SINAMICS S120

Commissioning Manual · 01/2013
CANopen interface

SINAMICS

SIEMENS

SIEMENS

SINAMICS

S120
CANopen interface

Commissioning Manual

Valid for:
Firmware version 4.6

IH2, 01/2013
6SL3097-4AA00-0BP0

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Order number: 6SL3097-4AA00-0BP0
℗ 01/2013 Technical data subject to change

Preface

SINAMICS documentation

The SINAMICS documentation is organized in the following categories:

- General documentation/catalogs
- User documentation
- Manufacturer/service documentation

More information

The following link provides information on the topics:

- Ordering documentation/overview of documentation
- Additional links to download documents
- Using documentation online (find and browse through manuals/information)
  http://www.siemens.com/motioncontrol/docu

Please send any questions about the technical documentation (e.g. suggestions for improvement, corrections) to the following e-mail address:
docu.motioncontrol@siemens.com

My Documentation Manager

Under the following link there is information on how to create your own individual documentation based on Siemens' content, and adapt it for your own machine documentation:
http://www.siemens.com/mdm

Training

Under the following link there is information on SITRAIN - training from Siemens for products, systems and automation engineering solutions:
http://www.siemens.com/sitrain

FAQs

You can find Frequently Asked Questions in the Service&Support pages under **Product Support**:
http://support.automation.siemens.com

SINAMICS

You can find information on SINAMICS under:
http://www.siemens.com/sinamics

Usage phases and their documents/tools (as an example)

Table 1 Usage phases and the available documents/tools

Usage phase	Document/tool
Orientation	SINAMICS S Sales Documentation
Planning/configuration	• SIZER engineering tool • Configuration Manuals, Motors
Deciding/ordering	SINAMICS S120 catalogs • SIMOTION, SINAMICS S120 and Motors for Production Machines (Catalog PM 21) • SINAMICS and motors for single-axis drives (catalog D 31) • SINUMERIK & SINAMICS Equipment for Machine Tools (Catalog NC 61) • SINUMERIK 840D sl Type 1B Equipment for Machine Tools (Catalog NC 62)
Installation/assembly	• SINAMICS S120 Equipment Manual for Control Units and Additional System Components • SINAMICS S120 Equipment Manual for Booksize Power Units • SINAMICS S120 Equipment Manual for Chassis Power Units • SINAMICS S120 Equipment Manual for AC Drives • SINAMICS S120M Equipment Manual Distributed Drive Technology
Commissioning	• STARTER commissioning tool • SINAMICS S120 Getting Started • SINAMICS S120 Commissioning Manual • SINAMICS S120 CANopen Commissioning Manual • SINAMICS S120 Function Manual • SINAMICS S120 Safety Integrated Function Manual • SINAMICS S120/S150 List Manual
Usage/operation	• SINAMICS S120 Commissioning Manual • SINAMICS S120/S150 List Manual
Maintenance/servicing	• SINAMICS S120 Commissioning Manual • SINAMICS S120/S150 List Manual
References	• SINAMICS S120/S150 List Manual

Target group

This documentation is intended for machine manufacturers, commissioning engineers, and service personnel who use the SINAMICS drive system.

Benefits

This Manual describes all the information, procedures and operational instructions required for commissioning and servicing SINAMICS S120.

Standard scope

The scope of the functionality described in this document can differ from the scope of the functionality of the drive system that is actually supplied.

- It may be possible for other functions not described in this documentation to be executed in the drive system. However, no claim can be made regarding the availability of these functions when the equipment is first supplied or in the event of servicing.
- Functions that are not available in a particular product version of the drive system may be described in the documentation. The functionality of the supplied drive system should only be taken from the ordering documentation.
- Extensions or changes made by the machine manufacturer must be documented by the machine manufacturer.

For reasons of clarity, this documentation does not contain all of the detailed information on all of the product types. This documentation cannot take into consideration every conceivable type of installation, operation and service/maintenance.

Technical Support

Country-specific telephone numbers for technical support are provided in the Internet under **Contact**:

http://www.siemens.com/automation/service&support

EC Declaration of Conformity

The EC Declaration of Conformity for the EMC Directive can be found on the Internet at:

http://support.automation.siemens.com

There – as a search term – enter the number **15257461** or contact your local Siemens office.

Benefits

Note

This "SINAMICS S120 Commissioning Manual CANopen" describes the steps involved when commissioning a CANopen interface in the SINAMICS drive line-up.

This Commissioning Manual supplements the description of "Initial commissioning servo control booksize format" to include a description of the initial commissioning procedure for the CANopen communication interface:

- SINAMICS S120 with the CBC10 communication module
- Detailed instructions on commissioning the entire SINAMICS drive line-up are available in the SINAMICS S120 Commissioning Manual.

ESD notices

Electrostatic sensitive devices (ESDs) are individual components, integrated circuits, boards or devices that may be damaged by either electrostatic fields or electrostatic discharge.

NOTICE

Damage caused by electric fields or electrostatic discharge

Electric fields or electrostatic discharge can cause malfunctions as a result of damaged individual components, integrated circuits, boards or devices.

- Package, store, transport and send electronic components, boards and devices only in the original product packaging or in other suitable materials, e.g. conductive foam rubber or aluminum foil.
- Only touch the components, boards and devices if you are grounded using one of the following measures:
  - Wearing an ESD wrist strap
  - Wearing ESD shoes or ESD grounding straps in ESD areas with conductive flooring
- Only place down electronic components, boards or devices on conductive surfaces (table with ESD surface, conductive ESD foam, ESD packaging, ESD transport container).

Safety instructions

⚠ DANGER

Danger of death when live parts are touched

Touching live components results in death or severe injuries.

- Only work on electrical equipment if you are qualified to do so.
- When carrying out any work always comply with country-specific safety rules and regulations.

Generally, six steps apply when establishing safety:

1. Prepare for shutdown and notify team members who will be affected by the procedure.
2. Switch off the machine so that it is in a no-voltage condition.
   - Switch off the machine.
   - Wait until the discharge time specified on the warning labels has elapsed.
   - Check that it really is in a no-voltage condition, from phase conductor to phase conductor and phase conductor to protective conductor.
   - Ensure that all auxiliary circuits are in a no-voltage condition.
   - Ensure that motors cannot move.
3. Identify all additional dangerous sources of energy e.g. compressed air, hydraulic systems, water.
4. Isolate or neutralize all dangerous energy sources, for example, by closing switches, grounding or short-circuiting and closing valves.
5. Lock out all energy sources so that they cannot be switched on again.
6. Make sure that the machine is completely locked out ... and that you have the right machine!

After you have completed the work, restore the operational readiness in the inverse sequence.

⚠ DANGER

General safety notices

- Commissioning is absolutely prohibited until it has been completely ensured that the machine, in which the components described here are to be installed, is in full compliance with the provisions of the EC Machinery Directive.
- SINAMICS devices and AC motors must only be commissioned by suitably qualified personnel.
- Personnel must take into account the information provided in the technical customer documentation for the product, and be familiar with and observe the specified danger and warning notices.
- When electrical equipment and motors are operated, the electrical circuits automatically conduct a dangerous voltage.
- Dangerous mechanical movements may occur in the system during operation.
- All of the work carried-out on the electrical machine or system must be carried-out with it in a no-voltage condition.
- SINAMICS devices with three-phase motors must only be connected to the power supply via an AC-DC residual-current-operated device with selective switching if it has been verified that the SINAMICS device is compatible with the residual-current-operated device in accordance with IEC 61800-5-1.
- The successful and safe operation of this equipment and motors is dependent on correct transport, proper storage, installation and mounting as well as careful operator control, service and maintenance.
- For special versions of the devices and motors, information and data in the catalogs and quotations additionally apply.
- In addition to the danger and warning information provided in the technical customer documentation, the applicable national, local, and plant-specific regulations and requirements must be taken into account.
- Only protective extra-low voltages (PELVs) that comply with EN 60204-1 may be connected to any connections and terminals between 0 and 48 V.

⚠ WARNING

Danger of death due to unexpected movement of machines when using mobile wireless devices or mobile phones

Using mobile radios or mobile phones with a transmit power > 1 W closer than approx. 2 m to the components may cause the devices to malfunction, influence the functional safety of machines therefore putting people at risk or causing material damage.

Switch off mobile radios and mobile telephones when you are close to the components.

Table of contents

Add infeed

1

1.1 Previous knowledge

To fully understand this Commissioning Manual, you must be familiar with CANopen terminology.

You must be familiar with the following standards:

Note

SINAMICS with CANopen complies with the following standards:

- CiA 301 (Application Layer and Communication Profile)
- CiA 303-3 (Indicator Specification)
- CiA 306 (electronic data sheet specification for CANopen)
- CiA 402 (Device Profile for Drives and Motion Control)

1.2 CAN bus structure for SINAMICS

The following diagram shows an example using SINAMICS S120 of how the hardware and software are arranged when a CANopen interface is commissioned.

The diagram shows the following:

- How a master application of a CANopen user is connected to a SINAMICS drive line-up.
- The CAN bus interface of the "CBC10" Communication Board.
- The associated CANopen slave software on the Control Unit and the meaning of the terms "transmit" and "receive", which are used for the transmit and receive message frames during commissioning.
- How a PC on which the STARTER commissioning tool has been installed can be connected via ETHERNET.

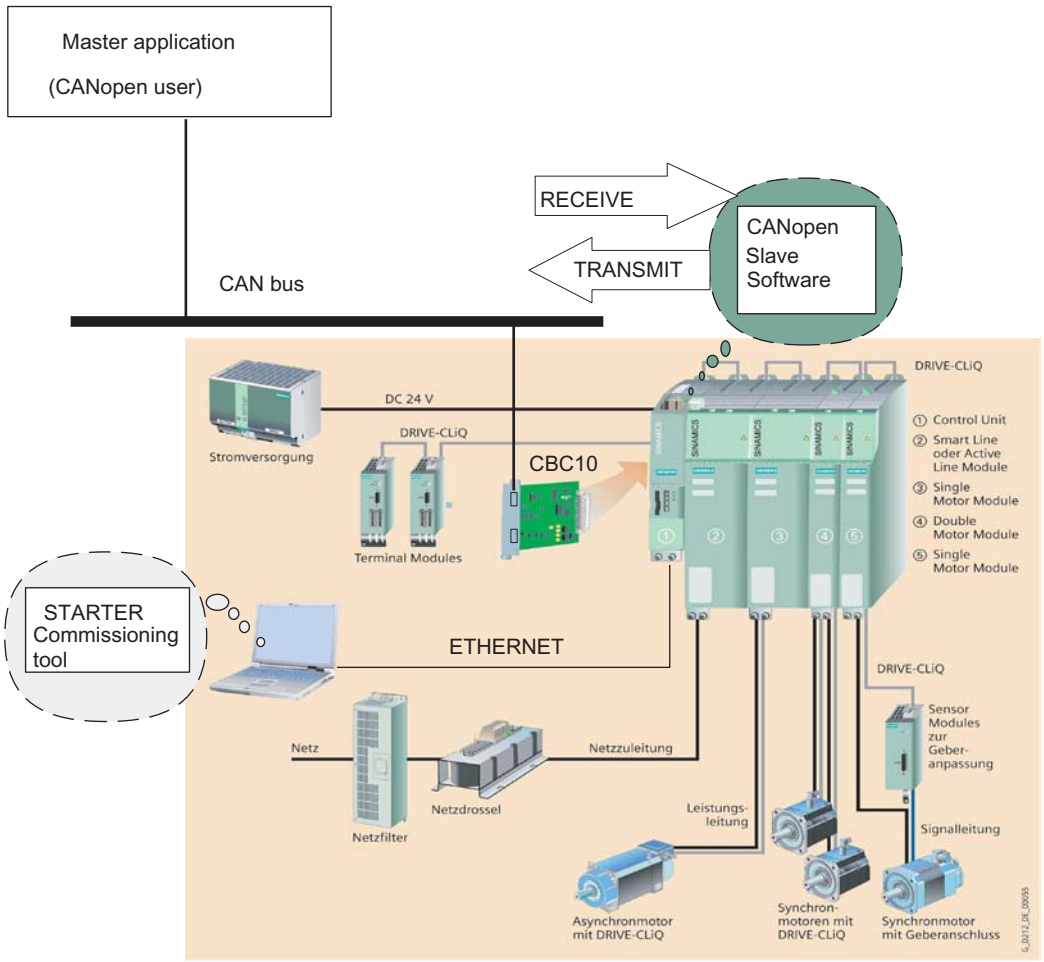


Figure 1-1 SINAMICS S120 drive line-up with CAN bus

1.3 CANopen object directory

When the drive objects are initialized, the CANopen objects are initialized in the object directory for the SINAMICS drive line-up (CANopen slave software).

Object directory

The following diagram shows the distribution of CANopen objects involved in the communications (the values are hexadecimal values):

- Communication objects for the Control Unit
- Drive-dependent communication objects
- Manufacturer-specific objects
- Drive-dependent objects of the drive profile "DSP 402"

CANopen supports a maximum of eight drive objects.

In the section "Communication objects (Page 133)", a table is provided listing the communication objects of CANopen and SINAMICS parameters that are used in SINAMICS for communication via the CANopen interface.

CANopen object directory

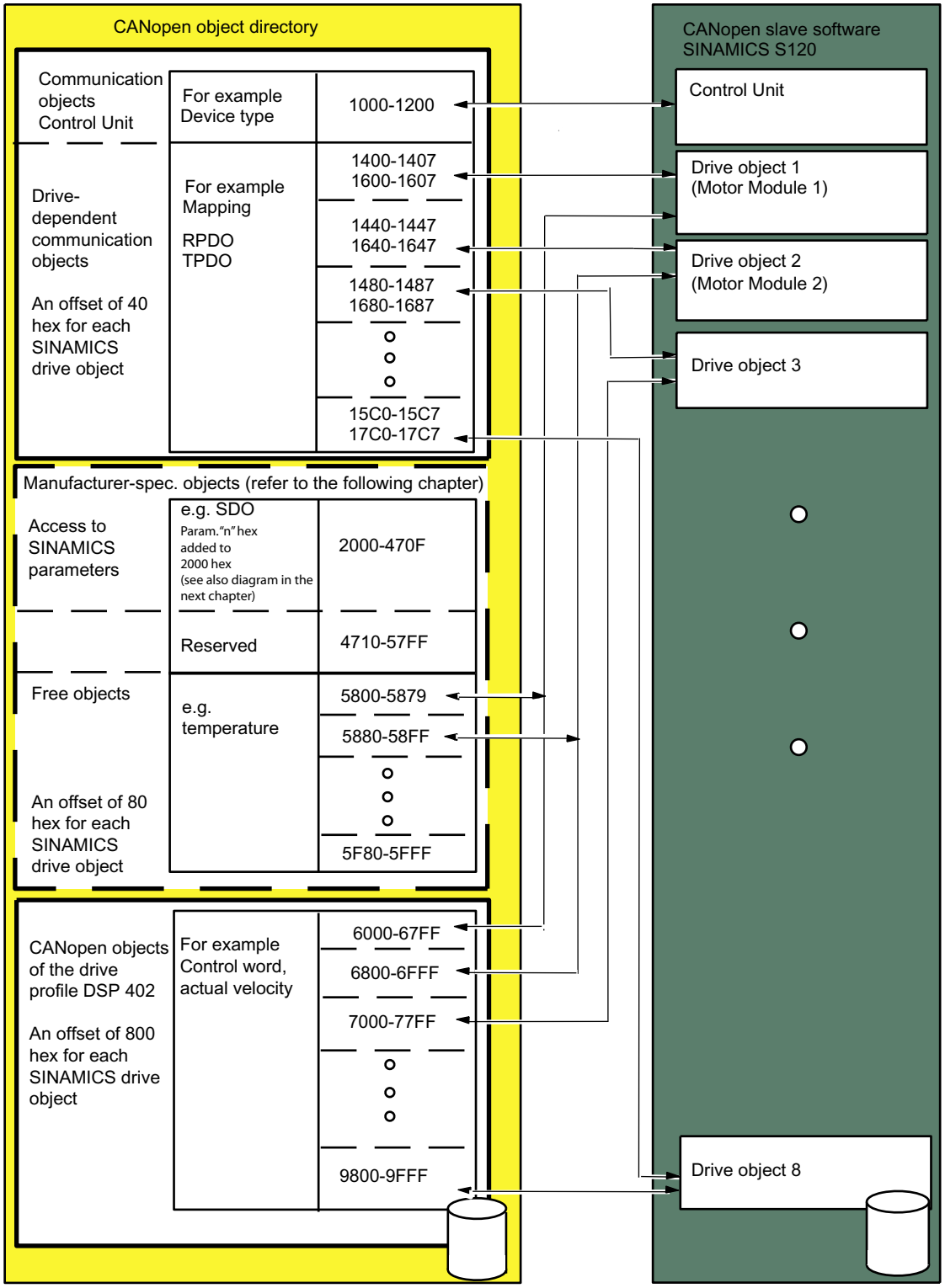


Figure 1-2 CANopen object directory

1.4 Manufacturer-specific objects

As shown in the previous section, there is a range for manufacturer-specific objects in the CANopen object directory.

Manufacturer-specific objects

Manufacturer-specific objects are defined as:

- Objects to access SINAMICS parameters
- Free objects to send/receive process data

This manufacturer-specific range starts in the object directory from address "2000 hex" and ends at "5FFF hex".

1.4.1 Objects to access SINAMICS parameters

Data values of the SINAMICS parameters can be accessed using the objects in the range from "2000 hex" to "470F hex" of the object directory.

Example

The following diagram shows the distribution of objects in the object directory.

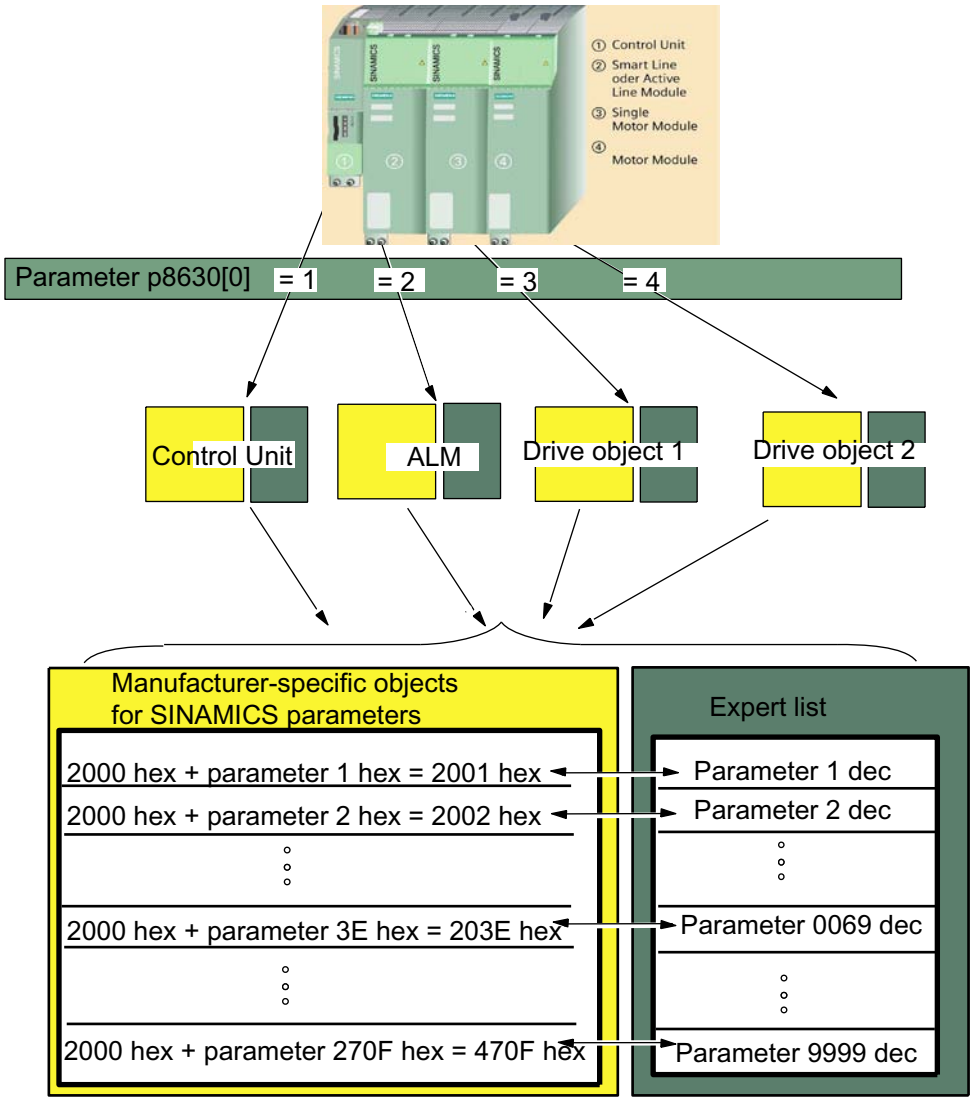


Figure 1-3 Manufacturer-specific objects

Parameter p8630[0…2]

Using parameter p8630 indices 0 to 2, you can define the way drive objects are accessed in SINAMICS. The following access methods are available:

- The drive object is selected in p8630[0]
  - 0: Virtual objects cannot be accessed
  - 1: Control Unit
  - 2…65535: Drive object
- Subindex range in p8630[1]
  - 0: 0…255
  - 1: 256…511
  - 2: 512…767
  - 3: 768…1023
- Parameter range in p8630[2]
  - 0: 0…9999
  - 1: 10000…19999
  - 2: 20000…29999
  - 3: 30000…39999
  - 4: 40000…49999
  - 5: 50000…59999

General procedure

All SINAMICS parameters can be addressed via SDO access.

