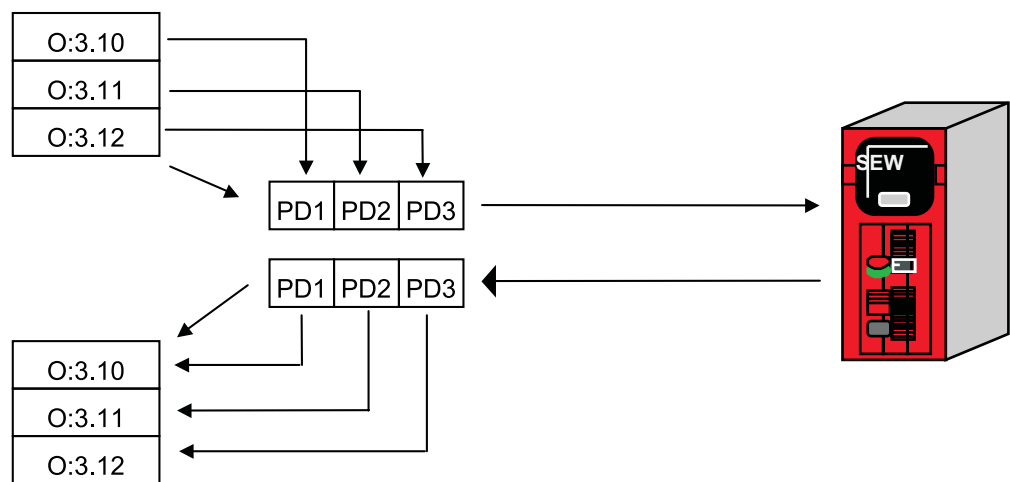
Project planning for three process data words

The DIP switches for the number of process data words of DFO11A must be set in accordance with Section [Assembly/Installation] / [Setting of the DIP Switches] / [Number of Process Data] to PD0=off, PD1=on, PD2=off.

This setting is displayed in the operating software or the DBG via parameter "PD Configuration" = 3 PD or 3PD + Param. This causes the processing of three process data words (6 bytes) in the inverter and transmission of three process input data words to the controller.

If the controller sends more than three process output data words only the first three process data are processed or sent back. With fewer than three process data, the PDO message is ignored.

If the controller sends three process output data words, three process output data words are processed and three process input data words are sent from the inverter to the controller.



03763AXX

Figure 9: Illustration of the 3 process data words in the PLC memory area

The output file of the PLC contains the process output data and the input file contains the process input data of the PLC. In the above example, the output data words O:3.10, O:3.11 and O:3.12 are copied onto the process output data words 1, 2 and 3 and processed by the inverter. The inverter returns three process input data words that are copied into the input data words I:3.10, I:3.11 and I:3.12 of the PLC.

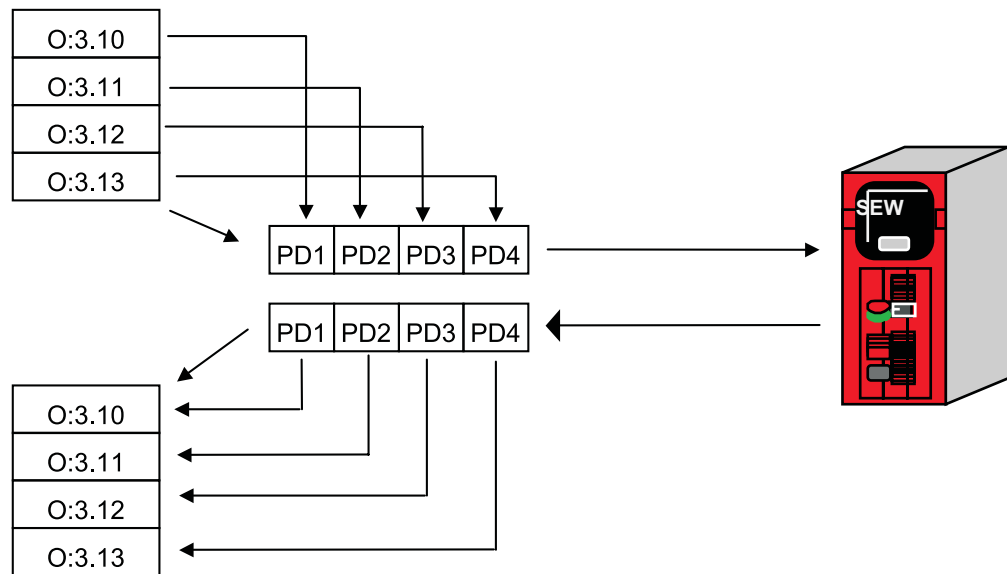


Project planning for four process data words

The DIP switches for the number of process data words of DFO11A must be set in accordance with Section [Assembly/Installation] / [Setting of the DIP Switches] / [Number of Process Data] to PD0=on, PD1=on, PD2=off.