This functions as follows:

- All SINAMICS parameters can be addressed using the objects "2000 hex" to "470F hex".
- The SDO access internally converts manufacturer-specific objects to parameters. "2000 hex" is added to the parameter number converted to a hexadecimal value.

  This number is the object number in the SDO request required to access the SINAMICS parameter.
- Since the parameter range of a SINAMICS drive object occupies the entire object space of the parameter access in the manufacturer-specific range, the drive object that is to be accessed is selected in parameter p8630[0] in SINAMICS.
- A SINAMICS parameter can be an r or p parameter. The manufacturer-specific objects contain the data values for these parameters.
- Depending on the switch position of parameter p8630[0], the data values for the modules can be read or written.

  If, for example, parameter r0062 of the setpoint velocity is to be read out from drive object 1, then:

  – The switch must be set to "3" in parameter p8630[0] (see previous diagram),
  – The parameter r0062 is converted to a hexadecimal value and "2000 hex" is added. With this hexadecimal number "203E hex", which corresponds to the object number, the parameter r0062 can be accessed via an SDO request.

Note

The switch in parameter p8630[0] determines the structure of the drive line-up. If an Active Line Module (ALM) is not installed, the count for the first drive object starts with "2". This number represents the drive ID.
The drive ID for each drive can be displayed with the parameter p8743[0...7] . The drive ID for the first drive is in the index "0".

1.4.2 Free objects

In the object directory (OD), you have the possibility of using free objects for process data (PZD) in the range from "5800 hex" to "5FFF hex" (also refer to section CANopen object directory (Page 15)).

Free objects

For each drive object, the following objects that can be freely interconnected are available in the objects directory:

Table 1- 1 Freely-interconnectable objects for drive object 1

OD index (hex)	Description	Data type per PZD
5800 to 580F	16 freely-interconnectable receive process data	Integer 16
5810 to 581F	16 freely-interconnectable transmit process data	Integer 16
5820 to 5827	8 freely-interconnectable receive process data	Integer32
5828 to 582F	Reserved	–
5830 to 5837	8 freely-interconnectable transmit process data	Integer32
5838 to 5879	Reserved	–

Note

The free objects for additional drive objects are formed by adding the offset "80 hex" to the object number of the freely interconnectable object.

For example, drive object 2 begins at "5880 hex".

You can interconnect any process data objects using receive/transmit words/double words of the receive and transmit buffer.

Transfer

Free objects can be accessed using the following communication services/utilities:

- PDO
- SDO

1.5 Number of PDO that can be created

PDO (process data object) are used to transfer process data, which is used for real-time access to selected data. For certain variables, mappings to certain PDOs are preconfigured.

PDO are, for example, control words such as "setpoint velocity", "actual velocity", etc.

Note

In a SINAMICS drive object a maximum of eight receive and eight send PDO can be defined. The SINAMICS drive line-up can manage a maximum of 25 receive PDO.

Since a maximum of eight PDOs can be defined per SINAMICS drive object, this means that for maximum number of RPDO created, a maximum of three SINAMICS drive objects would be possible. An additional RPDO could also be created.

1.6 Send and receive message frames for process data

In the SINAMICS drive line-up, the STARTER commissioning tool offers two options for commissioning a CANopen interface:

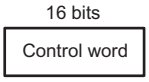
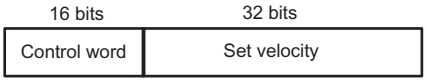
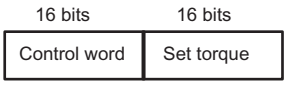
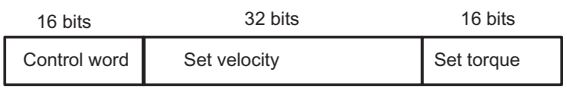
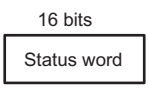
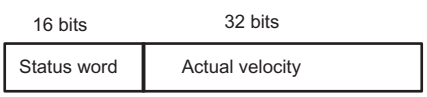
1. Using predefined message frames ("Predefined Connection Set") and COB IDs.
2. Using free PDO mapping (user-defined message frames).

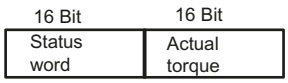
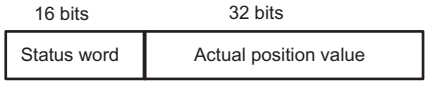
In the first step, you are advised to activate the "predefined connection set" and then update the preconfigured parameters in accordance with your application in the free PDO mapping process. For automatic commissioning, the "Predefined Connection Set" is already set.

Predefined message frames for SINAMICS

The following process data objects are predefined and mapped in the "predefined connection set" in the receive and transmit message frames for the corresponding drive objects. The table contains the hexadecimal object values for drive object 1. Each additional drive object begins with an offset of "800 hex".

Table 1- 2 Process data objects in the predefined connection set

Type	Process data	Message frame
Receive message frame	RPDO1	 6040 hex
	RPDO2	 6040 hex + 60FF hex
	RPDO3	 6040 hex + 6071 hex
	RPDO4	 6040 hex + 60FF hex + 6071 hex
Transmit message frame	TPDO1	 6041 hex
	TPDO2	 6041 hex + 606C hex

Type	Process data	Message frame
	TPDO3	 6041 hex + 6077 hex
	TPDO4	 6041 hex + 6063 hex

Note

For each drive object in the expert list, the process data objects for mapping the message frames begin as follows:

- For receive message frames starting from parameter p8710
- For transmit message frames starting from parameter p8730

Users must create the corresponding BICO interconnections of the PZD interface.

1.7 PDO mapping

PDO mapping is used to map drive objects (process data, e.g. setpoints or actual values) and "free objects" from the object directory for the PDO service as message frame.

The PDO message frame transfers the data values of these objects.

Once detected, each drive object is assigned up to eight receive and eight transmit PDO.

A CAN message frame can transfer up to 8 bytes of user data. The user decides which data is to be transferred in a PDO.

Example

The following diagram uses an example to illustrate PDO mapping (values are hexadecimal (e.g. object size "20 hex" = 32 bits)):

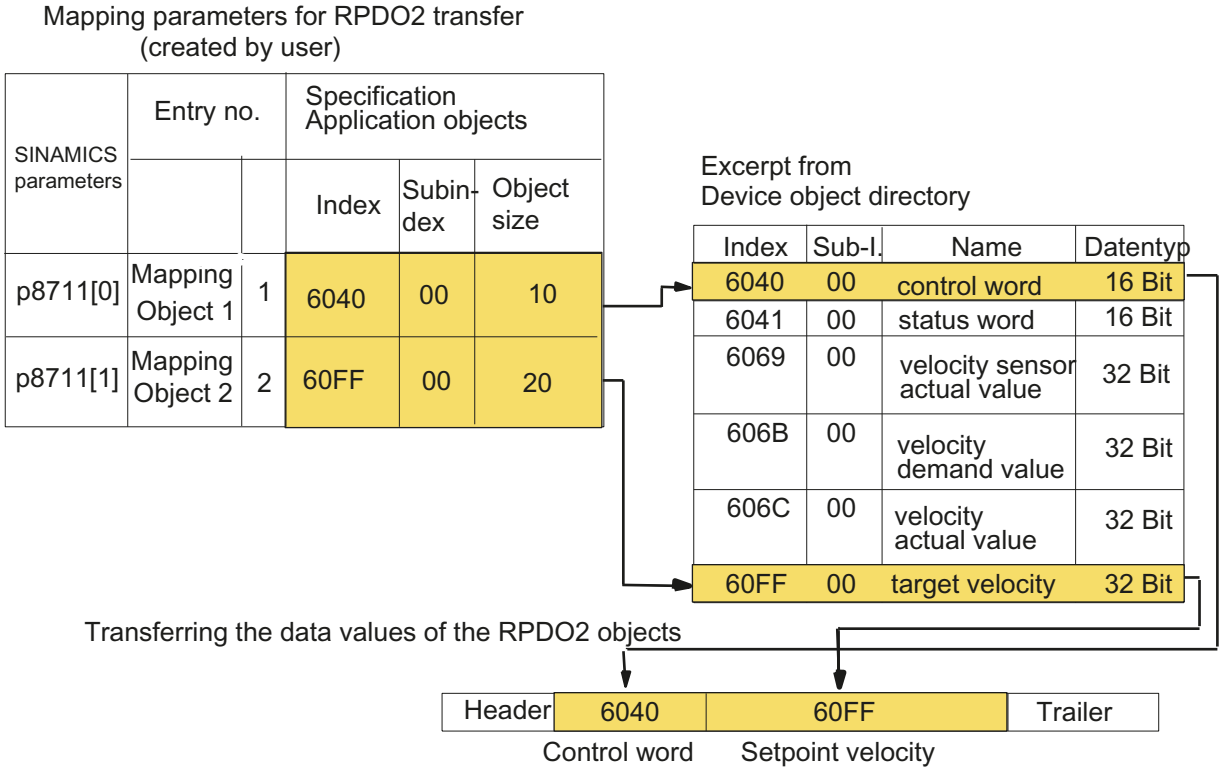


Figure 1-4 PDO mapping

1.8 COB-ID

Each COB (communication object) can be uniquely identified by means of an identifier, which is a part of the COB. CAN specification 2.0A supports up to 2048 COB, which are identified by means of 11-bit identifiers.

A list of COB identifiers, which contains all the COB that can be accessed via CAN, is available in the object directory for the relevant SINAMICS drive unit.

The COB ID prioritizes the communication objects by means of an identifier assignment.

General procedure

The COB ID is user defined.

CANopen defines a preset identifier assignment ("Predefined Connection Set") for the communication objects. The following table shows how the preset identifier assignment is structured.

Table 1- 3 Identifier assignment

Communication objects	Function code		Resulting COB ID		OD index (hex)[1]
	dec	bin	hex	Explanation	
NMT commands (NMT) [2]	0	0000	0	0 dec	–
Sync message (SYNC)	1	0001	80	128 dec	1005, 1006, 1007
Alarm objects (EMERGENCY)	1	0001	81–FF	129–255 dec	1014, 1015
Tx–PDO1	3	0011	181–1FF	180 hex + node ID	1800
Rx–PDO1	4	0100	201–27F	200 hex + node ID	1400
Tx–PDO2	5	0101	281–2FF	280 hex + node ID	1801
Rx–PDO2	6	0110	301–37F	300 hex + node ID	1401
Tx–PDO3	7	0111	381–3FF	380 hex + node ID	1802
Rx–PDO3	8	1000	401–47F	400 hex + node ID	1402
Tx–PDO4	9	1001	481–4FF	480 hex + node ID	1803
Rx–PDO4	10	1010	501–57F	500 hex + node ID	1403
Tx–SDO[2]	11	1011	581–5FF	580 hex + node ID	1200
Rx–SDO[2]	12	1100	601–67F	600 hex + node ID	1200
Node monitoring (NMT error control) [2]	14	1110	701–77F	700 hex + node ID	100C, 100D

[1] The OD index for the Tx and Rx PDO depends on the number of drive objects in the drive line-up. Each additional drive object begins with the description in an offset of 40 hex (with Tx/Rx, the x stands for the relevant SINAMICS drive object, T = transmit, R = receive)
e.g.: For TPDO, the OD index for the first drive object begins with 1800 hex and, for RPDO, with 1400 hex; each additional drive object begins with an offset of 40 hex –> 1840 hex and 1440 hex for the second drive object, and so on.
With the "Predefined Connection Set", the COB IDs are incremented by 1 for each additional SINAMICS drive object.

[2] These COB IDs are set by default.

1.9 SDO services

SDO services allow you to access the object directory for the connected drive unit. An SDO connection is a peer-to-peer connection between an SDO client and a server.

The drive unit with its object directory is an SDO server.

The identifiers for the first SDO server of a drive unit are defined according to CANopen.

For communication between the server and client, the following applies:

- Server receives: COB-ID 600 hex + node ID
- Server sends: COB-ID 580 hex + node ID

Properties

The SDOs have the following properties:

- Confirmed transfer of objects
- The transfer procedure is always asynchronous
- Transfer of values greater than 4 bytes (normal transfer)
- Transfer of values with no more than 4 bytes (expedited transfer)
- Corresponds with the acyclic PROFIBUS parameter channel
- All drive unit variables can be addressed via SDO

Structure of the SDO protocols

The SDO services use the appropriate protocol depending on the task. The following lists the most important protocols for the SDO services.

SDO Protocol Write

This protocol is used to write data to the drive unit.

Confirmation with the signal "Write Response".

Table 1- 4 SDO Protocol Write

Write Request			
CANopen Master Request ----> SINAMICS message			
Byte 0	**Bytes 1...2**	**Byte 3**	**Bytes 4...7**
cs = 2F hex	index	sub index	data byte 4
cs = 2B hex	index	sub index	data byte 4...5
cs = 27 hex	index	sub index	data byte 4...6
cs = 23 hex	index	sub index	data byte 4...7
Write Response			
SINAMICS confirmation ----> CANopen Master Response			
Byte 0	**Bytes 1...2**	**Byte 3**	**Bytes 4...7**
cs = 60	index	sub index	reserved

Explanation of the above table:

- cs: command specifier
  In the cs, it is determined how much of the sent byte represents relevant data. The rest has no significance.
- index, subindex:
  stands for the object index and subindex, which are to be accessed.
- reserved:
  reserved for later use, always 0.

SDO Protocol Read

This protocol is used to read data from the drive unit.

Request to read via the signal "Read Request".

Table 1- 5 SDO Protocol Read

Read Request			
CANopen Master Request ----> SINAMICS message			
Byte 0	**Bytes 1...2**	**Byte 3**	**Bytes 4...7**
cs = 40	index	sub index	reserved
SINAMICS confirmation <---- CANopen Master Response			
Read Response			
Byte 0	**Bytes 1...2**	**Byte 3**	**Bytes 4...7**
cs = 4F hex	index	sub index	data byte 4
cs = 4B hex	index	sub index	data byte 4...5
cs = 47 hex	index	sub index	data byte 4...6
cs = 43 hex	index	sub index	data byte 4...7

Explanation of the above table:

- cs: command specifier
  In the cs, it is determined how much of the sent byte represents relevant data. The rest has no significance.
- index, subindex: stands for the SDO index and subindex.
- reserved:
  reserved for later use, always 0.

SDO Protocol Cancel Transfer Protocol

This protocol is used to perform the SDO service "Cancel Transfer Protocol".

Table 1- 6 SDO Protocol Cancel Transfer Protocol

SINAMICS Request ----> CANopen Master Message			
Error Response			
Byte 0	**Bytes 1..2**	**Byte 3**	**Bytes 4..7**
cs = 80	index	subindex	abort code

Explanation of the above table:

- cs: command specifier
  This object is always 4 bytes.
- index, subindex: stands for the SDO index and subindex.
- abort code:
  Contains 4 bytes abort code about the reason for the abort.
  The abort code is a value with the format UNSIGNED32.

SDO abort codes

Table 1- 7 SDO abort codes

Abort code	Description
0503 0000 hex	Toggle bit not alternated.
0504 0000 hex	SDO protocol timed out.
0504 0001 hex	Client/server command specifier not valid or unknown.
0504 0002 hex	Invalid block size (block mode only).
0504 0003 hex	Invalid sequence number (block mode only).
0504 0004 hex	CRC error (block mode only).
0504 0005 hex	Out of memory.
0601 0000 hex	Unsupported access to an object.
0601 0001 hex	Attempt to read a write only object.
0601 0002 hex	Attempt to write a read only object.
0602 0000 hex	Object does not exist in the object dictionary.
0604 0041 hex	Object cannot be mapped to the PDO.
0604 0042 hex	The number and length of the objects to be mapped would exceed PDO length.
0604 0043 hex	General parameter incompatibility reason.
0604 0047 hex	General internal incompatibility in the device.
0606 0000 hex	Access failed due to an hardware error.
0607 0010 hex	Data type does not match, length of service parameter does not match.
0607 0012 hex	Data type does not match, length of service parameter too high.
0607 0013 hex	Data type does not match, length of service parameter too low.
0609 0011 hex	Sub-index does not exist.
0609 0030 hex	Value range of parameter exceeded (only for write access).
0609 0031 hex	Value of parameter written too high.
0609 0032 hex	Value of parameter written too low.
0609 0036 hex	Maximum value is less than minimum value.
060A 0023 hex	Resource not available: SDO connection.
0800 0000 hex	General error.
0800 0020 hex	Data cannot be transferred or stored to the application.
0800 0021 hex	Data cannot be transferred or stored to the application because of local control.
0800 0022 hex	Data cannot be transferred or stored to the application because of the current device state.
0800 0023 hex	Object dictionary dynamic generation failed or no object dictionary is present (e. g. object dictionary is generated from file and generation fails because of a file error).

1.10 PDO services

The real-time data transfer for CANopen takes place using "Process Data Objects (PDO)".

The PDO are linked with entries in the object directory and represent the interface with the drive objects. Data type and mapping of the drive objects in a PDO are determined by the PDO mapping structure inside the object directory. The number of PDO and the mapping of the drive objects in a PDO are transmitted to the unit during the unit configuration process.

This transmission is implemented at the corresponding entries in the object directory via SDO services.

PDOs are used in two different ways. The transmit PDO (TPDO) is responsible for transmitting data and the receive PDO (RPDO) is responsible for receiving data. CANopen devices that support TPDO are called PDO producers and CANopen devices that support RPDO are called PDO consumers. The PDO is identified by the PDO communications parameter and the PDO mapping parameter. The structure of this parameter is listed in both the following tables.

Parameter

Table 1- 8 PDO communication parameters 1400 hex ff, 1800 hex ff

Subindex	Name	Data type
00 hex	Number of subindices	UNSIGNED8
01 hex	COB-ID of the PDO	UNSIGNED32
02 hex	Transmission type of the PDO	UNSIGNED8
03 hex*	Inhibit time	UNSIGNED16
04 hex*	Reserved	UNSIGNED8
05 hex*	Event timer	UNSIGNED16

* Only valid for 1800 hex ff

Table 1- 9 PDO mapping parameter 1600h ff, 1A00h ff

Subindex	Name	Data type
00 hex	Number of mapped objects in the PDO	UNSIGNED8
01 hex	First object to be mapped	UNSIGNED32
02 hex	Second object to be mapped	UNSIGNED32
03 hex	Third object to be mapped	UNSIGNED32
04 hex	Fourth object to be mapped	UNSIGNED32

The PDO communications parameter describes the communication options of the PDO. The PDO mapping parameter contains information on the content of the PDO.

Communication parameters and mapping parameters need to be defined for each PDO.

The definition of the PDO within a unit profile always refers to the first logical device inside a CANopen device. If the PDO definition should be valid for the second logical device, the PDO number of the CANopen device used needs to be increased by the value 64 (40h hex) corresponding to the PDO number in the device profile (see following table).

Table 1- 10 Example: PDO number calculation

Logical device in the CANopen device	PDO number in the CANopen device	PDO number in the device profile
1. logical device	PDO number + 0 (PDO1 to PDO64)	PDO number (PDO1 to PDO64)
2. logical device	PDO number + 64 (PDO65 to PDO128)	PDO number (PD O1 to PDO64)
3. logical device	PDO number + 128 (PDO129 to PDO192)	PDO number (PDO1 to PDO64)
4. logical device	PDO number + 192 (PDO193 to PDO256)	PDO number (PDO1 to PDO64)
5. logical device	PDO number + 256 (PDO257 to PDO320)	PDO number (PDO1 to PDO64)
6. logical device	PDO number + 320 (PDO321 to PDO384)	PDO number (PDO1 to PDO64)
7. logical device	PDO number + 384 (PDO385 to PDO448)	PDO number (PDO1 to PDO64)
8. logical device	PDO number + 448 (PDO449 to PDO512)	PDO number (PDO1 to PDO64)

The indices of the corresponding entries in the object directory are calculated with the following formula:

- RPDO communication parameter index = 1400 hex + RPDO number -1
- TPDO communication parameter index = 1800 hex + TPDO number -1
- RPDO mapping parameter index = 1600 hex + RPDO number -1
- TPDO mapping parameter index = 1A00 hex + TPDO number -1

Data transfer types

The following PDO transmission types are available:

- Synchronous transmission
- Asynchronous transmission

In order that the communicating devices remain synchronized during transmission, a synchronization object (SYNC object) must be transmitted at periodic intervals.

The following diagram shows the principle of synchronous and asynchronous transmission:

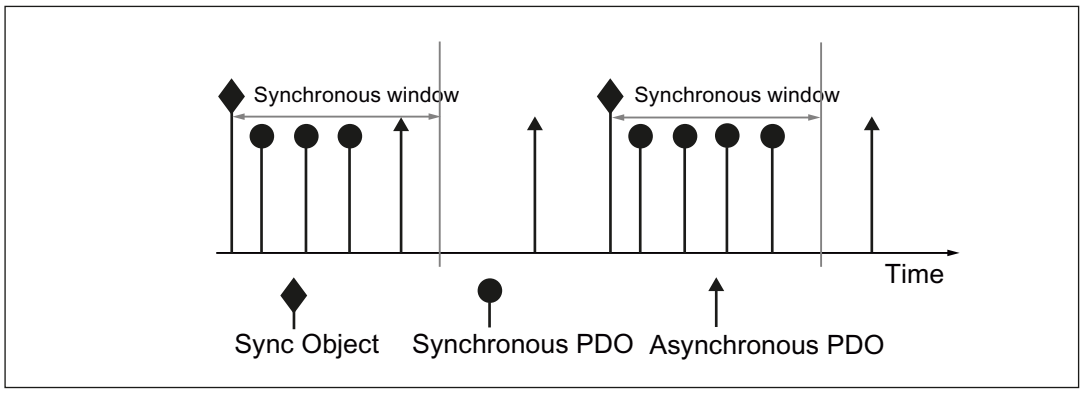


Figure 1-5 Principle of synchronous and asynchronous transmission

The type of transmission is characterized by a PDO's parameter for transmission type.

For synchronous TPDOs, the transmission type also identifies the transmission rate as a factor of the SYNC object "transmission intervals". Here, the transmission type "1" means that the message will be transmitted in every SYNC object cycle. The transmission type "n" means that the message will be transmitted in every nth SYNC object cycle.

Asynchronous TPDOs are transmitted without reference to the SYNC signal.

Data from synchronous RPDOs, which are received after a SYNC signal, is not transmitted to the application until after the next SYNC signal.

The SYNC object is sent periodically from the SYNC producer. The SYNC signal represents the basic network cycle. The time interval between two SYNC signals is is determined by the standard parameter "Communication cycle time".

In order to provide real-time access to the CAN bus, the SYNC object has a very high-priority identifier, the factory setting is 80h. The service runs unconfirmed. The SYNC object identifier can be changed to another value. Then all the CANopen slaves on the same bus should also be changed appropriately, so that communication remains possible.

The SYNC object identifier has the object index 1005 hex and the time between two SYNC signals is defined in object index 1006 hex.

Note

The SYNC signal does not synchronize the applications in the SINAMICS drive, only the communication on the CANopen bus

Data from asynchronous RPDO is passed on to the application directly.

1.11 BICO interconnection in conjunction with CANopen

Each drive unit contains a large number of interconnectable input and output variables as well as internal control variables.

BICO technology (Binector Connector Technology) allows the drive to be adapted to a wide variety of requirements.

Digital and analog signals, which can be connected freely by means of BICO parameters, are identified by the prefix BI, BO, CI or CO in their parameter name.

These parameters are identified accordingly in the parameter list or in the function diagrams.

There are:

- Binectors, with
  BI: binector input, BO: binector output
- Connectors, with
  CI: connector input, CO: connector output

Interconnecting signals using BICO technology

To interconnect two signals, a BICO input parameter (signal sink) must be assigned to the required BICO output parameter (signal source).

Note

Detailed information on the interconnection of signals with BICO technology can be found in the SINAMICS S120 Commissioning Manual or in the S120/S150 List Manual.

BICO interconnection for CANopen parameters

The drive objects created during PDO mapping are interconnected in a receive and transmit buffer for CANopen. Each process data object only appears once in each buffer.

For an explanation of how the process data objects for the receive and transmit buffer are interconnected, see section "Interconnecting process data in the receive and transmit buffers (Page 100)".

1.12 Bootup protocol

After the bootup of the NMT slave, this protocol signals that the NMT slave has taken on the state "Pre-Operational" after having been in the "Initialization" state.

Bootup protocol COB-ID = 700h + node ID

1 data byte with the value 0 is transferred.

1.13 Network management (NMT service)

Network management is node oriented and has a master-slave structure.

The NMT service can be used to initialize, start, monitor, reset, or stop nodes. All NMT services have the COB-ID = 0.

The drive unit is an NMT slave.

Overview

The following diagram shows a status diagram of a CANopen node with a SINAMICS interface. This is followed by a table listing the NMT services that are available for controlling the status transitions.

The NMT service is described in detail in the CANopen standard "CiA 301 (Application Layer and Communication Profile)"

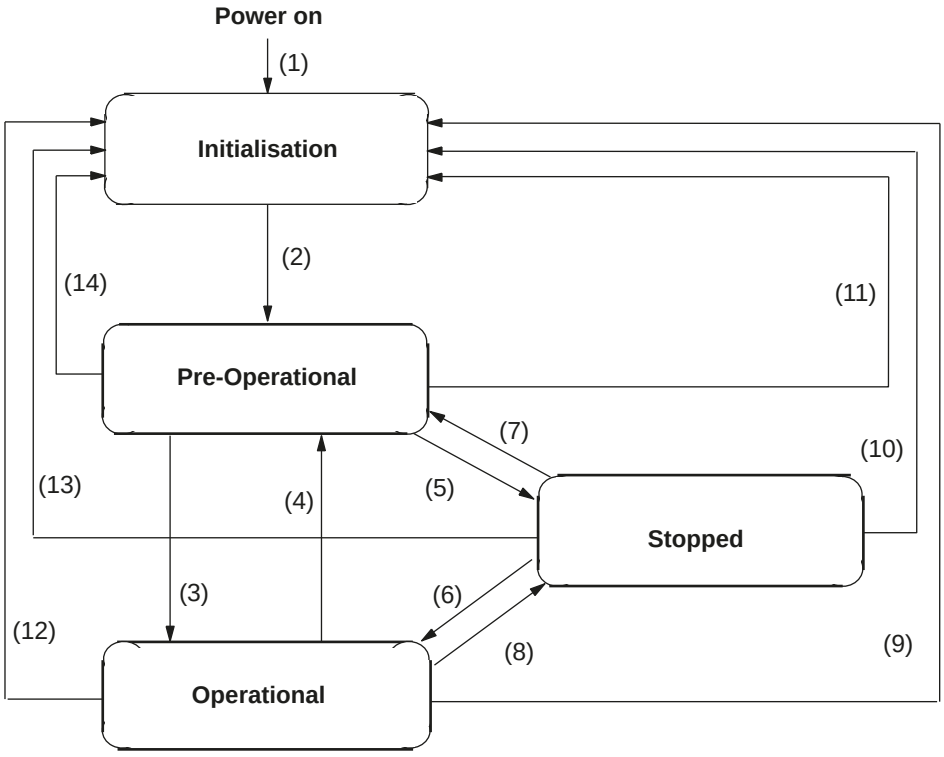


Figure 1-6 State diagram of a CANopen node

Note

In the "Pre-Operational" state, only SDO communication is possible; PDO communication is inhibited.

In the "Operational" state, PDO communication also takes place.

The diagnostics LED -> green (CANopen RUN LED) indicates the status.

NMT services

The following table lists the NMT services that are available for controlling the status transitions.

Table 1- 11 Transitions in the status diagram

Transitions	Services
(1)	After POWER ON, the Control Unit automatically switches to "Initialization".
(2)	After initialization, it switches to "Pre-Operational".
(3), (6)	Start_Remote_Node command (CS = 1)
(4), (7)	Enter_Pre-Operational_State command (CS = 128)
(5), (8)	Stop_Remote_Node command (CS = 2)
(9), (10), (11)	Reset_Node command (CS = 129)
(12), (13), (14)	Reset_Communication command (CS = 130)

The NMT services have the following functions:

- Start Remote Node
  Command for switching from the "Pre-Operational" communication state to "Operational". The drive can only transmit and receive process data in the "Operational" state.
- Stop Remote Node:
  command for switching from "Pre-Operational" to "Stopped" or from "Operational" to "Stopped". The node can only process NMT commands in the "Stopped" state.
- Enter Pre-Operational:
  command for switching from "Operational" or "Stopped" to "Pre-Operational". In the "Pre-Operational" state, the node cannot process any PDO. It can, however, be parameterized or operated via SDOs, which means that setpoints can also be specified.
- Reset Node:
  command for switching from "Operational", "Pre-Operational", or "Stopped" to "Initialization". When the "Reset Node" command is issued, all the objects (1000 hex - 9FFF hex) are reset to the state that was present after "Power On".
- Reset Communication:
  command for switching from "Operational", "Pre-Operational", or "Stopped" to "Initialization". When the "Reset Communication" command is issued, all the communication objects (1000 hex - 1FFF hex) are reset to their original state.

NMT protocols

For the protocols shown below, the following generally applies:

The NMT master can direct a request to several NMT slaves simultaneously.
The protocol diagrams are valid for data exchange between an NMT master and an NMT slave. For the COB identifier the following is valid for all protocols: COB-ID = 0.

Node-ID: 1-127, Node-ID = 0: All nodes

Table 1- 12 COB-ID = 0

NMT Master Request ----> NMT Slave message		
Command	**Byte 0**	**Byte 1**
Start	cs = 1 (01 hex)	Node ID
Stop	cs = 2 (02 hex)	Node ID
Enter Pre-Operational	cs = 128 (80 hex)	Node ID
Reset Node	cs = 129 (81 hex)	Node ID
Reset Communication	cs = 130 (82 hex)	Node ID

cs: NMT command specifier

NMT state after power up

As an extension to the NMT services, automatic switching to the "Operational" state after POWER ON can also be achieved using parameter settings (see "State diagram of a CANopen node").

Using parameter p8684 , the CANopen-NMT state is set, which is effective after booting or as result of the NMT service "Reset Node" or "Reset Communication".

Possible values:

- 4: Stopped
- 5: Operational
- 127: Pre-Operational (factory setting).

The NMT state "Pre-Operational" is selected in the factory setting, as this also corresponds to the CANopen standard.

The CANopen NMT state can be displayed or the desired state set using the parameter p8685.

1.14 CANopen device state machine

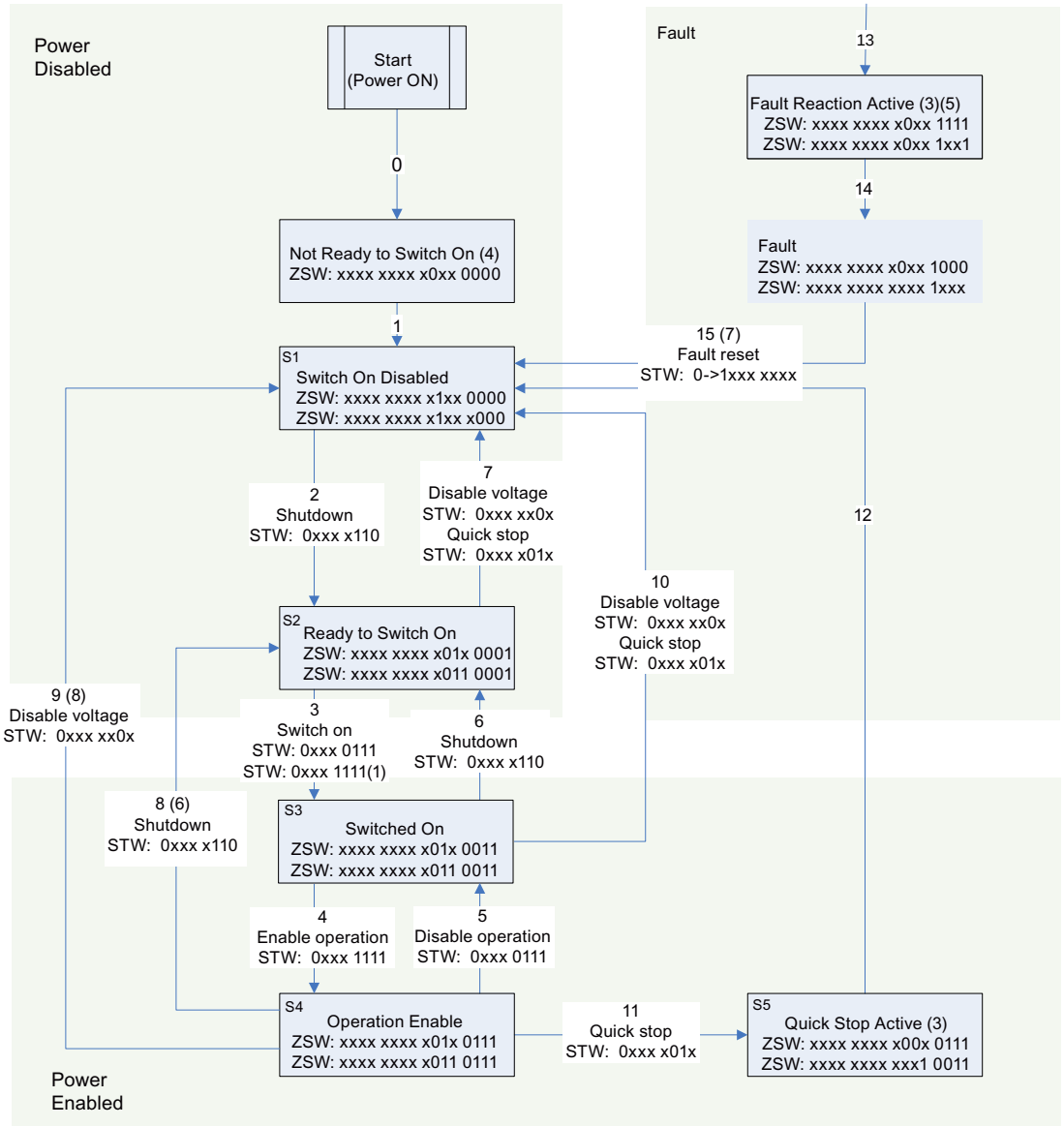
The CANopen device state machine describes the device status and the possible drive device status transitions. Each individual status describes a particular internal or external behavior. Depending on the drive device status, only certain transition commands are accepted.

The drive device status is changed using a defined control word and/or corresponding to an internal event. The current status can be read from the status word of the CANopen.

Table 1- 13 Status of the CANopen device machine

Status	Condition
Not Ready to Switch On	Low power is present at the drive. The drive is initialized. The drive function is deactivated.
Switch On Disabled	Drive initialization has been completed. The drive parameters may be changed. The drive function is deactivated. It is not permissible that a high voltage is connected to the drive.
Ready to Switch On	The drive parameters may be changed. The drive function is deactivated. It is permissible that a high voltage is connected to the drive.
Switched On	The drive parameters may be changed. The drive function is deactivated. A high voltage was connected to the drive. The current amplifier is ready.
Operation Enable	No faults were identified. The drive parameters may be changed. The drive-function is activated and the motor is supplied with current. This corresponds to normal drive operation.
Quick Stop Active	The quick stop function is executed. The drive parameters may be changed. The drive-function is activated and the motor is supplied with current.
Fault Reaction Active	A non-fatal fault has occurred at the drive. The quick stop function is executed. The drive parameters may be changed. The drive-function is activated and the motor is supplied with current.
Fault	The drive parameters may be changed. A fault has occurred at the drive.

States of the CANopen device state machine and their mapping on SINAMICS drives



(1) Transition via "Switched On" in "Operation Enable"
(2) Transition is only permitted if "Quick stop option code" 0x605A = 5. However, the optional object is not supported.
(3) The state is only recognized if PZD sampling time p8848 <= 2 ms, since the underlying SINAMICS sequence control is calculated in 2 ms.

(4) Internal state, not visible in CANopen ZTW, since no CANopen communication is possible at this time.
(5) The state will not run if a fault appears in "Switch On Disabled".
(6) Automatic switching via "Switched On" to "Ready to Switch On", only visible for CBC10 sampling time p8848 <= 2ms.
(7) Automatic switching via "Switch On Disabled" to "Ready to Switch On" with the command "Fault Reset" and "Shutdown", only visible for CBC10 sampling time p8848 <= 2 ms.
(8) Display of state xx03hex (for STW=0hex) or xx27hex (for STW=27hex) for one cycle, as the power unit displays the actual state, only visible for CBC10 sampling time p8848 <= 2ms.

Sx = PROFIdrive state numbers
ZSW = Status word according to CANopen (Object 6041)
ZSW = Status word for SINAMICS
STW = Control word (Object 6040)

Figure 1-7 CANopen device state machine

1.15 Fault monitoring

Fault monitoring services

The fault monitoring services are used to find errors within the CAN network.

Local faults within the device that, for example, lead to a reset or state change are not affected.

The fault monitoring services are based on CANopen device messages sent out periodically. There are two options for fault monitoring.

Note

For effective fault monitoring, one of the two methods must be activated – either "Life Guarding" or "Heartbeat". Both functions are deactivated in the basic setting.

Life guarding

The NMT master Issues monitoring queries using the node guarding protocol. If one of the NMT slaves addressed does not reply within a certain time – the "Node lifetime" – or the state of the NMT slave has changed, then the NMT master informs its master application.

If the "Life guarding" method (NMT slave monitors the NMT master) is supported by the NMT slave, it uses the entries for "Guard time" and "Lifetime factor" from its object directory to determine its "Node lifetime". If the NMT slave is not addressed/monitored by the NMT master within its "Node lifetime", it informs its local application of this using a "Life guarding event". If the entries for "Guard time" and "Lifetime factor" in the slave's object directory are at "0" (default setting), then the NMT slave does not monitor the NMT master.

Monitoring of the NMT slave starts when the first "Remote transmit request" (RTR) is received from the NMT master via its COB ID. This usually happens during the bootup phase.

Heartbeat

A "Heartbeat producer" (CANopen device) issues heartbeat messages periodically. One or several CANopen devices in the network recognize this heartbeat message. If a heartbeat cycle of the "Heartbeat producer" is missing, a Heartbeat Consumer can respond to this.

Fault monitoring protocols

Node guarding protocol

The NMT master queries each NMT slave at regular time intervals (Node guard time). This time interval can be different for each NMT slave. The slave's answer contains information about its state. The "Node lifetime" results from the "Node guard time" multiplied by the "Lifetime factor" and can be different for each NMT slave. If the NMT slave is not queried by the NMT master during its "Node lifetime", a "remote node error" is displayed in the NMT slave using a "Life guarding event".

A "Node guarding event" is created in the following cases:

- The RTR ("Remote transmission request") is not confirmed within the "Node lifetime"
- The state reported by the NMT slave does not match the state expected by the NMT master.

In the case of a CAN communication error, e.g. too many telegram failures, fault F08700 is signaled (for details see SINAMICS S120/150 List Manual). The fault is displayed in parameter r0949. The reaction of the drive to the fault is set with p8641.

The values for the "Node guard time" and the "Lifetime factor" are stored in the appropriate NMT slave's object directory.

Table 1- 14 Node guarding protocol COB-ID = 700h + Node-ID

Time	NMT Master	Communication		NMT Slave
Node lifetime	Inquiry ---->	Remote transmit request		----> Signaling
	Confirmation <----	**Byte 0, bit 7**	**Byte 0, Bit 6..0**	<---- Response
	Node guard time	**t**	**s**	
	Inquiry ---->	Remote transmit request		----> Signaling
	Confirmation <----	**Byte 0, bit 7**	**Byte 0, Bit 6..0**	<---- Response
		t	**s**	
	Signaling after expiry of "Node lifetime": "Node guarding event" **			Signaling after expiry of "Node lifetime": "Life guarding event" **

Code:

** for monitoring faults

s Status of the NMT slave

t toggle bit

The value of this bit should change between two consecutive responses of the NMT slave. The toggle bit value for the first response after the "Guarding protocol" has been activated must be 0. The toggle bit in the "Guarding protocol" should be reset to 0 if the NMT state "Reset communication" was run (no other status of an NMT state resets the toggle bit). If a response is received with the same value for the toggle bit as in the previous response, then the new response is treated as if it was not received.

Heartbeat protocol

The heartbeat protocol is a fault monitoring service without transmission of RTR signals (RTR = Remote transmit request).

A "heartbeat producer" transmits a heartbeat message cyclically. One or several "heartbeat consumers" receive this message. The relationship between "producer" and "consumer" is controlled via the object directory.

The "heartbeat consumer" monitors the receipt of the "heartbeat" within a time period, the "heartbeat consumer time". If the "heartbeat" is not received in the "heartbeat consumer" within the "heartbeat consumer time", a "heartbeat event" is triggered. The SINAMICS drive is only a "heartbeat producer".

Table 1- 15 Heartbeat protocol COB-ID = 700h + Node-ID

Heartbeat producer	Data byte 0		Heartbeat consumer
Inquiry ---->	**Byte 0, bit 7**	**Byte 0, Bit 6..0**	----> Signaling
| Heartbeat producer time |	**r**	**s**	| Heartbeat consumer time |
Inquiry ---->	**Byte 0, bit 7**	**Byte 0, Bit 6..0**	----> Signaling
	r	**s**	| Heartbeat consumer time |
			After expiry of the heartbeat consumer time: Heartbeat event

Code:

r reserved (always 0)

s Status of the "heartbeat producer"

If a "heartbeat producer time" is configured (object number 1017), the heartbeat protocol starts immediately. Here state changes from the "Initialization" state to the "Pre-Operational" state are possible.

In this case a boot-up message is regarded as the first heartbeat message.