This setting is displayed in the operating software or the DBG via parameter "PD Configuration" = 4 PD or 4PD + Param.

A project planning with more than three process data words goes beyond the standard fieldbus profile so that the additional process data word 4 can be processed only by an IPOS program running in the inverter. The process output words are fetched with the IPOS command `_GetSys()`, the process input words are written with the command `_SetSys()`.



03688AXX

Figure 10: Illustration of the process data word in the PLC memory area

The output file of the PLC contains the process output data and the input file contains the process input data of the PLC. In the above example, the output data words O:3.10 and O:3.13 are copied onto the process output data words 1 and 4 and processed by the inverter. The inverter returns four process input data words that are copied into the input data words I:3.10 and I:3.13 of the PLC.



Project planning for five process data words

The DIP switches for the number of process data words of DFO11A must be set in accordance with Section [Assembly/Installation] / [Setting of the DIP Switches] / [Number of Process Data] to PD0=off, PD1=off, PD2=on.

This setting is displayed in the operating software or the DBG via parameter "PD Configuration" = 5 PD or 5PD + Param. Since the maximum length of a PDO is four words, the PDO2 object is used to transmit the fifth word.

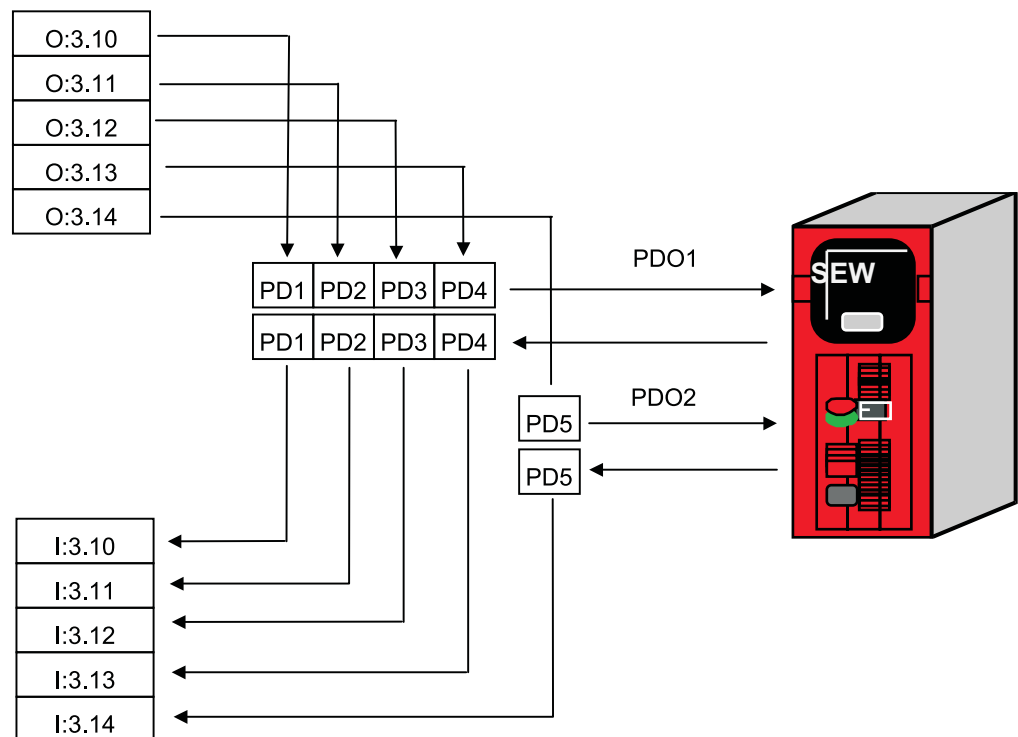


Caution!

Since the process data are transmitted using two different CANopen objects, it cannot be assumed that the transmitted process data are consistent with each other. For example, if a 32 bit precise position is to be transmitted with process data, it is absolutely necessary to ensure that the low and high word of the position are transmitted in one PDO.