Note

Node guarding protocol and heartbeat protocol cannot be used at the same time. As soon as the "heartbeat producer time" is not equal to zero, the heartbeat protocol is used automatically.

1.16 Save parameters restore factory settings

Parameters can be saved and the factory settings restored using the following Control Unit communication objects.

- Parameter save -> communication object 1010 hex
- Restore factory settings -> communication object 1011 hex

Parameter save (object 1010 hex).

- Subindex 0: (1010.0):

  The number of subindices of this object are shown in this subindex.

- Subindex 1: (1010.1):

  By writing the ASCII character sequence "evas" - which corresponds to the hexadecimal value "65 76 61 73" - to this subindex, all of the drive parameters are saved in the non-volatile memory (memory card).

  This corresponds to writing a "1" to drive parameter p0977.

- Subindex 2: (1010.2)**:

  By writing the ASCII character sequence "evas" - which corresponds to the hexadecimal value "65 76 61 73" - to this subindex, only the communication objects (objects of the number range 1000 hex - 1FFF hex) of the drive are saved in the non-volatile memory (memory card).

- Subindex 3: (1010.3)**:

  By writing the ASCII character sequence "evas" - which corresponds to the hexadecimal value "65 76 61 73" - to this subindex, only the application objects (objects of the number range 6000 hex - 9FFF hex) of the drive are saved in the non-volatile memory (memory card).

  The parameters of the communication objects are saved in the CCxxxxxn.ACX files in the directory \USER\SINAMICS\DATA\ on the memory card.

  The parameters of the application objects are saved in the CAxxxxxn.ACX files in the directory \USER\SINAMICS\DATA\ on the memory card.

  "n" corresponds to the drive object ID to which the parameters belong.

** A partial data save (subindex 2 or 3) is only possible if previously a complete data save was performed (subindex 1).

When reading subindices 1...3, a value of 1 is obtained, this has the following significance: Device saves parameters using a write access of the object.

Restoring the factory setting of the parameters (object 1011 hex)

- Subindex 0: (1011.0):

  The number of subindices of this object are shown in this subindex.

- Subindex 1: (1011.1):

  By writing the ASCII character sequence "daol" - which corresponds to the hexadecimal value "64 61 6F 6C" - to this subindex, all of the drive parameters are reset to their original factory setting.

  This corresponds to writing a "1" to drive parameter p0976.

- Subindex 2: (1011.2):

  By writing the ASCII character sequence "daol" - which corresponds to the hexadecimal value "64 61 6F 6C" - to this subindex, only the communication objects (objects of the number group 1000 hex - 1FFF hex) of the drive are reset to their factory setting.

- Subindex 3: (1011.3):

  By writing the ASCII character sequence "daol" - which corresponds to the hexadecimal value "64 61 6F 6C" - to this subindex, only the application objects (objects of the number group 6000 hex - 9FFF hex) of the drive are reset to their factory setting.

When reading subindices 1...3, a value of "1" is obtained, this has the following significance: The device sets the parameters back to their factory setting via write access of the object.

Note

The factory setting value is, for the corresponding parameters, effective after a successful write operation. According to CANopen, it should only be effective after the NMT commands "Reset Node" or "Reset Communication" or after switching on/off.

1.17 CAN bus sampling time

The sampling time of the CBC10 can be set with the parameter p8848 "IF2 PZD sampling time".

In the factory setting, the sampling time is 4 ms. Asynchronous message frames can be received and sent within a period of 4 ms.

Cycle time

- For cyclic receive telegrams, the cycle time, according to the Shannon sampling theorem, must be greater than at least twice the sampling time. If the cycle time is long enough, no receive message frame is lost and alarm A8751 is not output.

  Example: The SYNC cycle should be 3 ms. The setting in p8848 =1 ms. So the cycle time is larger than double the sampling time.

- For receive message frames, whose data does not simultaneously change according to the Shannon sampling theorem faster than twice the sampling time, then the cycle time can be increased. Alarm A8751, which is output when the cycle time is increased, can be suppressed by changing over the message type to "No message" using parameters p2118, p2119.

Setting the CAN bus sampling time using p8848:

- Set the device commissioning parameter filter p0009 = 3 (drive basis configuration).
- Change and save parameter p8848.
- The changes become active immediately.

1.18 Number of controllable drives via CAN bus

Depending on the request to the drives, up to four axes can be controlled simultaneously via the CAN bus, where, for CANopen a maximum of four transmit message frames and four receive message frames can be used for communication. If all eight of the maximum possible communication message frames are used for drives, only three drives can be controlled -> 3 * 8 = 24.

The maximum possible number of drives is limited as follows:

1. By the requirements placed on the drives (e.g. slave drives frequently require less than eight message frames)
2. By the maximum number of communication message frames, namely 25.

Servo control

- A maximum of four axes can be controlled.

Vector control

- A maximum of two axes can be controlled.

Prerequisites for commissioning

2

This section describes the commissioning requirements:

- CBC10
- STARTER commissioning tool

Note

All the parameters, faults, alarms, and function diagrams for CANopen in the SINAMICS drive line-up are described in the SINAMICS S120/S150 List Manual.

2.1 Prerequisites for commissioning

To commission a CAN bus in a SINAMICS drive line-up, the following hardware and software components are required:

- SINAMICS S120:
  - CBC10
  - Memory card with firmware
- Connecting the Control Unit Ethernet interface to a PC/PG with Ethernet interface
- STARTER commissioning tool on the PG/PC.

Note

For descriptions of the components in a SINAMICS drive line-up, the wiring, the PROFIBUS interface to a PC/PG and installing the STARTER commissioning tool, please refer to the manuals SINAMICS S120 Manual, Control Units and supplementary system components and SINAMICS S120 Manual, Power Units Booksize, as well as the SINAMICS S120 Commissioning Manual.

2.2 CBC10 for CAN bus

The CBC10 is used to connect drives of the SINAMICS S120 drive system to higher-level automation systems with a CAN bus.

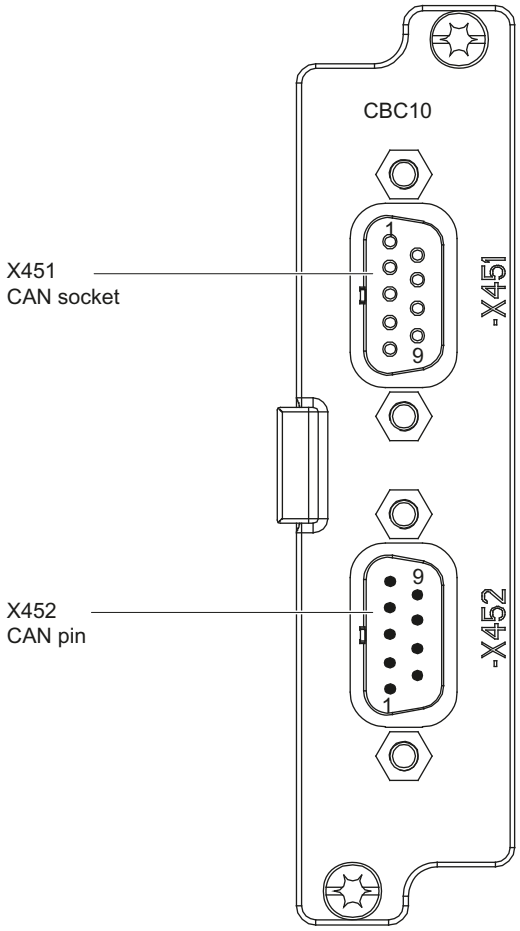


Figure 2-1 View of CBC10

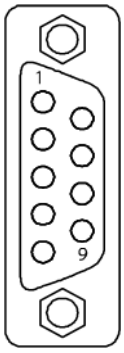
The CBC10 uses two 9-pin sub D connectors for the connection to the CAN bus system.

The connectors can be used as inputs or outputs. Unused pins are plated through.

The following baud rates (among others) are supported: 10, 20, 50, 125, 250, 500, 800 kBaud, and 1 Mbaud.

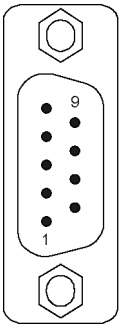
The X451 CAN bus interface has the following socket assignment.

Table 2- 1 CAN BUS interface X451

	Pin	Designation	Technical data
	1	Reserved	
	2	CAN_L	CAN signal (dominant low)
	3	CAN_GND	CAN ground
	4	Reserved	
	5	CAN_SHLD	Optional shield
	6	GND	CAN ground
	7	CAN_H	CAN signal
	8	Reserved	
	9	Reserved	
Type: 9-pin SUB-D female			

The X452 CAN bus interface has the following socket assignment.

Table 2- 2 CAN BUS interface X452

	Pin	Designation	Technical data
	1	Reserved	
	2	CAN_L	CAN signal (dominant low)
	3	CAN_GND	CAN ground
	4	Reserved	
	5	CAN_SHLD	Optional shield
	6	GND	CAN ground
	7	CAN_H	CAN signal
	8	Reserved	
	9	Reserved	
Type: 9-pin SUB-D male			

2.2.1 CBC10 Installation

Mounting and installation steps

The CBC10 is installed in the option slot on Control Unit CU320-2 as follows (see diagram below):

1. Unscrew and remove the protective cover.
2. Insert the CBC10.
3. Fix the CBC10 in place.

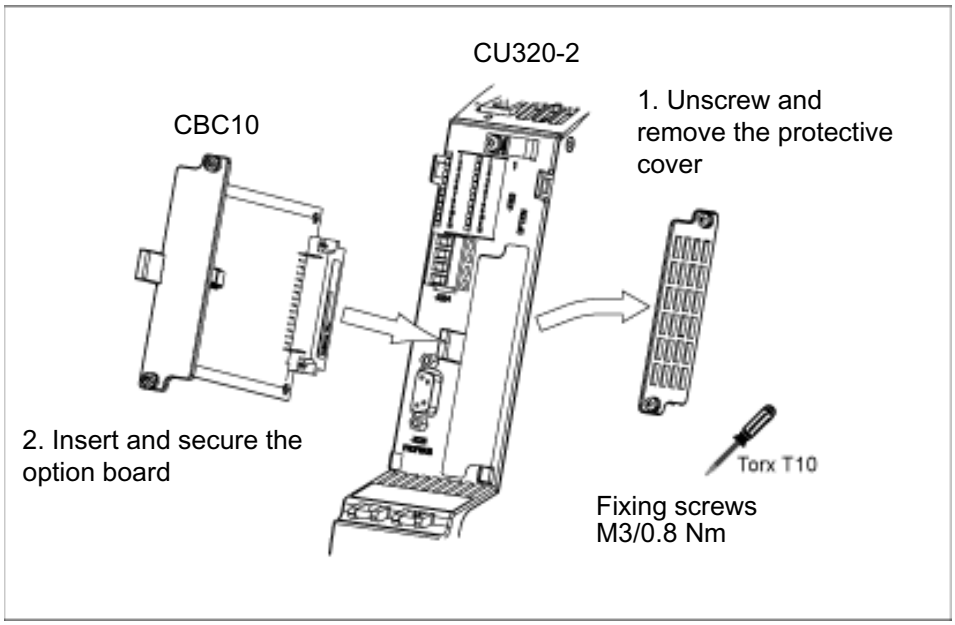


Figure 2-2 Mounting the CBC10

2.2.2 CANopen functionality

The CBC10 supports CANopen transfer types with SDO (service data objects) and PDO (process data objects).

The CBC10 also supports free PDO mapping.

The CBC10 supports CANopen communication profile DS 301 version 4.0, device profile DSP 402 (drives and motion control) version 2.0, and indicator profile DR303-3 version 1.0.

For communication monitoring purposes, the CBC10 supports node guarding and the heartbeat protocol (heartbeat producer).

The CBC10 features an SDO server that can be used to read or write all the SINAMICS parameters.

The CBC10 firmware supports the following modes:

- Profile velocity mode
- Profile torque mode
- Velocity mode

Node guarding/life guarding

SINAMICS waits a certain time (node lifetime) for message frames from the master application and permits a specific number (lifetime factor) of failures within a specified time interval (node guard time).

The node lifetime is calculated by multiplying the node guard time by the lifetime factor.

Heartbeat protocol

SINAMICS (producer) cyclically transmits (heartbeat time) its communication status (sign of life) on the CAN bus to the master application.

Profile velocity mode

This mode allows velocity setpoints to be defined and set according to variable velocity profiles.

Profile torque mode

This operating mode supports the closed-loop torque control and the objects intended for this purpose.

Velocity mode

This operating mode supports simple velocity control (open loop) with ramps and the objects intended for this purpose.

See also

Operating modes (Page 129)

2.2.3 Diagnostics LED "OPT"

The following diagnostics LED "OPT" on the Control Unit CU320-2 indicates the status of the CANopen node at the device.

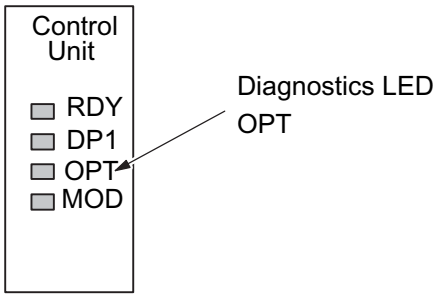


Figure 2-3 Overview of the LEDs on the Control Unit

The diagnostics LED "OPT" on the Control Unit, which displays both the module and communication status, provides users with all the required information about the current status of the CBC10.

General procedure

The different flashing frequencies indicate the following:

- Diagnostics LED OPT -> red
  Is there a fault?
- Diagnostics LED OPT -> green
  In which state is the node in the communication state machine?

Diagnostics LED OPT -> red

Table 2- 3 Diagnostics LED OPT -> red (CANopen error LED)

ERROR LED flashing frequency	Status	Meaning
Off	No error	Ready to run
Single flash	Warning limit reached	At least one of the CAN controller error counters has reached the "Error Passive" alarm threshold (too many incorrect telegrams).
Double flash	Error Control Event	A guard event has occurred.
On	Bus off	The CAN controller is "Bus off".

Diagnostics LED OPT -> green

Table 2- 4 Diagnostics LED -> green (CANopen RUN LED)

ERROR LED flashing frequency	Status	Meaning
Single flash	Stopped	The node is in the "STOPPED" state.
Blinking	PRE-OPERATIONAL	The node is in the "PRE-OPERATIONAL" state.
On	OPERATIONAL	The node is in the "OPERATIONAL" state.

2.3 STARTER commissioning tool

Calling the STARTER application

1. Click on the STARTER symbol  of your user interface.

   Or

2. Call the menu command "Start > SIMATIC > STEP 7 > STARTER" in your Windows Start menu.

2.3.1 BICO interconnection procedure in STARTER

You can parameterize the drive settings in the OFFLINE mode via STARTER by means of BICO interconnection. Parameterization can be carried out via the following means:

- Expert list
- Graphical screen interface

The steps described below explain the general BICO interconnection procedure in the STARTER commissioning tool.

BICO interconnection in the expert list

When carrying out BICO interconnection via the expert list, proceed as follows:

You want to interconnect parameter p0840 of the control word with parameter r8890[0], for example.

1. In the project navigator, select for example "Drive_1", and call the expert list via the shortcut menu "Expert list"
2. Search for parameter p0840.

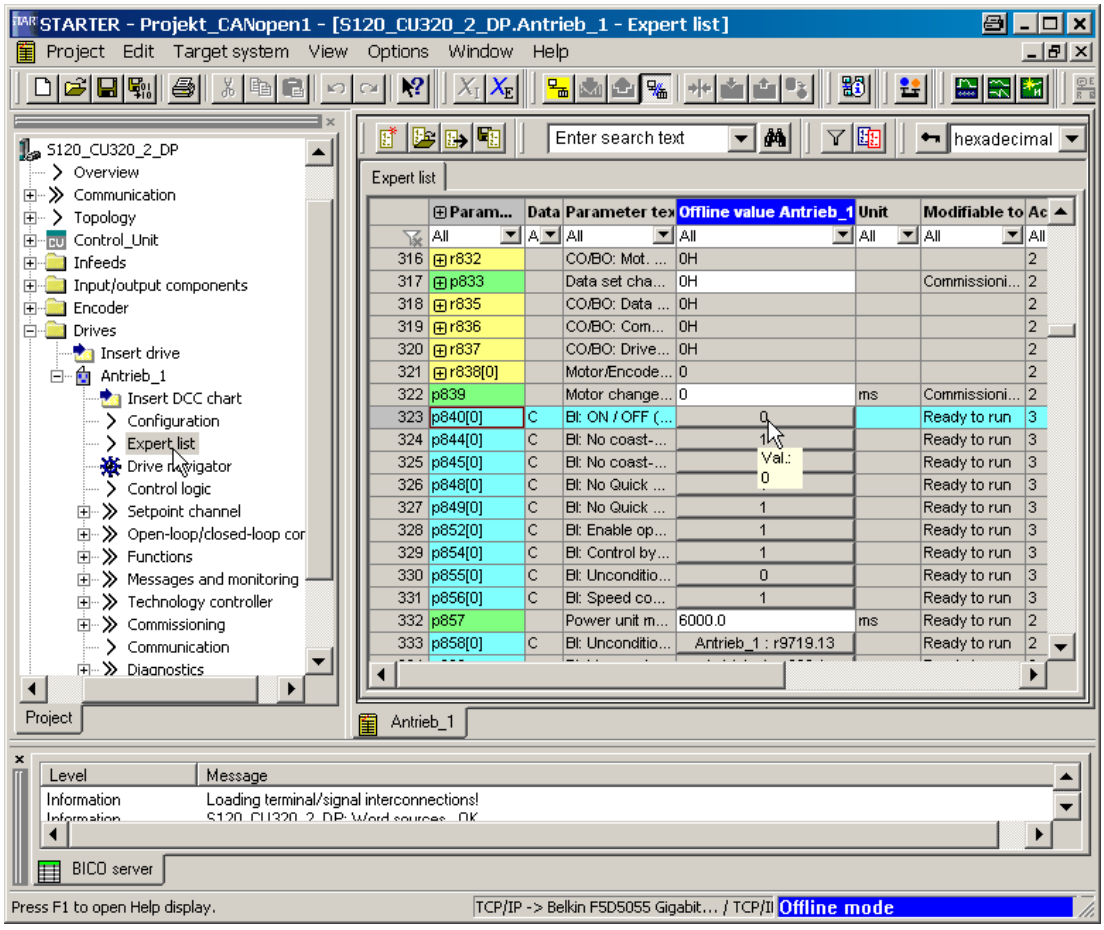


Figure 2-4 Interconnect 1

3. Click the button to interconnect with a parameter.

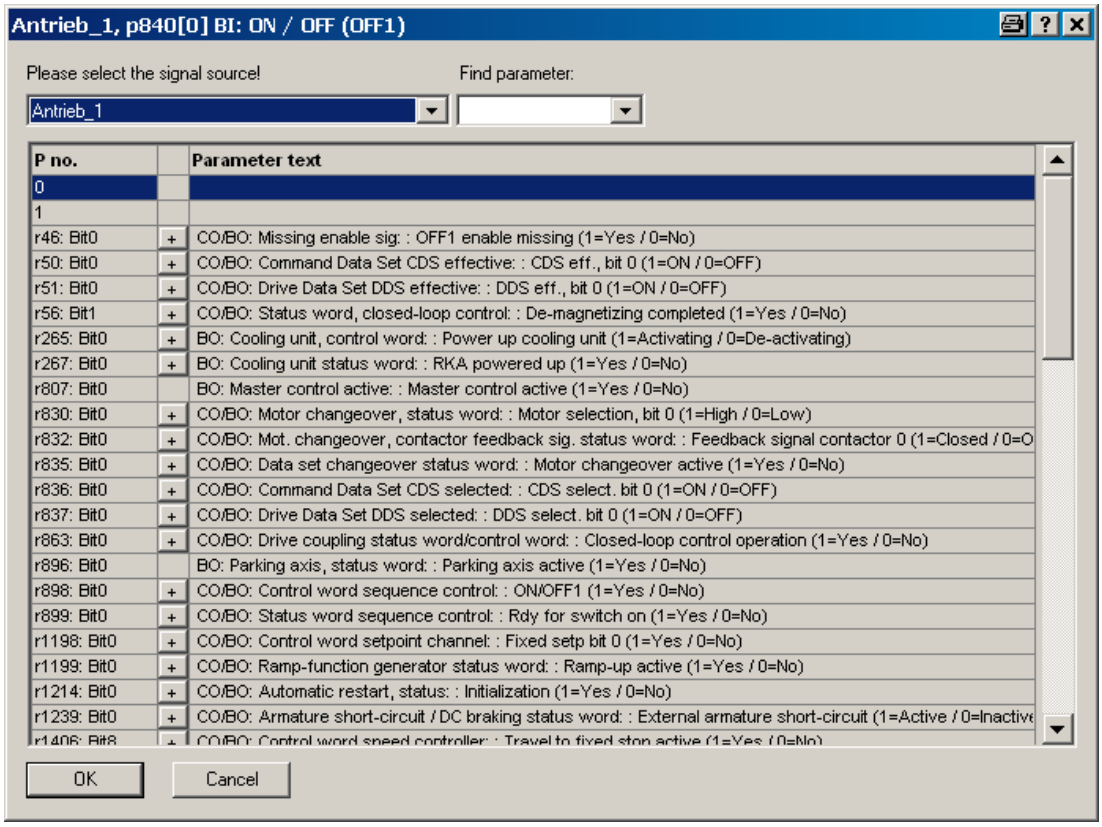


Figure 2-5 Interconnect 2

A list from which you can select the available r parameters is now displayed.

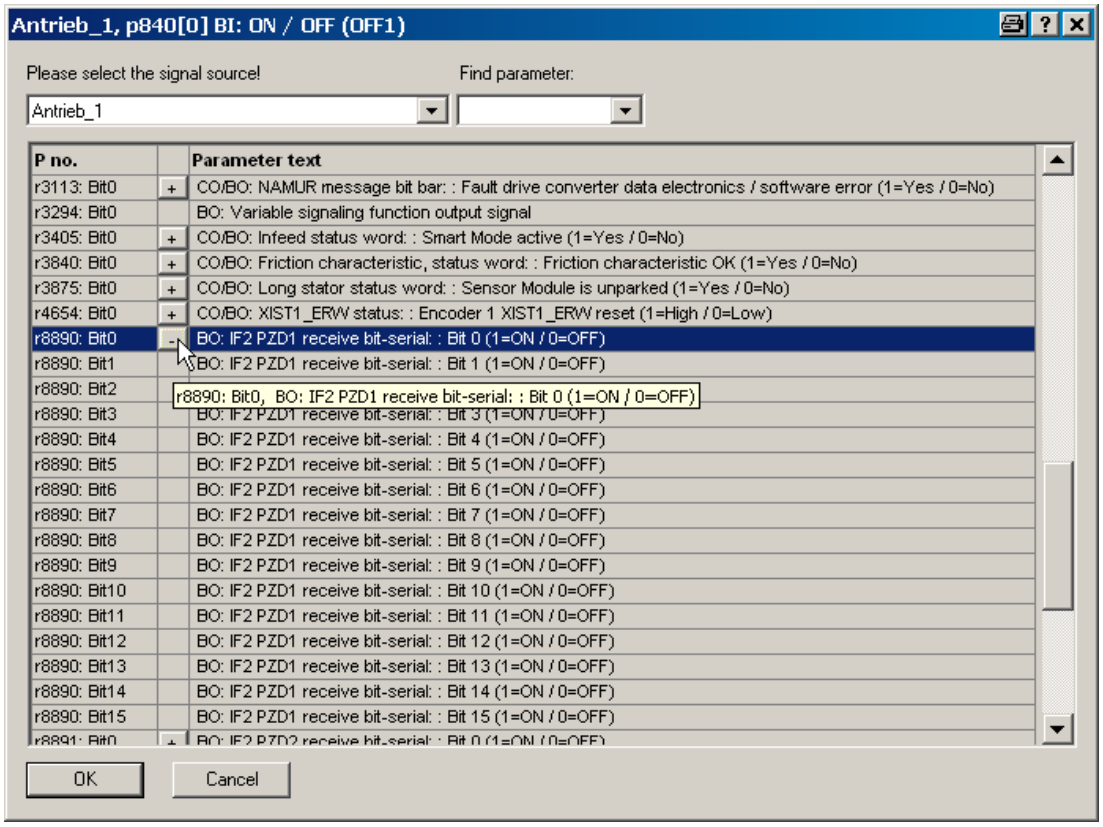


Figure 2-6 Interconnect 3

4. Click the plus symbol of parameter r8890:bit0.

5. Double-click on r8890: Bit0.

   In the expert list, you can now see that p0840 has been interconnected with parameter r8890[0].

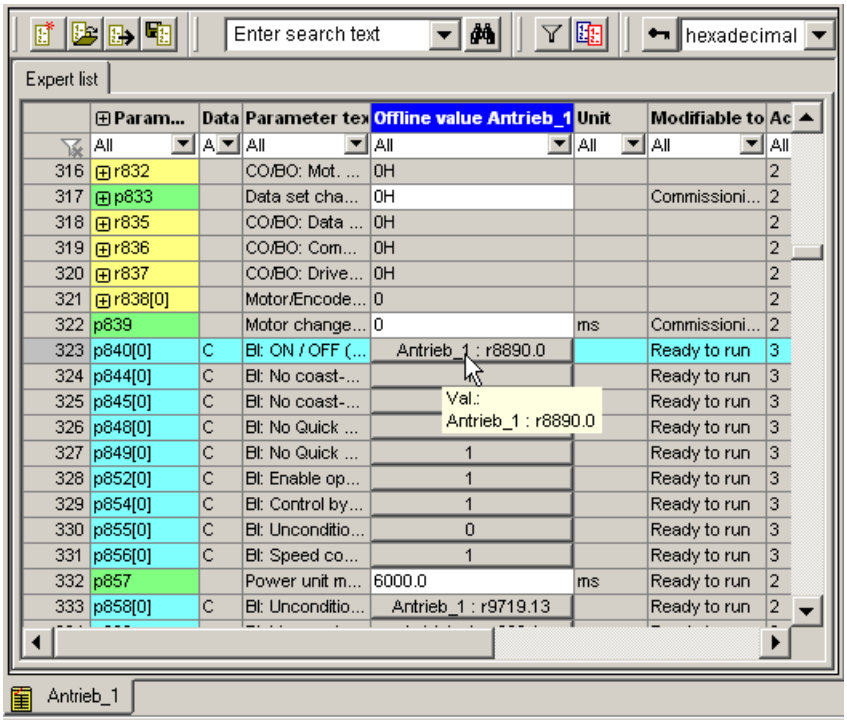


Figure 2-7 Interconnect 5

Graphical screen interface

When carrying out BICO interconnection via the graphical screen interface, proceed as follows:

For the setpoint velocity, which is a 32-bit data type, you want to interconnect parameter p1155[0] for "Speed setpoint 1" with parameter r8860[1], for example.

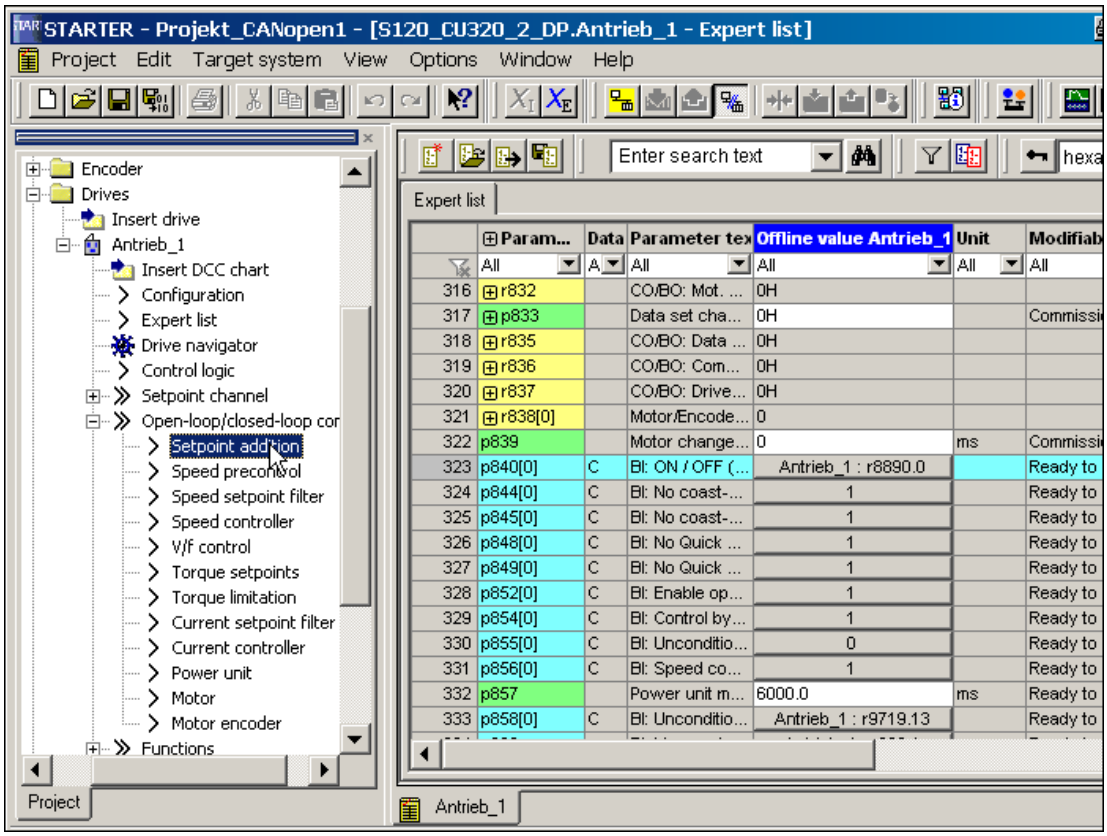


Figure 2-8 Interconnecting screens 1

1. In the project navigator, double-click the "Setpoint addition" selection under "Drive_1 > Speed control".

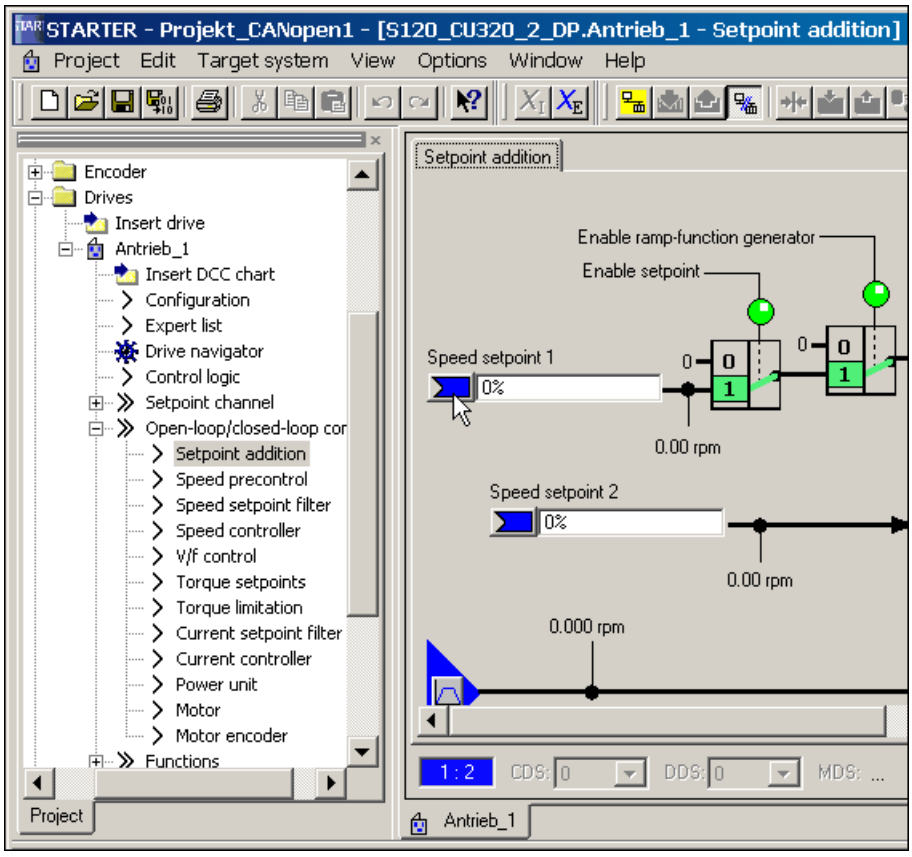


Figure 2-9 Interconnecting screens 2

2. Click the blue field to the left of the field for "Speed setpoint 1" and then click the selection "Further interconnections", which is now displayed.

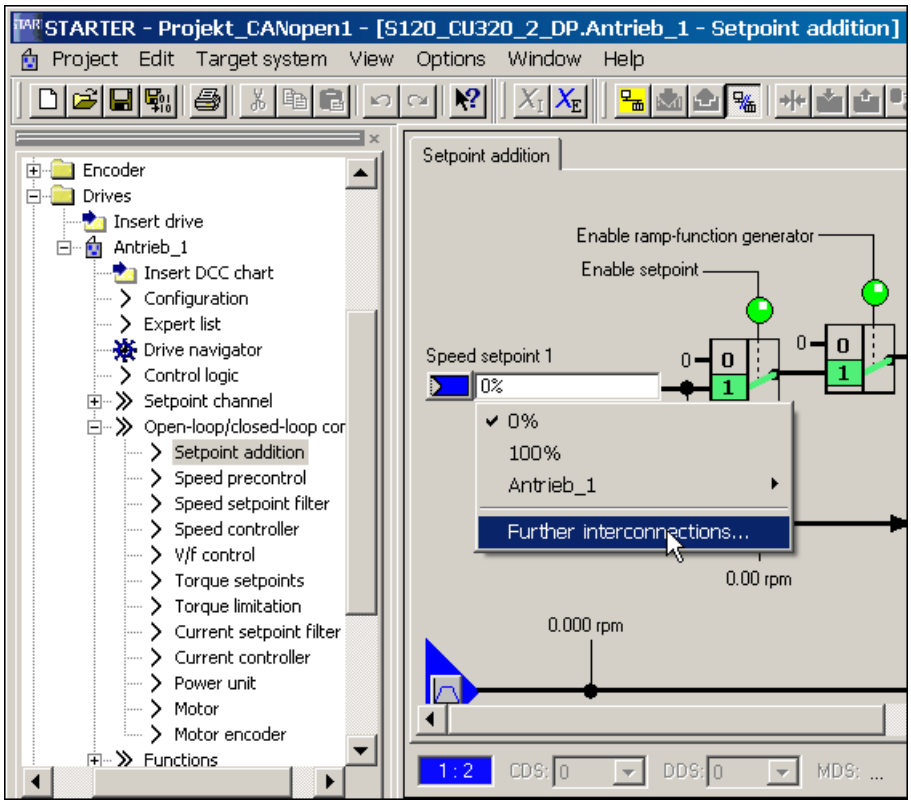


Figure 2-10 Interconnecting screens 3

A list from which you can select the available r parameters is now displayed.

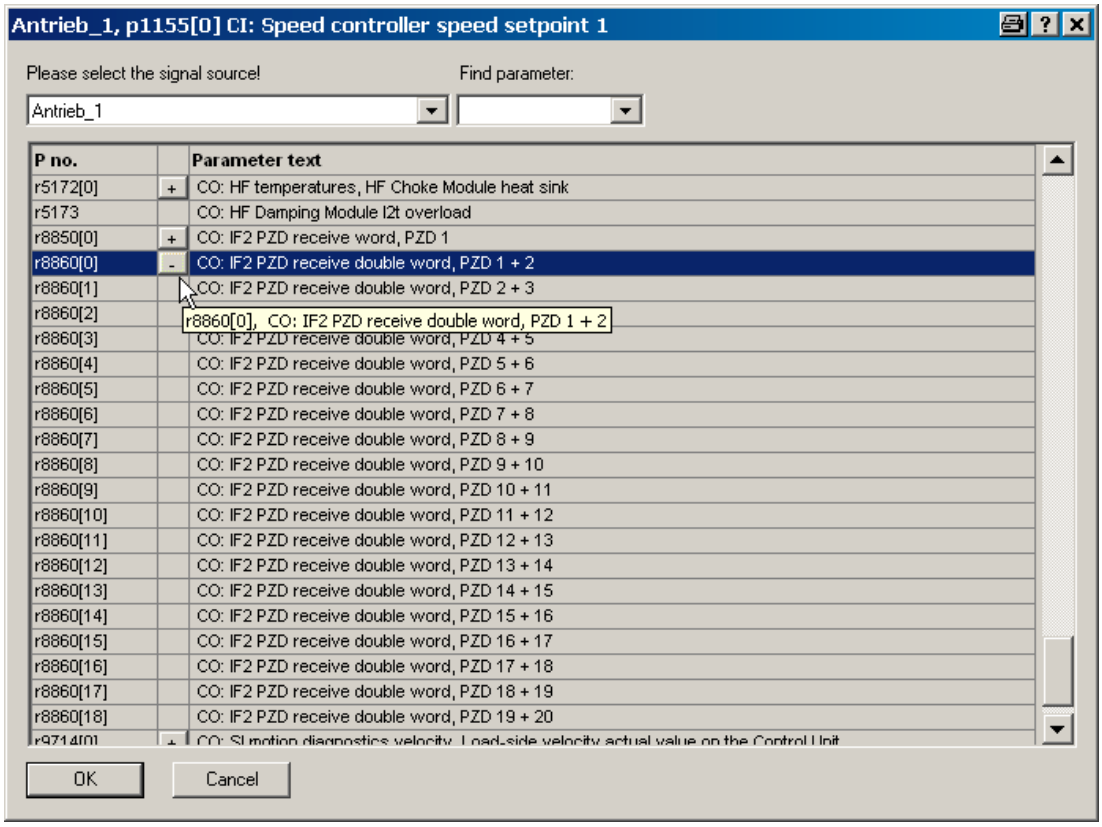


Figure 2-11 Interconnecting screens 4

3. Double-click on r8860: Bit1

In the graphic screen interface, you can now see that p1155 has been interconnected with parameter r8860[1] .

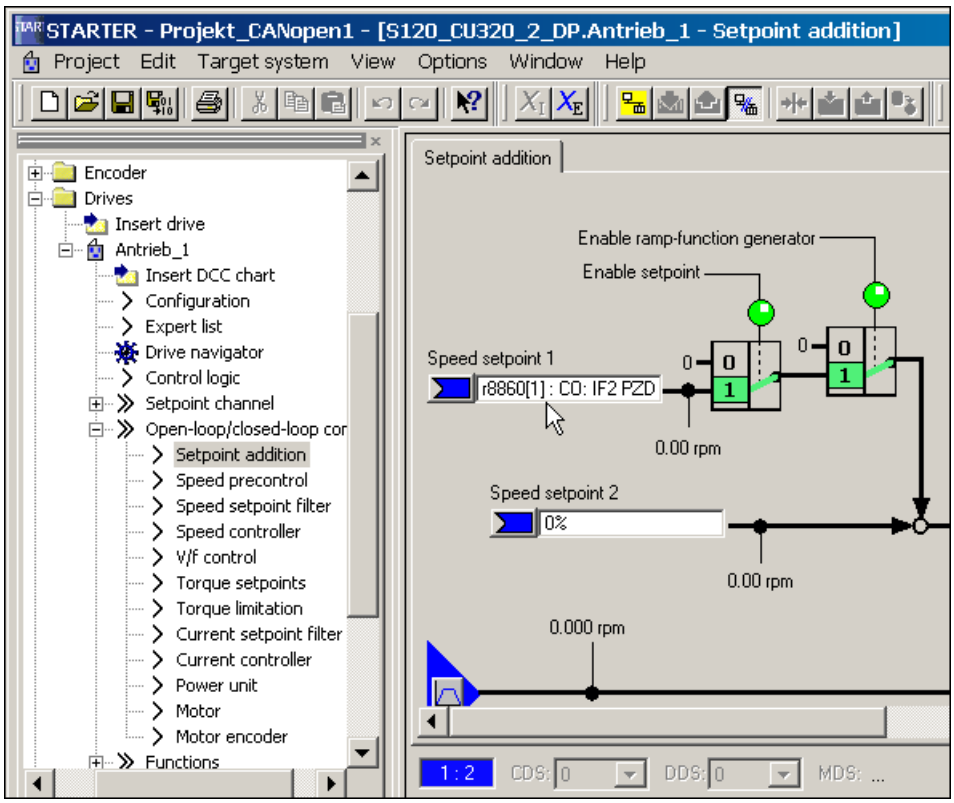


Figure 2-12 Interconnecting screens 5

See also

BICO interconnection in conjunction with CANopen (Page 34)

Interconnecting process data in the receive and transmit buffers (Page 100)

Commissioning

3

Section content

This section shows you how to carry out initial commissioning for the CANopen interface in the SINAMICS drive line-up using the STARTER commissioning tool.

This section first looks at the initial commissioning procedure and describes a sample configuration.

The initial commissioning procedure described here takes place with STARTER commissioning tool in the ONLINE mode. At the end of each step, notes are included (if necessary) to explain how the procedure differs in OFFLINE mode.

Requirements

Before carrying out the commissioning steps described in this section, make sure you have read the section "Prerequisites for commissioning (Page 49)".

3.1 Initial commissioning procedure

Initial commissioning

The following steps must be carried out during the initial commissioning procedure for the CANopen interface in the SINAMICS drive line-up:

1. Making hardware settings on the CBC10.
2. Configure the drive unit using the STARTER commissioning tool in the ONLINE mode.
3. Configure the COB IDs and process data objects for the receive and transmit message frames.
4. Interconnect the receive and transmit buffers.
5. In the ONLINE mode, load the projects from the drive unit to the PC/PG and save.

3.2 Example configurations

Note

The commissioning procedure described here is based on the sample configuration shown in a SINAMICS S120 drive line-up for the following modules:

- one infeed (Active Line Module)
- two drives (Single Motor Modules)

For a more detailed description of this configuration, refer to the section "Initial commissioning servo control, booksize format" in the SINAMICS S120 Commissioning Manual.

This Commissioning Manual supplements the description of "Initial commissioning servo control booksize format" to include a description of the initial commissioning procedure for the CANopen communication interface with the "CBC10".

Overview

The following overview diagram illustrates the drive objects.