Project planning with more than three process data words goes beyond the standard fieldbus profile so that the additional process data words 4 and 5 can be processed only by an IPOS program running in the inverter. The process output words are fetched with the IPOS command `_GetSys()`, the process input words are written with the command `_SetSys()`.

Section 4 shows an application example with IPOS program for five process data words.



03689AXX

Figure 11: Illustration of the process data word in the PLC memory area

The output file of the PLC contains the process output data and the input file contains the process input data of the PLC. In the above example, the output data words O:3.10 and O:3.14 are copied onto the process output data words 1 and 5 and processed by the inverter. The inverter returns five process input data words that are copied into the input data words I:3.10 and I:3.14 of the PLC.



4 Application Example

This section contains several examples that illustrate the operating principle of IPOS programs together with the CANopen fieldbus. For simpler examples with up to three process data that use only the SEW fieldbus profile, it is not necessary to include an IPOS. These examples can be found in the "Fieldbus Device Profile" manual.

4.1 Positioning with five process data

Objective

A MOVIDRIVE® must travel to a 32 bit precise position. It is integrated as Node No. 3 in a 500 kbps CANopen system. Five process data are transmitted from the PLC to the MOVIDRIVE® with the following assignment:

Process output data word of the PLC	Function
1	Control word for inverter
2	Target position High
3	Target position Low
4	Positioning speed (resolution 0.2 1/min)
5	Positioning ramp (resolution 1 ms)

Process input data word of the PLC	Function
1	Status word of the inverter
2	Actual position High
3	Actual position Low
4	Actual speed (resolution 0.2 1/min)
5	Following error

Since a PDO can only transmit four words, the process data 1 through 4 are located in PDO1 and process data 5 is located in PDO2. The 32 bit position is located completely in one PDO and, therefore, transmitted consistently.

**Procedure**

- Set the DIP switches of DFO11A according to the following table.

Designation of the DIP switch	Position
PD2	on
PD1	off
PD0	off
NA6	off
NA5	off
NA4	off
NA3	off
NA2	off
NA1	on
NA0	on
DR1	on
DR0	off
R	on, if applicable, if this node is installed at the end of the CAN bus

- MOVIDRIVE® startup "CFC and IPOS."
- Some parameters must now be configured in MOVIDRIVE®. This can be done manually using MOVITOOLS or with the PLC via the SDO. The parameters must be set according to the following table.

Menu no.	Index	Parameter	Value	Value decimal
100	8461	Setpoint source	Fieldbus	3
101	8462	Control source	Fieldbus	2
870	8304	Process output data description 1	CONTROLWORD1	9
871	8305	Process output data description 2	IPOS PO DATA	12
872	8306	Process output data description 3	IPOS PO DATA	12
873	8307	Process input data description 1	STATUSWORD1	6
874	8308	Process input data description 2	POSITION HI	5
875	8309	Process input data description 3	POSITION LO	4
876	8310	Release PO data	ON	1
600	8335	Binary input DI1	WITHOUT FUNCTION	0
601	8336	Binary input DI2	WITHOUT FUNCTION	0
602	8337	Binary input DI3	WITHOUT FUNCTION	0
603	8338	Binary input DI4	WITHOUT FUNCTION	0
604	8339	Binary input DI5	WITHOUT FUNCTION	0



- The following IPOS program must be compiled with the IPOS^{plus}® compiler, loaded into the inverter and started.

```

/*=====
IPOS source file
=====*/
#include <const.h>
// Help for the predefined structures with
// F1
GSPODATA10 Po10;
SSPIDATA10 Pi10;
SSPOSSPEED Speed;
SSPOSRAMP Ramp;
long Pos32;
/*=====
Main function (IPOS entry function)
=====*/
main()
{
/*-----
Initialization
-----*/
// Process output words are transmitted on
// fieldbus(3)
Po10.BusType = 3;
// 5 process output words are coming in and
// 5 process input words are transmitted again
Po10.Len = 5;
Pi10.Len = 5;
/*-----
Main program loop
-----*/
while(1)
{
// first fetch the process output words
// from the input buffer
_GetSys( Po10, GS_PODATA );
// Process data no. 1 is the control word and
// configured by shell menu item 870
// ( 870 = CONTROLWORD1)
// Process data no. 2 is the target position Hi
// (TargetPos is defined in const.h)
Pos32 = Po10.PO2 << 16;
// Process data no. 3 is the target position Lo
// (TargetPos is defined in const.h)
Pos32 += Po10.PO3;
// Process data no. 4 is the
// positioning speed left and right
// to be converted from 0.2 rpm => 0.1 rpm precision
Speed.CW = Po10.PO4 * 2;
Speed.CCW = Po10.PO4 * 2;
_SetSys(SS_POSSPEED, Speed);
}
}

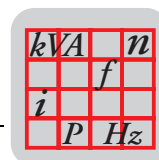
```



```

// Process data no. 5 is the positioning ramp
// left and right in ms
Ramp.Up      = Po10.PO5;
Ramp.Down    = Po10.PO5;
_SetSys(SS_POSRAMP, Ramp);
// and now approach TargetPosition (TargetPos is
// defined in const.h)
TargetPos = Pos32;
// now set the process input words
// process input words 1 through 3 are
// adjusted to the desired size using the
// shell menu items 873 through 875
// (e.g. 873 = STATUSWORD 1)
//      874 = POSITION HI )
//      875 = POSITION LO )
// Process input word 4 represents
// the actual speed
_GetSys(Pi10.PI4, GS_ACTSPEED);
// GetSys(..., GS_ACTSPEED) provides the
// speed with an accuracy of 0.1 rpm
// however, desired is an accuracy of
// 0.2 rpm as process input word => / 2
Pi10.PI4 /= 2;
// Process input word 5 represents
// the actual following error
// (LagDistance is defined in const.h)
Pi10.PI5 = LagDistance;
// and write process input data into
// the output buffer
_SetSys(SS_PIDATA, Pi10);
}
}

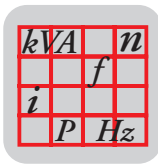
```



5 Appendix

The following object directory lists the index, subindex, data type, and access type of the objects supported by the CANopen option pcb.

Index	Subindex	Function	Data type	Access	Default
0x1000	0	device type	UNSIGNED32	ro	0
0x1001	0	error register	UNSIGNED8	ro	-
0x1002	0	manufacturer status register	UNSIGNED32	ro	-
0x1003	0	pre-defined error field length	UNSIGNED8	rw	-
	1	pre-defined error field	UNSIGNED32	ro	-
	2	pre-defined error field	UNSIGNED32	ro	-
	3	pre-defined error field	UNSIGNED32	ro	-
	4	pre-defined error field	UNSIGNED32	ro	-
	5	pre-defined error field	UNSIGNED32	ro	-
0x1004	0	number of PDOs supported	UNSIGNED32	ro	20002 _{hex}
	1	number of syn. PDOs supported	UNSIGNED32	ro	0
	2	number of asy. PDOs supported	UNSIGNED32	ro	20002 _{hex}
0x1008	0	manufacturer device name	VISIBLE STRING	ro	DFO11A (MDX), No. 823162, ...
0x1009	0	manufacturer hardware version	VISIBLE STRING	ro	8147124.51
0x100A	0	manufacturer software version	VISIBLE STRING	ro	8234760...
0x100B	0	node ID	UNSIGNED32	ro	-
0x100C	0	guard time	UNSIGNED16	rw	0
0x100D	0	life time factor	UNSIGNED8	rw	0
0x100E	0	COB-ID node guarding	UNSIGNED32	ro	-
0x100F	0	number of SDOs supported	UNSIGNED32	ro	10001 _{hex}
0x1200	0	SDO server parameter	UNSIGNED8	ro	2
	1	COB-ID Client->Server	UNSIGNED32	ro	-
	2	COB-ID Server->Client	UNSIGNED32	ro	-
0x1400	0	RX PDO 1 communication parameter	UNSIGNED8	ro	2
	1	COB-ID	UNSIGNED32	ro	-
	2	transmission type	UNSIGNED8	ro	254
0x1401	0	RX PDO 2 communication parameter	UNSIGNED8	ro	2
	1	COB-ID	UNSIGNED32	ro	-
	2	transmission type	UNSIGNED8	ro	254
0x1600	0	RX PDO 1 mapping parameter	UNSIGNED8	ro	-
	1	first mapped object	UNSIGNED32	ro	20200110 _{hex}
	2*	second mapped object	UNSIGNED32	ro	202000210 _{hex}
	3*	third mapped object	UNSIGNED32	ro	202000310 _{hex}
	4*	fourth mapped object	UNSIGNED32	ro	202000410 _{hex}
0x1601	0	RX PDO 2 mapping parameter	UNSIGNED8	ro	-
	1*	first mapped object	UNSIGNED32	ro	20200510 _{hex}
	2*	second mapped object	UNSIGNED32	ro	20200610 _{hex}
	3*	third mapped object	UNSIGNED32	ro	20200710 _{hex}