CANopen configuration example: Assignment of the modules to the SINAMICS drive object

- Single Motor Module 1 is the first SINAMICS drive object
- Single Motor Module 2 is the second SINAMICS drive object

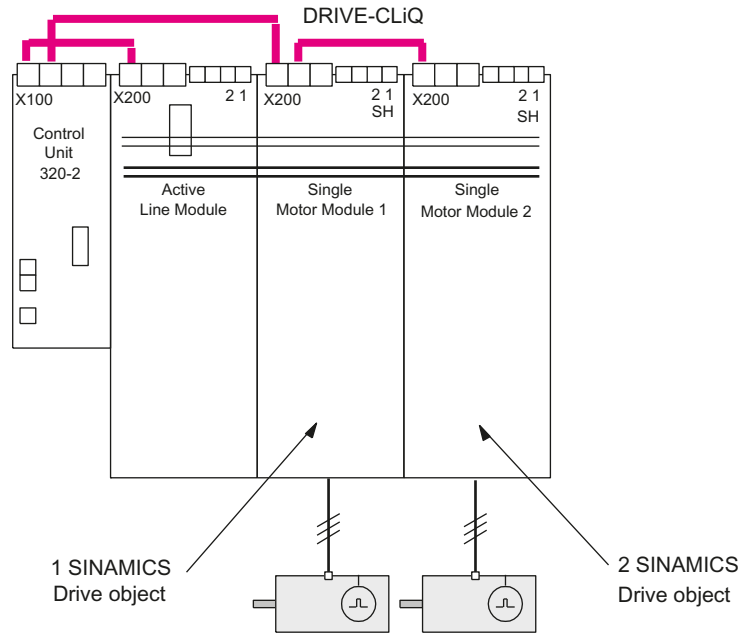


Figure 3-1 SINAMICS S120 components (example)

3.3 Making the hardware settings on CBC10

To ensure that data can be transferred reliably via the CAN bus, switches S1/S2 on the CBC10 must be set accordingly (see table below). Set the following:

- Bus terminating resistor
- Operation with/without ground

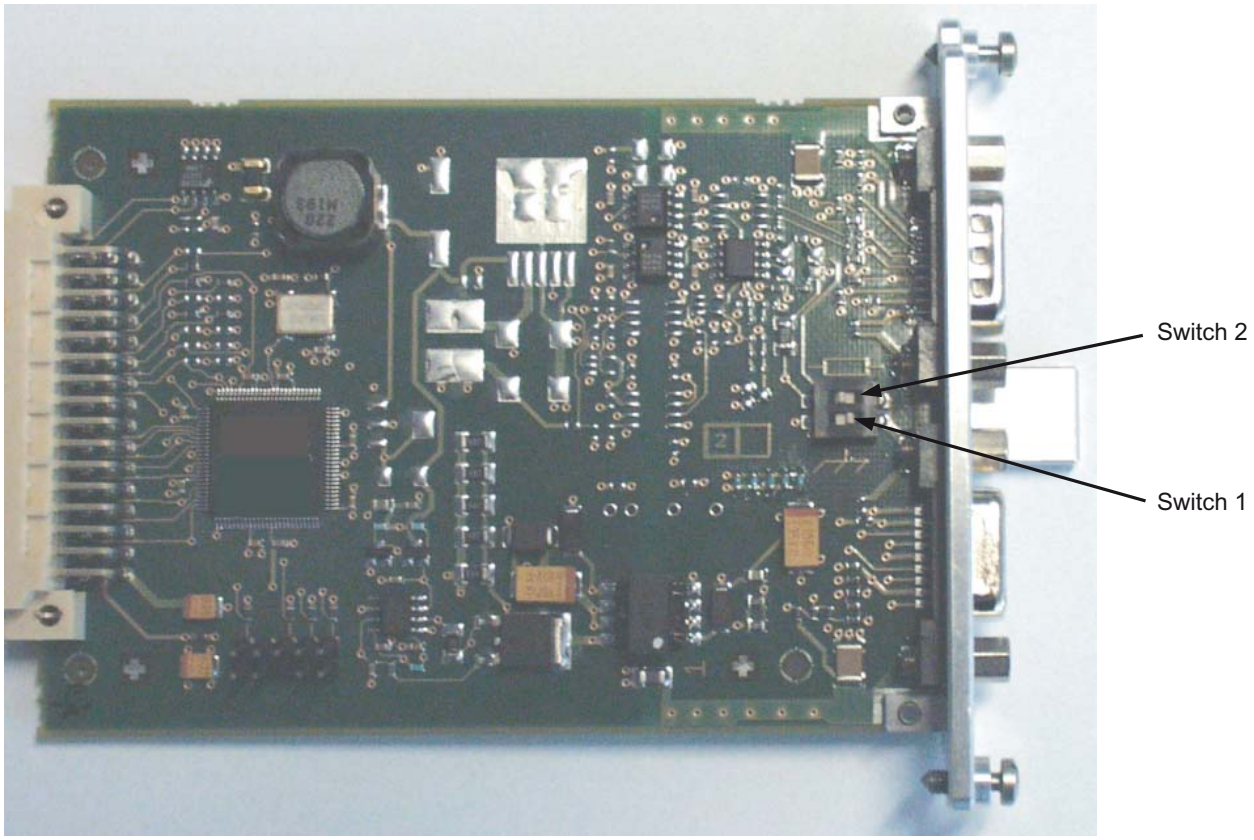
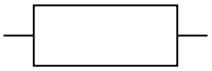
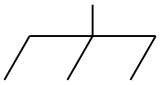


Figure 3-2 Switch 2/1

Table 3- 1 2-pin SMD DIL switch

ID on the component	Switch	Function	Switch position		Default
	2	Bus terminating resistor 120 Ω	OFF	Inactive	OFF
			ON	Active	
	1	Operation with/without ground	OFF	Ground-free operation	OFF
			ON	Operation with ground	

Note

A bus terminating resistor must be located at the end of the bus line. The bus terminating resistor at the CBC10 can be used for this purpose.

3.4 Configure the drive unit using the STARTER commissioning tool

Note

When performing these steps, observe the notes in the SINAMICS S120 Commissioning Manual.

Configure the drive unit in the STARTER commissioning tool by carrying out the following steps:

1. Search for the drive unit ONLINE.

   The component topology and configuration of the drive unit are automatically determined.
2. Configure the motor.
3. Configure the "CBC10" interface on the Control Unit.
   - CAN interface
   - PDO message frame
   - Monitoring
4. Load the project to the drive unit.

3.4.1 Searching for the drive unit ONLINE

The SINAMICS firmware is able to recognize the actual topology automatically and store it in the appropriate parameters.

Procedure

To ensure that the drive unit configuration is identified automatically, open a new project in the STARTER commissioning tool. Proceed as follows:

1. Click on the STARTER symbol  on your user interface to start the STARTER commissioning tool.

   Or

   Call the menu command "Start > SIMATIC > STEP 7 > STARTER" in your Windows Start menu.

   The STARTER Project Wizard is launched.

2. Click on the "Find drive units online..." button.

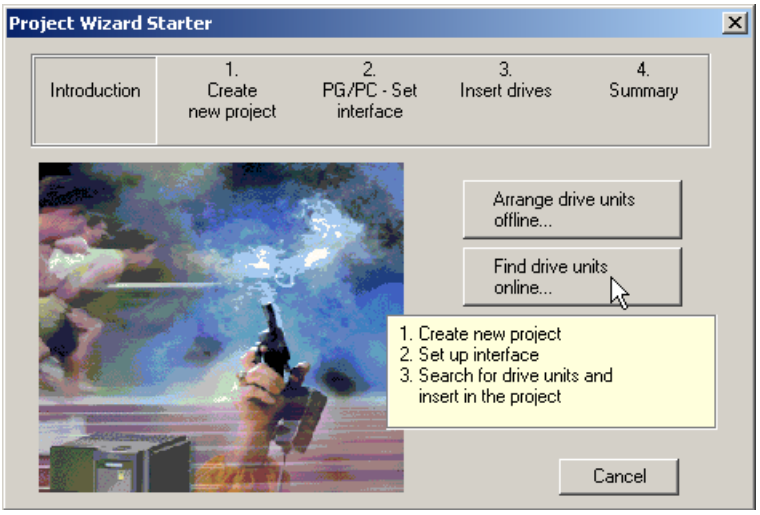


Figure 3-3 Find drive units online...

The project wizard guides you through the procedure for creating a new project.

3. In the next dialog box, enter a name for the project, e.g. "Project_CANopen_0" and click "Continue >".

   The Project Wizard searches for the drive unit ONLINE and inserts it in the project.

4. Click on "Next >".

   The Project Wizard displays a summary of the project.

5. Click on "Finish".

Result

The new project and drive unit are displayed in the STARTER commissioning tool.

Note

Search for drive units or, more precisely, Control Units. This means more than one drive unit will be found if there is more than one Control Unit in the system. The peripheral components of a drive unit (Control Unit, Active Line Module, and so on) are not yet displayed.

3.4.2 Determining the component topology and configuration of the drive unit automatically

Once you have created the project and entered the drive unit with its bus address (e.g. "126") ONLINE, you have to enter the associated component topology and drive unit configuration ONLINE.

Procedure

1. Select the drive unit "Drive_unit_Adr126" in the project navigator.
2. Click on the symbol "Connect to selected target devices" .

   An ONLINE connection is established and the dialog "Online/OFFLINE comparison" will be displayed.
3. Click on "Close" and, if necessary, restore the factory settings.
4. Select the drive unit "Drive_unit_Adr126" in the project navigator.
5. Click on the symbol "Restore factory settings" .
6. Confirm the following queries and messages by choosing "OK":
   - "Restore factory settings?" dialog
   - "The factory settings have been restored" dialog
   - "The data has been successfully copied from RAM to ROM" dialog
7. Double-click the "Automatic configuration" entry in the project navigator below the drive unit.

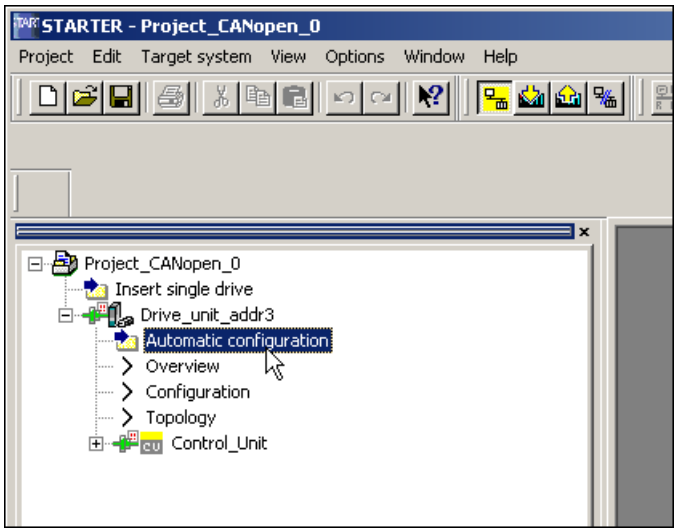


Figure 3-4 Automatic configuration

8. In the "Automatic configuration" dialog, click on "Configure".

   The STARTER commissioning tool automatically searches for all drive unit components that have been correctly connected and loads them to STARTER. In this case, it has recognized two drive objects.

9. In the "Drive object type" dialog box, choose "Servo".

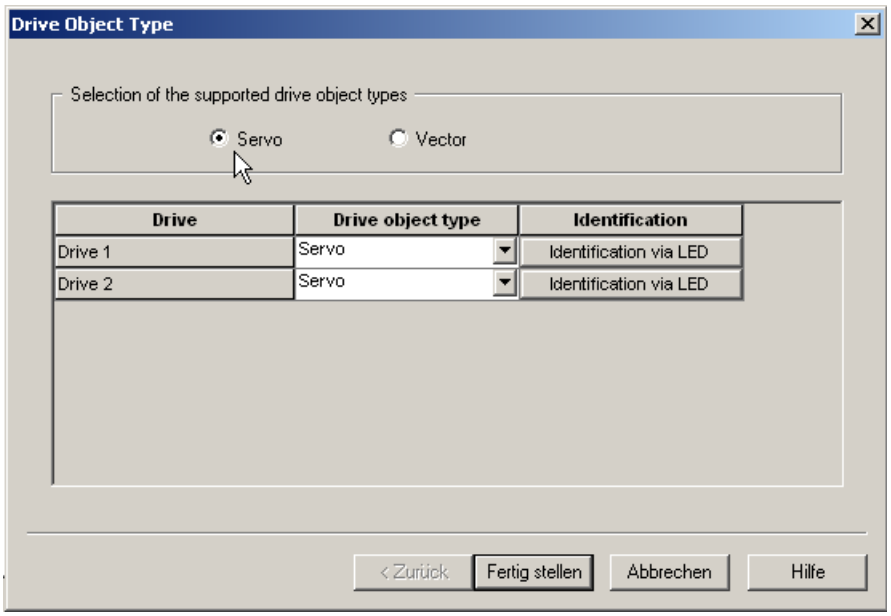


Figure 3-5 Drive object type

Note

When selecting "Vector", CANopen is supported, just like for servo, using the Velocity Mode profile and free process data.

10. Click on "Complete".

    The system loads the data from RAM to ROM and to the PG.

    A note is subsequently displayed that you should additionally configure the motors (see Configuring the motor (Page 76)).

11. Click on "OK" to confirm the note.

    The initialization phase has been completed.

12. In the "Automatic configuration" dialog, click on "Close".

Result

Automatic configuration is complete. All the drive unit components that have been found, such as "Control Unit", "CBC10 ", "Infeed" and "Drive", are displayed in the project navigator.

The following screenshot shows a configured drive in the STARTER commissioning tool. In the work area, the view showing the target and actual topology has been selected. Among other things, you can see that the CBC10 in the drive line-up has been recognized.

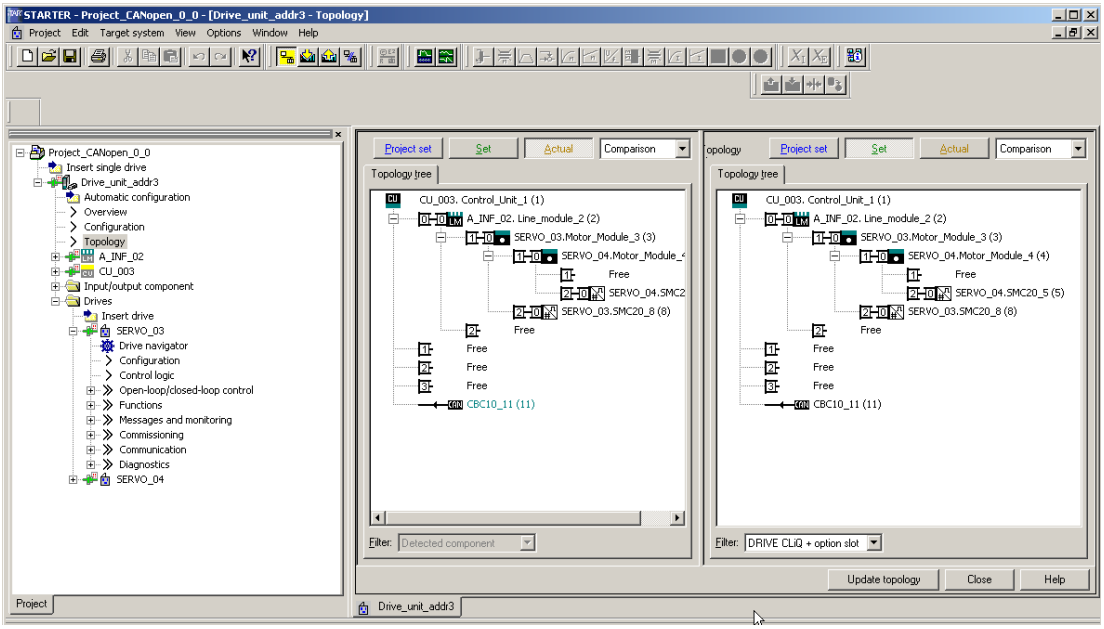


Figure 3-6 Target and actual topology drive configuration (example)

3.4.3 Configuring the motor

In the previous steps, you automatically determined the component topology and configuration of the drive unit and integrated the data in the STARTER project.

Procedure

Note

Only change the configuration of the motor and the encoder. Leave all other configurations as they are (for example, the infeed).

In the following steps, you will be shown how to configure the drive motor and encoder.

1. Click on the symbol "Disconnect from target system... " .

   The modified data is loaded from RAM to ROM and to the PG. The motors are configured in the OFFLINE mode and are then loaded to the target system in ONLINE mode.

2. In the project navigator, switch to the "Drives" folder and double-click "Configuration" below the respective drive.

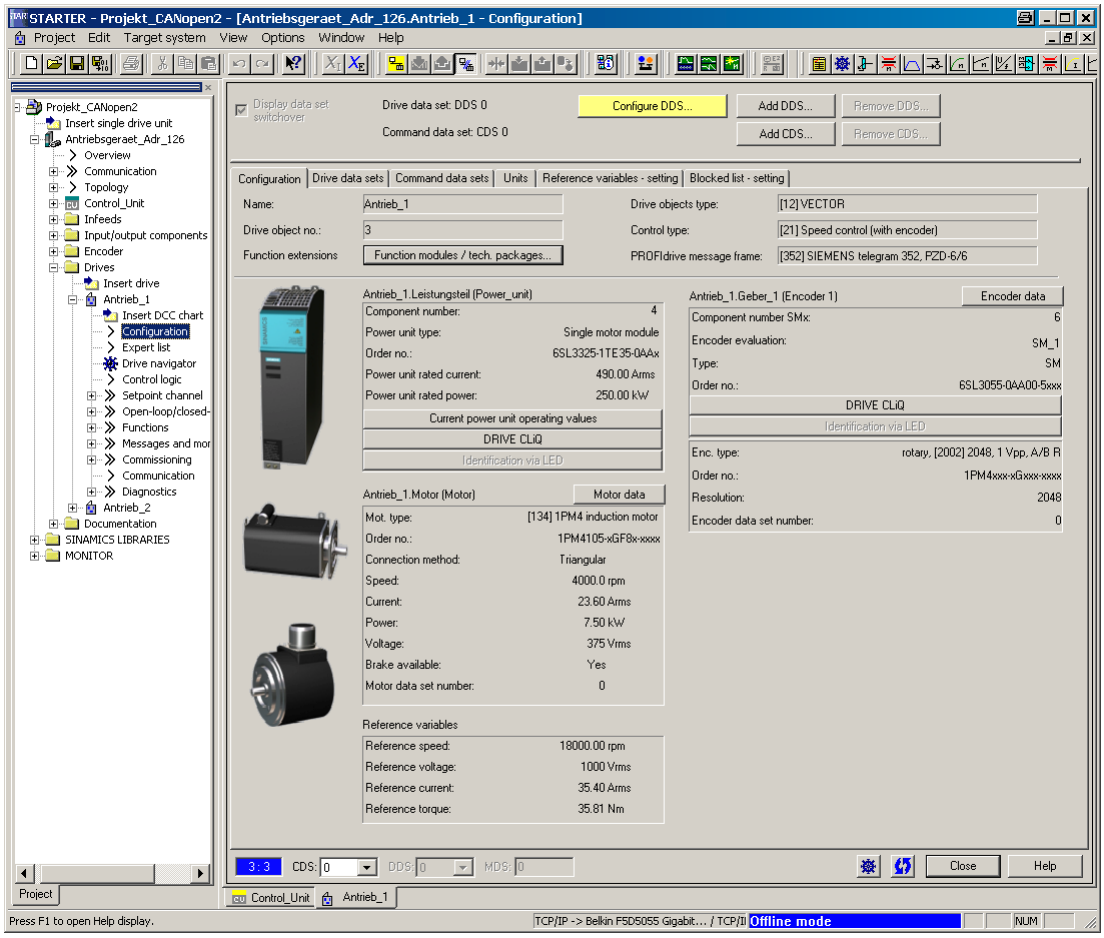


Figure 3-7 Configure the drive (parameterize the motor)

3. In the dialog box that is displayed, click on "Configure DDS...".

   In the "Configuration ..." dialog that is displayed, for the drive object (function module), you can define whether you wish to operate this with/without extended setpoint channel. The commissioning procedure described here is carried out without an extended setpoint channel (ramp-function generator). The field for the extended setpoint channel must be deactivated.

   Note

   When the ramp-function generator is activated (with setpoint channel), the interconnection from CI: p2151 = r1119 can be changed, so that to evaluate bit 10 (target reached) in status word (r8784) the setpoint can be retrieved (taken) from in front of the ramp-function generator.

   When the ramp-function generator is active, objects 6086 hex (motion profile type) and 6083 hex (profile acceleration) of the drive profile become active.

4. Click on "Next > " until you get to motor configuration.

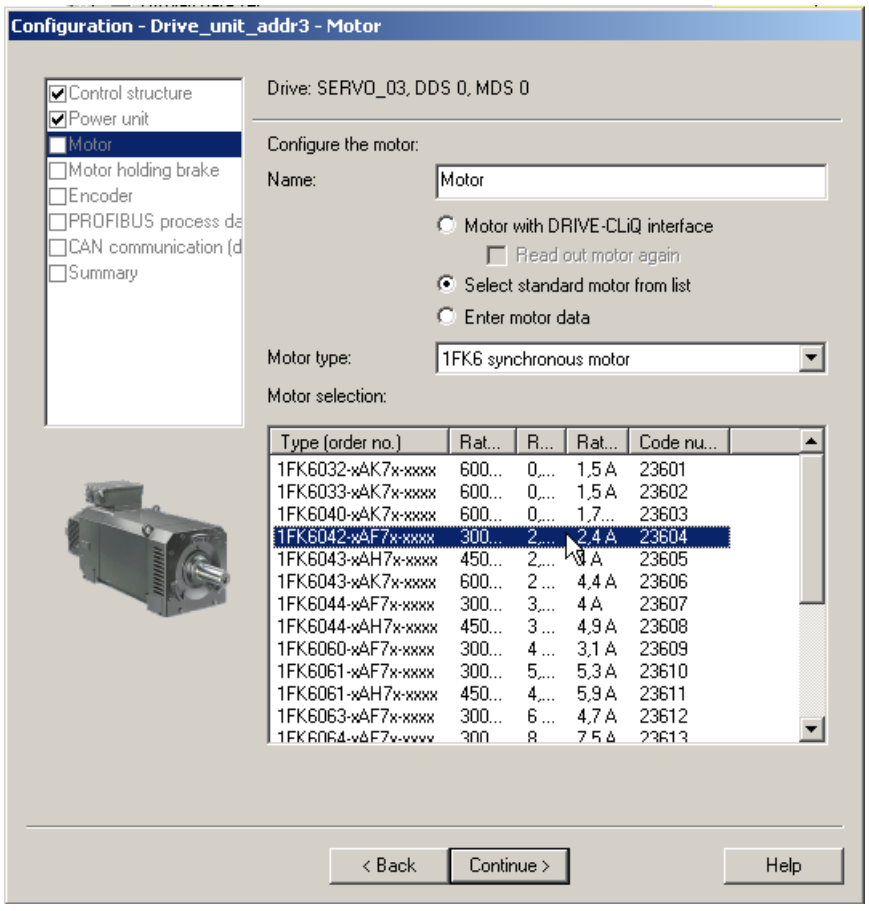


Figure 3-8 Configure the motor

5. Choose the motor type and the required motor (order no.) (see the rating plate).
6. Click on "Next >" until you get to encoder configuration.

7. Select the motor encoder and click on "Next >" until you get to the dialog with the summary.
8. Click on "Complete".

Result

The motor and the encoder of the first drive are configured with these settings. Using the same steps you now also configure the second drive.

See also

Configuring the interface CBC10 on the CU320-2 Control Unit (Page 80)

3.4.4 Configuring the interface CBC10 on the CU320-2 Control Unit

Precondition

You have configured the drive unit with the CBC10 in the STARTER commissioning tool and connected the drive OFFLINE with STARTER.

Procedure

1. In the project navigator, double-click on each of the entries "Drive unit_adr_126" > "Control_Unit" > "CAN Option Module" > "Configuration".

   The configuration of the CBC10 is displayed in the dialog shown below.

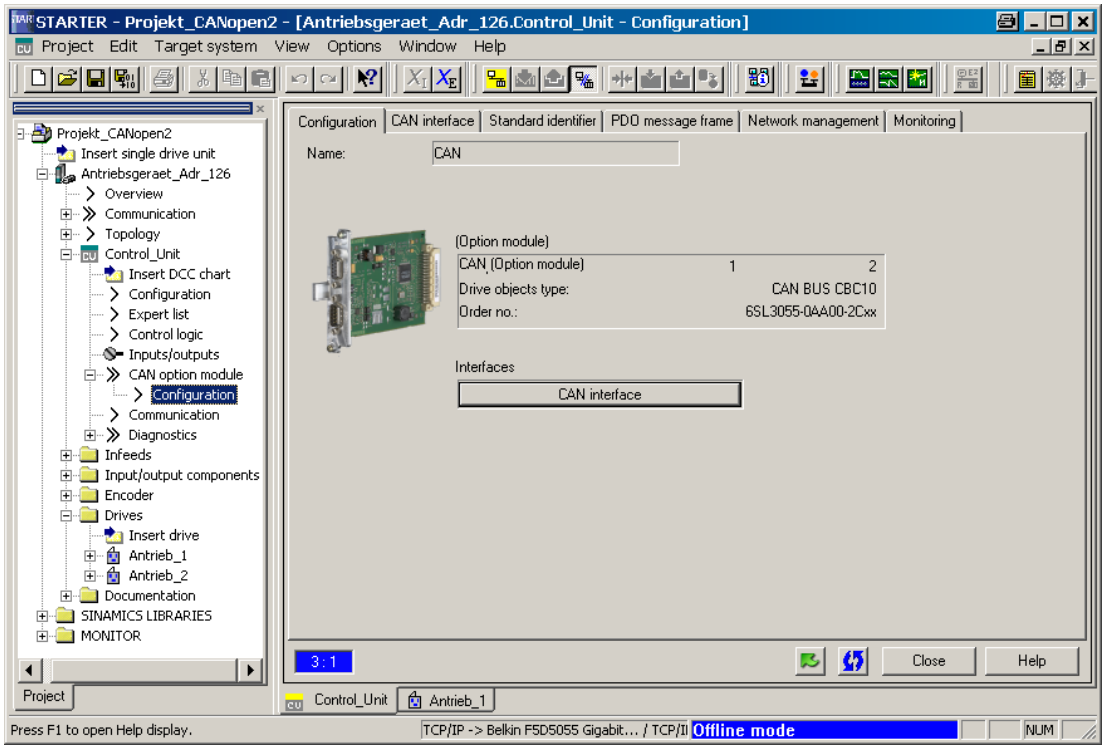


Figure 3-9 Configuration "CBC10"

2. For the CBC10, at the CU320-2 Control Unit, configure the following transmission properties:
   - Transmission rate
   - CAN bus address (Page 81) (Node ID)
   - Number of PDO message frames (Page 82)
   - Node monitoring (Page 83) (heartbeat, life guarding)

3.4.4.1 CAN interface

The transmission rate and the CAN bus address/node ID can be set at the "CAN interface" tab.

Procedure

1. Select the "CAN interface" tab.

   The factory setting for the transmission rate is 20 kbit/s.

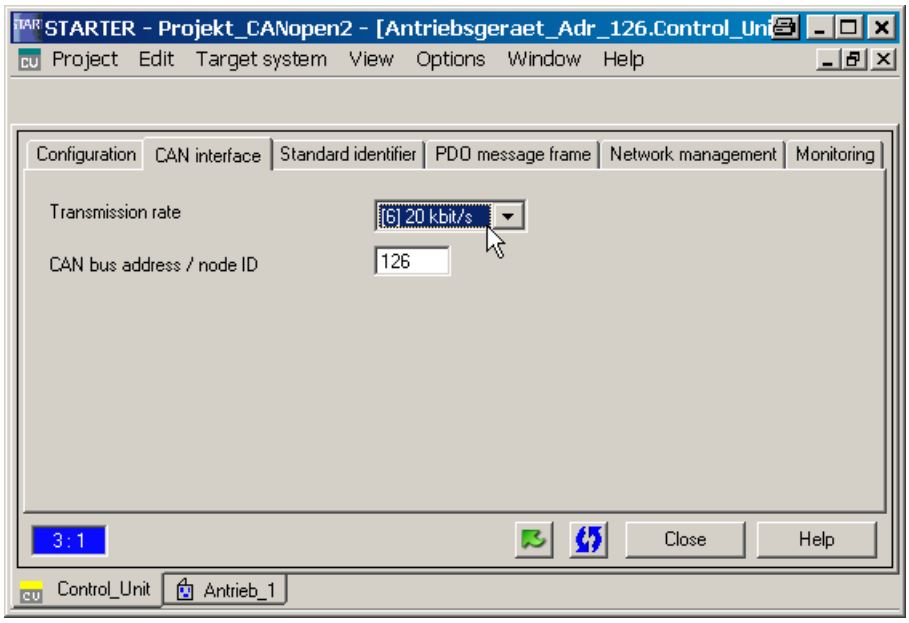


Figure 3-10 CAN interface

2. For commissioning, select another transmission rate (e.g. "1 Mbit/s").

Note

If, during commissioning, you switch the control off/on or carry out a RESET, the factory settings will be restored.

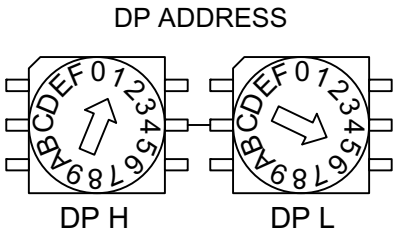
You have two options for the CAN bus address/node ID:

– In this dialog, you can set a value of 1...127 if the address switch on the Control Unit (labeled "DP address") is set to 0.

– Directly using the address switch on the Control Unit.
  Make the settings for "Bus address/node ID" (see also the example).

Example

Setting the CAN bus address via the address switch on the Control Unit.



Example: 15 hex = 21 dec

Figure 3-11 Example: Bus address via the address switch on the Control Unit

Note

Pay particular attention to the following information:

- Permissible CAN bus address is 1...127.
- The address set on the switch is displayed in p8620.
- Each change using the address switch or p8620 is not effective until POWER ON.
- While the SINAMICS is being booted, the address switch is queried first in order to set the bus address. If this is set to 0, the address can be set using parameter p8620.
- If the address is set to a valid node address (1...127), this is copied to parameter p8620, where it is displayed.

3.4.4.2 PDO message frame

This description of the initial commissioning procedure uses a predefined setting ("predefined connection set") with a fixed assignment of SDO and PDO Whereby, the device features the following for each SINAMICS drive object (in this case, the "Single Motor Module"):

- Four receive PDO
- Four transmit PDO

Four receive and four transmit PDOs are set as default in the CU320-2 for each Motor Module. A maximum of eight transmit and receive PDOs can be set.

Activating the predefined connection set

In the OFFLINE mode, to activate the "predefined connection set", carry out the following steps:

1. Select the drive in the project navigator; e.g. "Drive_1".
2. In the "Expert list" shortcut menu, call the expert list of the drive.
3. Search for parameter p8744.
4. Choose "Predefined connection set (1)", to set the "predefined connection set".
5. Repeat steps 1 to 4 for the next SINAMICS drive object, "drive_2".

3.4.4.3 Monitoring

SINAMICS supports the following optional monitoring services to ensure the functionality of CANopen network nodes:

- "Heartbeat"
  SINAMICS (producer) cyclically transmits (heartbeat time) its communication status on the CAN bus to the master application.
- "Life guarding"
  SINAMICS waits a certain time (node life time) for message frames from the master application and permits a specific number (life time factor) of failures within a specified time interval (node guard time).
  The node life time is calculated by multiplying the node guard time by the life time factor.

Note

Only one node monitoring service can be activated at any one time, either "heartbeat" or "life guarding".

Activating both monitoring services is mutually interlocked.

Procedure

1. Select the "Monitoring" tab.

Data transmission mechanism	Default setting for commissioning
Heartbeat	100 ms
Life guarding	Time interval (guard time) = 100 ms
	Number of failures (life time factor) = 3

2. Select one of the data transmission mechanisms and make the necessary settings.

Fault situation

In the case of an error in the CAN communication, e.g. too many message frame failures, fault F(A)08700(2) is signaled (for details see the SINAMICS S120/S150 List Manual). The fault is displayed in parameter r0949. The reaction of the drive to the fault is set with p8641. The reaction of the CAN node is set with p8609.

Note

Parameters p8609 and p8641 determine how the drive or CAN node will respond in the event of a CAN communication or device error.

The factory settings for the parameters are

p8609 = 1, => no change is made

p8641 = 0, => there is no response

Using these parameters, different values can be set for the particular drive object.

Parameter p8609

Sets the behavior of the CAN node referred to the communications error or equipment fault.

- Values:
  - 0: Pre-operational
  - 1: No change (factory setting)
  - 2: Stopped
- Index (corresponds to the CANopen object 1029 hex):
  - [0] = Behavior for communication errors
  - [1] = Behavior for device faults

Parameter p8641

Sets the drive behavior if a CAN communication error occurs.

- Values:
  - 0: No reaction (factory setting)
  - 1: OFF1
  - 2: OFF2
  - 3: OFF3

3.4.5 Loading the project to the drive unit

Procedure

1. Click on the symbol "Connect to selected target devices"  .

   An ONLINE connection is established and an ONLINE/OFFLINE comparison takes place. If any discrepancies are identified, they are displayed (see screenshot below).

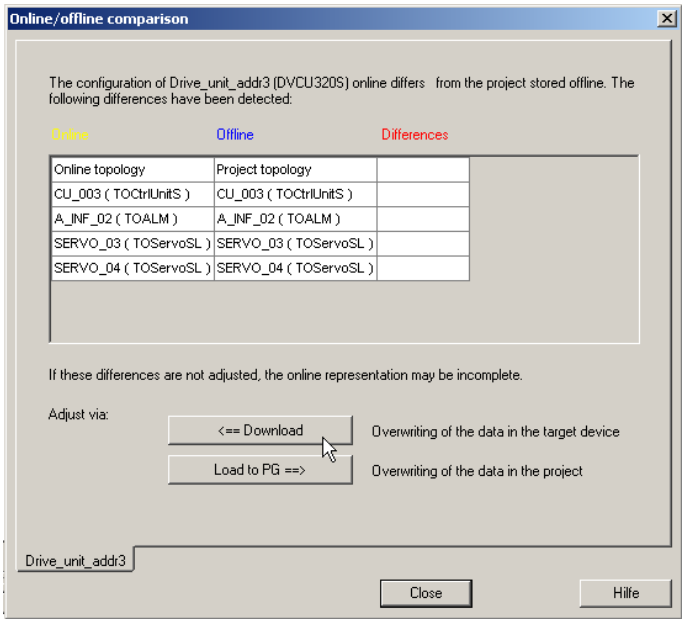


Figure 3-12 ONLINE/OFFLINE comparison (example)

   Because you changed the data OFFLINE, this data must now be loaded into the target device.

2. In the dialog box "ONLINE/OFFLINE comparison, click on the button <-- Load to target device.
3. Click "Yes" at the prompt "Are you sure?".
4. Click in the "The data was successfully loaded to the target device" dialog.
5. When loading from RAM to ROM, click on "OK".

   Discrepancies were identified again during the ONLINE/OFFLINE comparison.

6. Click on the button "Load to PG -->".

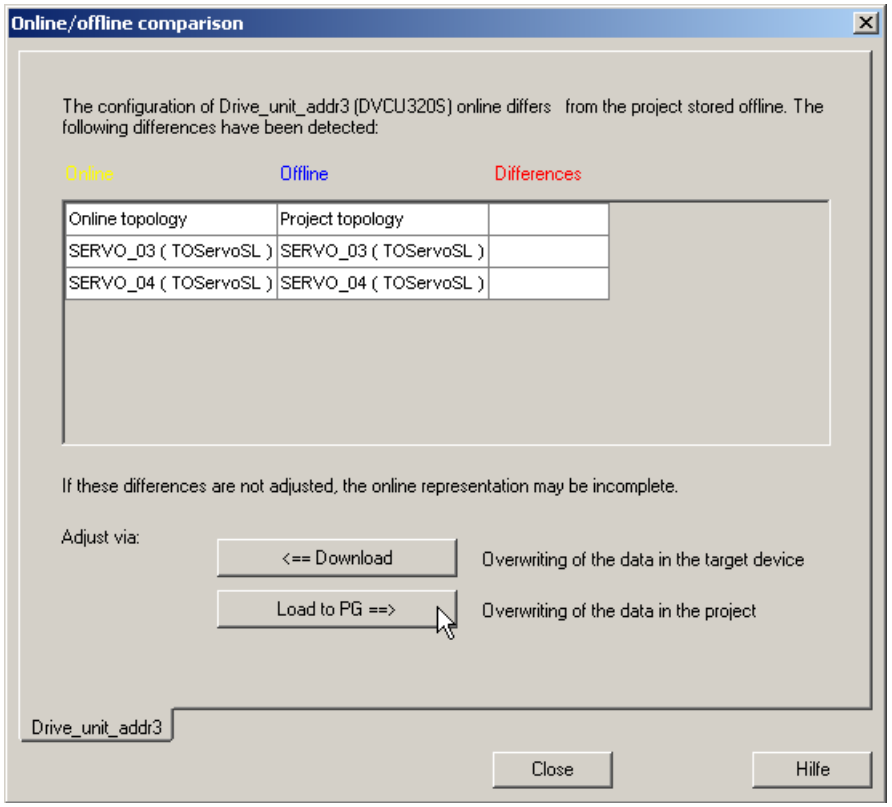


Figure 3-13 Load to PG

7. Click "Yes" at the prompt "Are you sure?".

8. Click in the "The data was successfully loaded to the PG" dialog.

   Discrepancies are no longer displayed in the "ONLINE/OFFLINE comparison" dialog.

9. Click on "Close".

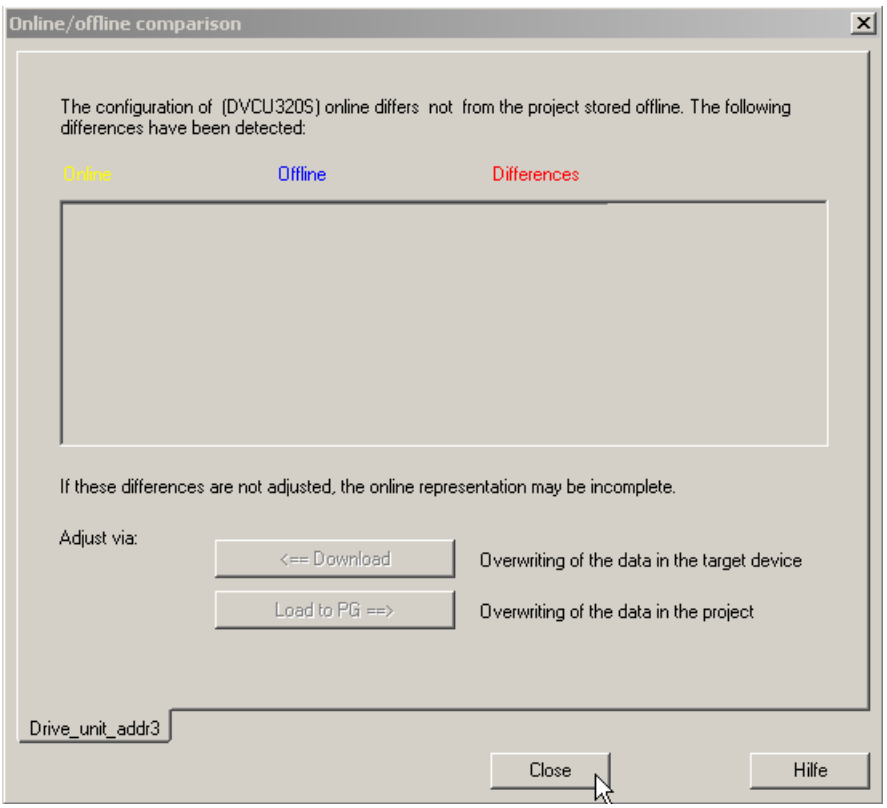


Figure 3-14 ONLINE/OFFLINE comparison OK

Result

This completes the procedure for configuring the drive unit hardware with the CANopen interface. However, before you start configuring the COB IDs and the process data objects for the receive and transmit message frames, you have to interconnect the infeed.

3.4.6 Interconnecting the infeed

In the current software version, the infeed (Active Line Module) cannot be addressed directly via the PDO.

The DC link can still be switch-on via a signal source by means of Active Line Module parameter p0840 (ON/OFF1), however, using the following BICO interconnection options:

- You can interconnect parameter p0840 with a free bit at the digital input of the CU (e.g. r0722 bit 15).
- You can interconnect parameter p0840 with a free bit of the control word from Motor Module 1 (e.g. r8890 bit 15).

Note

In the sample configuration described here, the control word is present at the receive buffer in PZD receive word 1 (see also the function diagram in the appendix).

Note

The principle procedure for BICO interconnections in the STARTER commissioning tool is described in the section "Principle procedure for BICO interconnections in STARTER (Page 57)".

Procedure

Interconnect parameter p0840 (ON/OFF1) of the infeed with the signal source, e.g. from the control word of Motor Module 1 (Drive_1):

1. In the project navigator, select the entry "Infeed".
2. Call the expert list with the "Expert > Expert list" shortcut menu and search for parameter p0840.
3. Interconnect the p parameter with r parameter r8890 bit 15 = PZD 1 from "Drive_1" (SERVO_03).

Note

Please also refer to the commissioning instructions in the SINAMICS S120 Commissioning Manual.

For instance, DC link must be switched-on before the Motor Modules are switched on.

3.5 Configuring receive and transmit PDO message frames

You need to commission the communication and mapping parameters for the individual drive objects. The STARTER commissioning tool is connected to the drive unit online. For each drive object, you can freely configure (free PDO mapping) the following for the receive and transmit message frames:

- Communication parameters
- Mapping parameters

Note

Automatic pre-assignment of parameters for the first commissioning

The communication parameters and mapping parameters are essentially automatically predefined ("predefined connection set") for each drive object when the CANopen interface is commissioned for the first time. The following initial commissioning procedure with free PDO mapping assumes that the parameters have been predefined and describes how they can be changed. If you do not want to parameterize free PDO mapping, you can omit the steps described below and continue commissioning with the section "Interconnecting receive and send buffers (Page 100)".