Index	Subindex	Function	Data type	Access	Default
	4*	fourth mapped object	UNSIGNED32	ro	20200810 _{hex}
0x1800	0	TX PDO 1 communication parameter	UNSIGNED8	ro	2
	1	COB-ID	UNSIGNED32	ro	-
	2	transmission type	UNSIGNED8	ro	254
0x1801	0	TX PDO 2 communication parameter	UNSIGNED8	ro	2
	1	COB-ID	UNSIGNED32	ro	-
	2	transmission type	UNSIGNED8	ro	254
0x1A00	0	TX PDO 1 mapping parameter	UNSIGNED8	ro	-
	1	first mapped object	UNSIGNED32	ro	20210110 _{hex}
	2*	second mapped object	UNSIGNED32	ro	20210210 _{hex}
	3*	third mapped object	UNSIGNED32	ro	20210310 _{hex}
	4*	fourth mapped object	UNSIGNED32	ro	20210410 _{hex}
0x1A01	0	TX PDO 2 mapping parameter	UNSIGNED8	ro	-
	1*	first mapped object	UNSIGNED32	ro	20210510 _{hex}
	2*	second mapped object	UNSIGNED32	ro	20210610 _{hex}
	3*	third mapped object	UNSIGNED32	ro	20210710 _{hex}
	4*	fourth mapped object	UNSIGNED32	ro	20210810 _{hex}
0x2010	0	node event source	UNSIGNED32	rw	1
0x2020	0	number of IPOS process output data (PO)	UNSIGNED32	ro	5
	1	IPOS PO1	SIGNED16	rw	0
	2	IPOS PO2	SIGNED16	rw	0
	3	IPOS PO3	SIGNED16	rw	0
	4	IPOS PO4	SIGNED16	rw	0
	5	IPOS PO5	SIGNED16	rw	0
0x2021	0	number of IPOS process input data (PI)	UNSIGNED32	ro	5
	1	IPOS PI1	SIGNED16	ro	0
	2	IPOS PI2	SIGNED16	ro	0
	3	IPOS PI3	SIGNED16	ro	0
	4	IPOS PI4	SIGNED16	ro	0
	5	IPOS PI5	SIGNED16	ro	0
0x2066	0	MoviLink Service	UNSIGNED32	wo	-
0x2067	0	MoviLink Data	SIGNED32	rw	-
0x206C - 0x5FFF	0	converter specific objects, see MOVIDRIVE® Fieldbus Device Profile	UNSIGNED32	ro	

* These entries depend upon the configurations of the number of process data words in accordance with Section [Assembly/Installation] / [Setting of the DIP Switches] / [Number of Process Data].



6 Index

A

Abort message 14
Access type 31
Address 9
Application example 27

B

Baud rate 8, 9
Bus cables, shielding, and routing 7
BUS ERROR, BUSOFF LED 10
Bus termination 7
BUS WARNING, BUSOFF LED 10
BUSOFF LED 10

C

Cabling 6
CAN messages 12
CANopen addresses 9
Changing states 11
COB-ID 15
COMM LED 10
Compensation current 7
CONFIG. ERROR, BUSOFF LED 10
Connector 6
Connector pin assignment 7
Connector wiring 6
Control mode 19
Control source 19

D

Data exchange 11
Data type 31
Device exchange 11
DIP switch 8
Display elements 10

E

EDS file 21
Error detection 17
Error History 17

F

Factory setting 19
Faulty service execution 14

G

Ground potential fluctuations 7
GUARD LED 10, 16

I

Index 31
Installation 6
Installation of EDS file 21
Inverter control mode 19

L

LED BUSOFF 10
LED COMM 10
LED GUARD 10, 16
LED STATE 10
LEDs 10
Lifeguarding 16

M

Module ID 8, 9

N

NMT services 11
NO ERROR, BUSOFF LED 10
Node Event 17
Node guarding 16
Number of process data words 8, 9

O

Object directory 17, 31
Operating parameter 18
Operating state 16
OVERFLOW, BUSOFF LED 10

P

PDO 14
PDO mapping 15
Pin assignment 7
Process Data Object 14
Process data words, number 8, 9
Process input data words 9
Process output data words 9
Profile area 17
Project Planning 11
Project planning for 1 process data word 22
Project planning of 2 process data words 23
Project planning of 3 process data words 24
Project planning of 4 process data words 25
Project planning of 5 process data words 26

R

Read access, example 13
Read maximum, example 18
Resident storing, operating parameters 18
RX-PDO 14

S

SDO 12
Service Data Object 12
Setpoint source 19
Shielding properties 7
Slave ID 8, 9
Startup 11, 18
STATE LED 10



Subindex 31
Sync Object 17

T

Terminals 7
Terminating resistor 8, 9
Timeout period 16
TX-PDO 15

W

Write access, example 13
Write volatile, example 18

SEW-EURODRIVE *around the world*

Argentina 1619 Garin	Tel. (0 33 27) 45 72 84 Fax (0 33 27) 45 72 21	Malaysia 81100 Johor Barhru	Tel. (07) 3 54 57 07 3 54 94 09 Fax (07) 3 54 14 04
Australia Tullamarine, Victoria 3043 (Melbourne)	Tel. (03) 99 33 10 00 Fax (03) 99 33 10 03	Netherlands NL-3004 AB Rotterdam	Tel. (0 10) 4 46 37 00 Fax (0 10) 4 15 55 52
Austria A-1230 Wien	Tel. (01) 6 17 55 00-0 Fax (01) 6 17 55 00-30	New Zealand Auckland	Tel. (09) 2 74 56 27 2 74 00 77 Fax (09) 2 74 01 65
Belgium B-1300 Wavre	Tel. (0 10) 23 13 11 Fax (0 10) 23 13 36	Norway N-1539 Moss	Tel. (69) 24 10 20 Fax (69) 24 10 40
Brazil 07210 Guarulhos-SP	Tel. (0 11) 64 60-64 33 Fax (0 11) 64 80-43 43	Peru Lima	Tel. (5 11) 349-52 80 Fax (5 11) 349-30 02
Canada Bramalea, Ontario L6T 3W1	Tel. (9 05) 7 91-15 53 Fax (9 05) 7 91-29 99	Portugal P-3050 Mealhada	Tel. (02 31) 20 96 70 Fax (02 31) 20 36 85
Chile RCH-Santiago de Chile	Tel. (02) 6 23 82 03 6 23 81 63 Fax (02) 6 23 81 79	Singapore Singapore 63 86 44	Tel. 86 21 701-705 Fax 8 61 28 27
China Tianjin 300457	Tel. (0 22) 25 32 26 12 Fax (0 22) 25 32 26 11	South Africa Johannesburg 2013	Tel. (0 11) 4 94 43 80 Fax (0 11) 4 94 23 00
Colombia Bogota	Tel. (05 71) 5 47 50 50 Fax (05 71) 5 47 50 44	South Korea Ansan-City	Tel. (03 45) 4 92-80 51 Fax (03 45) 4 92-80 56
Denmark DK-2670 Greve	Tel. 4395 8500 Fax 4395 8509	Spain E-48170 Zamudio	Tel. 9 44 31 84 70 Fax 9 44 31 84 71
Finland FIN-15860 Hollola 2	Tel. (3) 58 93 00 Fax (3) 7 80 62 11	Sweden S-55303 Jönköping	Tel. (0 36) 34 42 00 Fax (0 36) 34 42 80
France F-67506 Haguenau	Tel. 03 88 73 67 00 Fax 03 88 73 66 00	Switzerland CH-4142 Münchenstein (Basel)	Tel. (0 61) 4 17 17 17 Fax (0 61) 4 17 17 00
Great Britain Normanton, West-Yorkshire WF61QR	Tel. 19 24 89 38 55 Fax 19 24 89 37 02	Thailand Bangkok	Tel. (0 38) 21 45 29 / 30 Fax (0 38) 21 45 31
Hong Kong Kowloon, Hong Kong	Tel. 2-7 96-04 77 Fax 2-7 95-91 29	Turkey Istanbul	Tel. (2 16) 4 41 91 63 Fax (2 16) 3 05 58 67
India Baroda	Tel. (02 65) 83 10 86 Fax (02 65) 83 10 87	USA Lyman, S.C. 29365 (South Carolina)	Tel. (8 64) 4 39-87 92 75 37 Fax (8 64) 9 49 30 39
Italy I-20020 Solaro (Milano)	Tel. 02-96 79 97 71 Fax 02-96 79 97 81	Venezuela Valencia	Tel. (0 41) 24 32 32 Fax (0 41) 25 49 16
Japan 438 Shizuoka-ken, Iwata-gun, Toyoda-cho	Tel. (0 53 83) 7 38 11-13 Fax (0 53 83) 7 38 14		