3.5.1 Predefined COB IDs and process data objects for the drive objects

Opening expert lists

Before you start commissioning free PDO mapping, open the expert lists for the individual drive objects as follows:

1. Select the drive in the project navigator; e.g. "Drive_1".
2. In the "Expert list" shortcut menu, call the expert list of drive object 1.
3. Select the drive in the project navigator; e.g. "drive_2".
4. In the "Expert list" shortcut menu, call the expert list of drive object 2.

   The expert lists for both drive objects are displayed in the STARTER commissioning tool. You can now see the automatically predefined parameters in the "predefined connection set".

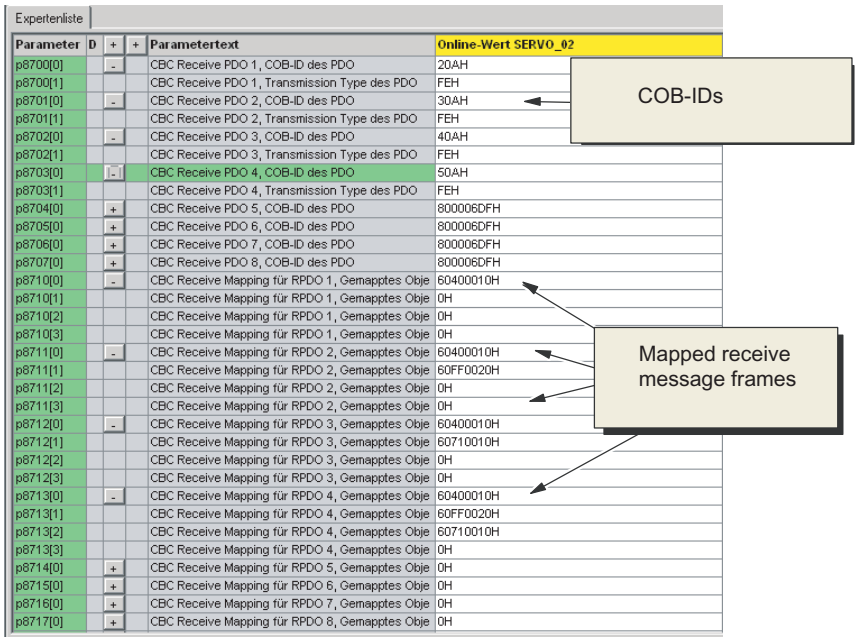
Note

You assign COB IDs for each drive object in the expert list starting from p8700 for receive message frames and from p8720 for transmit message frames.

For each drive object in the expert list, the process data objects for mapping the message frames start from parameter p8710 for receive message frames and from parameter p8730 for transmit message frames.

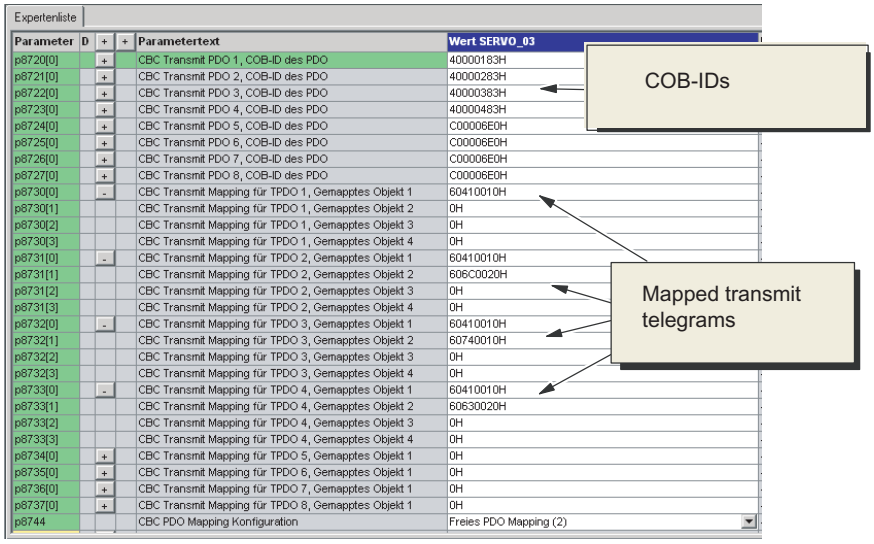
COB IDs and mapped process data objects for drive object 1

The following screenshots show the predefined COB IDs and mapped process data objects for transmitting and receiving drive object 1 in the expert list in the STARTER commissioning tool.



Parameter	D	+	+	Parametertext	Online-Wert SERVO_02
p8700[0]		-		CBC Receive PDO 1, COB-ID des PDO	20AH
p8700[1]				CBC Receive PDO 1, Transmission Type des PDO	FEH
p8701[0]		-		CBC Receive PDO 2, COB-ID des PDO	30AH
p8701[1]				CBC Receive PDO 2, Transmission Type des PDO	FEH
p8702[0]		-		CBC Receive PDO 3, COB-ID des PDO	40AH
p8702[1]				CBC Receive PDO 3, Transmission Type des PDO	FEH
p8703[0]		-		CBC Receive PDO 4, COB-ID des PDO	50AH
p8703[1]				CBC Receive PDO 4, Transmission Type des PDO	FEH
p8704[0]		+		CBC Receive PDO 5, COB-ID des PDO	800006DFH
p8705[0]		+		CBC Receive PDO 6, COB-ID des PDO	800006DFH
p8706[0]		+		CBC Receive PDO 7, COB-ID des PDO	800006DFH
p8707[0]		+		CBC Receive PDO 8, COB-ID des PDO	800006DFH
p8710[0]		-		CBC Receive Mapping für RPDO 1, Gemapptes Obje	60400010H
p8710[1]				CBC Receive Mapping für RPDO 1, Gemapptes Obje	0H
p8710[2]				CBC Receive Mapping für RPDO 1, Gemapptes Obje	0H
p8710[3]				CBC Receive Mapping für RPDO 1, Gemapptes Obje	0H
p8711[0]		-		CBC Receive Mapping für RPDO 2, Gemapptes Obje	60400010H
p8711[1]				CBC Receive Mapping für RPDO 2, Gemapptes Obje	60FF0020H
p8711[2]				CBC Receive Mapping für RPDO 2, Gemapptes Obje	0H
p8711[3]				CBC Receive Mapping für RPDO 2, Gemapptes Obje	0H
p8712[0]		-		CBC Receive Mapping für RPDO 3, Gemapptes Obje	60400010H
p8712[1]				CBC Receive Mapping für RPDO 3, Gemapptes Obje	60710010H
p8712[2]				CBC Receive Mapping für RPDO 3, Gemapptes Obje	0H
p8712[3]				CBC Receive Mapping für RPDO 3, Gemapptes Obje	0H
p8713[0]		-		CBC Receive Mapping für RPDO 4, Gemapptes Obje	60400010H
p8713[1]				CBC Receive Mapping für RPDO 4, Gemapptes Obje	60FF0020H
p8713[2]				CBC Receive Mapping für RPDO 4, Gemapptes Obje	60710010H
p8713[3]				CBC Receive Mapping für RPDO 4, Gemapptes Obje	0H
p8714[0]		+		CBC Receive Mapping für RPDO 5, Gemapptes Obje	0H
p8715[0]		+		CBC Receive Mapping für RPDO 6, Gemapptes Obje	0H
p8716[0]		+		CBC Receive Mapping für RPDO 7, Gemapptes Obje	0H
p8717[0]		+		CBC Receive Mapping für RPDO 8, Gemapptes Obje	0H

Figure 3-15 COB IDs and mapped process data objects for receive message frame drive object 1

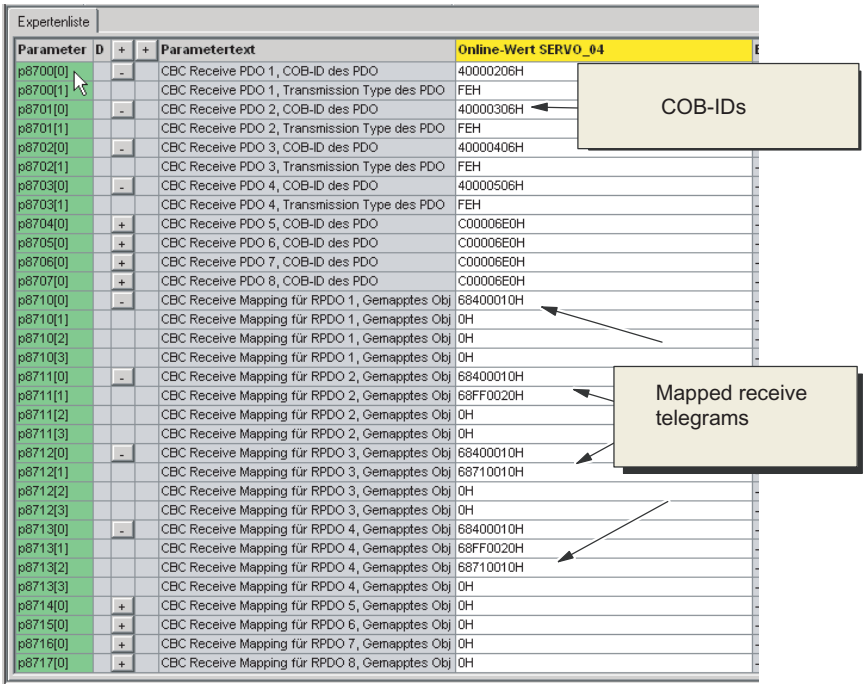


Parameter	D	+	+	Parametertext	Wert SERVO_03
p8720[0]		+		CBC Transmit PDO 1, COB-ID des PDO	40000183H
p8721[0]		+		CBC Transmit PDO 2, COB-ID des PDO	40000283H
p8722[0]		+		CBC Transmit PDO 3, COB-ID des PDO	40000383H
p8723[0]		+		CBC Transmit PDO 4, COB-ID des PDO	40000483H
p8724[0]		+		CBC Transmit PDO 5, COB-ID des PDO	C00006E0H
p8725[0]		+		CBC Transmit PDO 6, COB-ID des PDO	C00006E0H
p8726[0]		+		CBC Transmit PDO 7, COB-ID des PDO	C00006E0H
p8727[0]		+		CBC Transmit PDO 8, COB-ID des PDO	C00006E0H
p8730[0]		-		CBC Transmit Mapping für TPDO 1, Gemapptes Objekt 1	60410010H
p8730[1]				CBC Transmit Mapping für TPDO 1, Gemapptes Objekt 2	0H
p8730[2]				CBC Transmit Mapping für TPDO 1, Gemapptes Objekt 3	0H
p8730[3]				CBC Transmit Mapping für TPDO 1, Gemapptes Objekt 4	0H
p8731[0]		-		CBC Transmit Mapping für TPDO 2, Gemapptes Objekt 1	60410010H
p8731[1]				CBC Transmit Mapping für TPDO 2, Gemapptes Objekt 2	606C0020H
p8731[2]				CBC Transmit Mapping für TPDO 2, Gemapptes Objekt 3	0H
p8731[3]				CBC Transmit Mapping für TPDO 2, Gemapptes Objekt 4	0H
p8732[0]		-		CBC Transmit Mapping für TPDO 3, Gemapptes Objekt 1	60410010H
p8732[1]				CBC Transmit Mapping für TPDO 3, Gemapptes Objekt 2	60740010H
p8732[2]				CBC Transmit Mapping für TPDO 3, Gemapptes Objekt 3	0H
p8732[3]				CBC Transmit Mapping für TPDO 3, Gemapptes Objekt 4	0H
p8733[0]		-		CBC Transmit Mapping für TPDO 4, Gemapptes Objekt 1	60410010H
p8733[1]				CBC Transmit Mapping für TPDO 4, Gemapptes Objekt 2	60630020H
p8733[2]				CBC Transmit Mapping für TPDO 4, Gemapptes Objekt 3	0H
p8733[3]				CBC Transmit Mapping für TPDO 4, Gemapptes Objekt 4	0H
p8734[0]		+		CBC Transmit Mapping für TPDO 5, Gemapptes Objekt 1	0H
p8735[0]		+		CBC Transmit Mapping für TPDO 6, Gemapptes Objekt 1	0H
p8736[0]		+		CBC Transmit Mapping für TPDO 7, Gemapptes Objekt 1	0H
p8737[0]		+		CBC Transmit Mapping für TPDO 8, Gemapptes Objekt 1	0H
p8744				CBC PDO Mapping Konfiguration	Freies PDO Mapping (2)

Figure 3-16 COB IDs and mapped process data objects for transmit message frame drive object 1

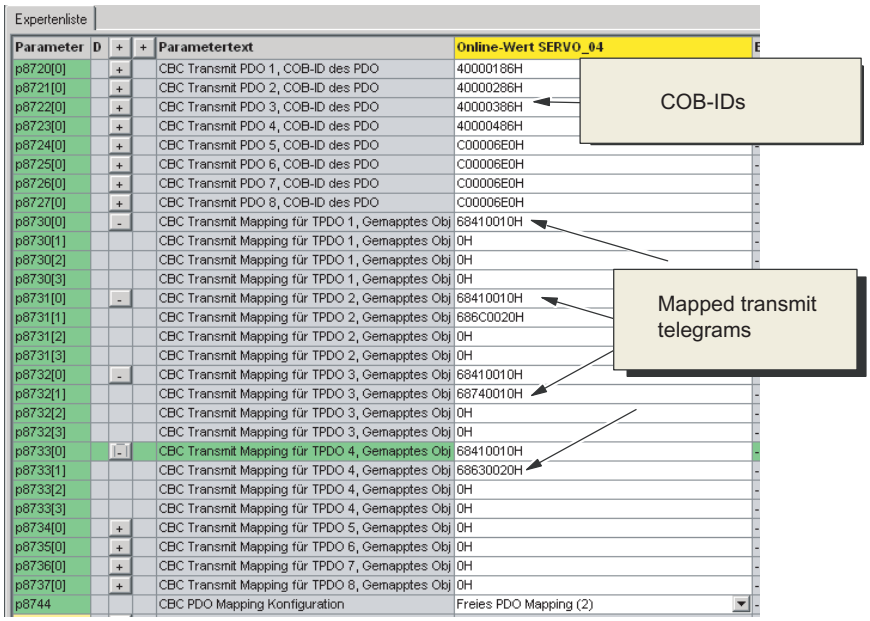
COB IDs and mapped process data objects for drive object 2

The following screenshots show the predefined COB IDs and mapped process data objects for transmitting and receiving drive object 2 in the expert list in the STARTER commissioning tool. The process data objects are parameterized with an offset of 800 hex for the second drive object.



Parameter	D	+	+	Parametertext	Online-Wert SERVO_04
p8700[0]		-		CBC Receive PDO 1, COB-ID des PDO	40000206H
p8700[1]				CBC Receive PDO 1, Transmission Type des PDO	FEH
p8701[0]		-		CBC Receive PDO 2, COB-ID des PDO	40000306H
p8701[1]				CBC Receive PDO 2, Transmission Type des PDO	FEH
p8702[0]		-		CBC Receive PDO 3, COB-ID des PDO	40000406H
p8702[1]				CBC Receive PDO 3, Transmission Type des PDO	FEH
p8703[0]		-		CBC Receive PDO 4, COB-ID des PDO	40000506H
p8703[1]				CBC Receive PDO 4, Transmission Type des PDO	FEH
p8704[0]		+		CBC Receive PDO 5, COB-ID des PDO	C00006E0H
p8705[0]		+		CBC Receive PDO 6, COB-ID des PDO	C00006E0H
p8706[0]		+		CBC Receive PDO 7, COB-ID des PDO	C00006E0H
p8707[0]		+		CBC Receive PDO 8, COB-ID des PDO	C00006E0H
p8710[0]		-		CBC Receive Mapping für RPDO 1, Gemapptes Obj	68400010H
p8710[1]				CBC Receive Mapping für RPDO 1, Gemapptes Obj	0H
p8710[2]				CBC Receive Mapping für RPDO 1, Gemapptes Obj	0H
p8710[3]				CBC Receive Mapping für RPDO 1, Gemapptes Obj	0H
p8711[0]		-		CBC Receive Mapping für RPDO 2, Gemapptes Obj	68400010H
p8711[1]				CBC Receive Mapping für RPDO 2, Gemapptes Obj	68FF0020H
p8711[2]				CBC Receive Mapping für RPDO 2, Gemapptes Obj	0H
p8711[3]				CBC Receive Mapping für RPDO 2, Gemapptes Obj	0H
p8712[0]		-		CBC Receive Mapping für RPDO 3, Gemapptes Obj	68400010H
p8712[1]				CBC Receive Mapping für RPDO 3, Gemapptes Obj	68710010H
p8712[2]				CBC Receive Mapping für RPDO 3, Gemapptes Obj	0H
p8712[3]				CBC Receive Mapping für RPDO 3, Gemapptes Obj	0H
p8713[0]		-		CBC Receive Mapping für RPDO 4, Gemapptes Obj	68400010H
p8713[1]				CBC Receive Mapping für RPDO 4, Gemapptes Obj	68FF0020H
p8713[2]				CBC Receive Mapping für RPDO 4, Gemapptes Obj	68710010H
p8713[3]				CBC Receive Mapping für RPDO 4, Gemapptes Obj	0H
p8714[0]		+		CBC Receive Mapping für RPDO 5, Gemapptes Obj	0H
p8715[0]		+		CBC Receive Mapping für RPDO 6, Gemapptes Obj	0H
p8716[0]		+		CBC Receive Mapping für RPDO 7, Gemapptes Obj	0H
p8717[0]		+		CBC Receive Mapping für RPDO 8, Gemapptes Obj	0H

Figure 3-17 COB IDs and mapped process data objects for receive message frames drive object 2



Parameter	D	+	+	Parametertext	Online-Wert SERVO_04
p8720[0]		+		CBC Transmit PDO 1, COB-ID des PDO	40000186H
p8721[0]		+		CBC Transmit PDO 2, COB-ID des PDO	40000286H
p8722[0]		+		CBC Transmit PDO 3, COB-ID des PDO	40000386H
p8723[0]		+		CBC Transmit PDO 4, COB-ID des PDO	40000486H
p8724[0]		+		CBC Transmit PDO 5, COB-ID des PDO	C00006E0H
p8725[0]		+		CBC Transmit PDO 6, COB-ID des PDO	C00006E0H
p8726[0]		+		CBC Transmit PDO 7, COB-ID des PDO	C00006E0H
p8727[0]		+		CBC Transmit PDO 8, COB-ID des PDO	C00006E0H
p8730[0]		-		CBC Transmit Mapping für TPDO 1, Gemapptes Obj	68410010H
p8730[1]				CBC Transmit Mapping für TPDO 1, Gemapptes Obj	0H
p8730[2]				CBC Transmit Mapping für TPDO 1, Gemapptes Obj	0H
p8730[3]				CBC Transmit Mapping für TPDO 1, Gemapptes Obj	0H
p8731[0]		-		CBC Transmit Mapping für TPDO 2, Gemapptes Obj	68410010H
p8731[1]				CBC Transmit Mapping für TPDO 2, Gemapptes Obj	686C0020H
p8731[2]				CBC Transmit Mapping für TPDO 2, Gemapptes Obj	0H
p8731[3]				CBC Transmit Mapping für TPDO 2, Gemapptes Obj	0H
p8732[0]		-		CBC Transmit Mapping für TPDO 3, Gemapptes Obj	68410010H
p8732[1]				CBC Transmit Mapping für TPDO 3, Gemapptes Obj	68740010H
p8732[2]				CBC Transmit Mapping für TPDO 3, Gemapptes Obj	0H
p8732[3]				CBC Transmit Mapping für TPDO 3, Gemapptes Obj	0H
p8733[0]		-		CBC Transmit Mapping für TPDO 4, Gemapptes Obj	68410010H
p8733[1]				CBC Transmit Mapping für TPDO 4, Gemapptes Obj	68630020H
p8733[2]				CBC Transmit Mapping für TPDO 4, Gemapptes Obj	0H
p8733[3]				CBC Transmit Mapping für TPDO 4, Gemapptes Obj	0H
p8734[0]		+		CBC Transmit Mapping für TPDO 5, Gemapptes Obj	0H
p8735[0]		+		CBC Transmit Mapping für TPDO 6, Gemapptes Obj	0H
p8736[0]		+		CBC Transmit Mapping für TPDO 7, Gemapptes Obj	0H
p8737[0]		+		CBC Transmit Mapping für TPDO 8, Gemapptes Obj	0H
p8744				CBC PDO Mapping Konfiguration	Freies PDO Mapping (2)

Figure 3-18 COB IDs and mapped process data objects for transmit message frames drive object 2

3.5.2 Activating free PDO mapping

Procedure

To activate free PDO mapping in SINAMICS drive object 2 (Single Motor Module 2), carry out the following:

1. Select the drive in the project navigator; e.g. "drive_2".
2. In the "Expert list" shortcut menu, call the expert list of drive object 2.
3. Select parameter p8744.

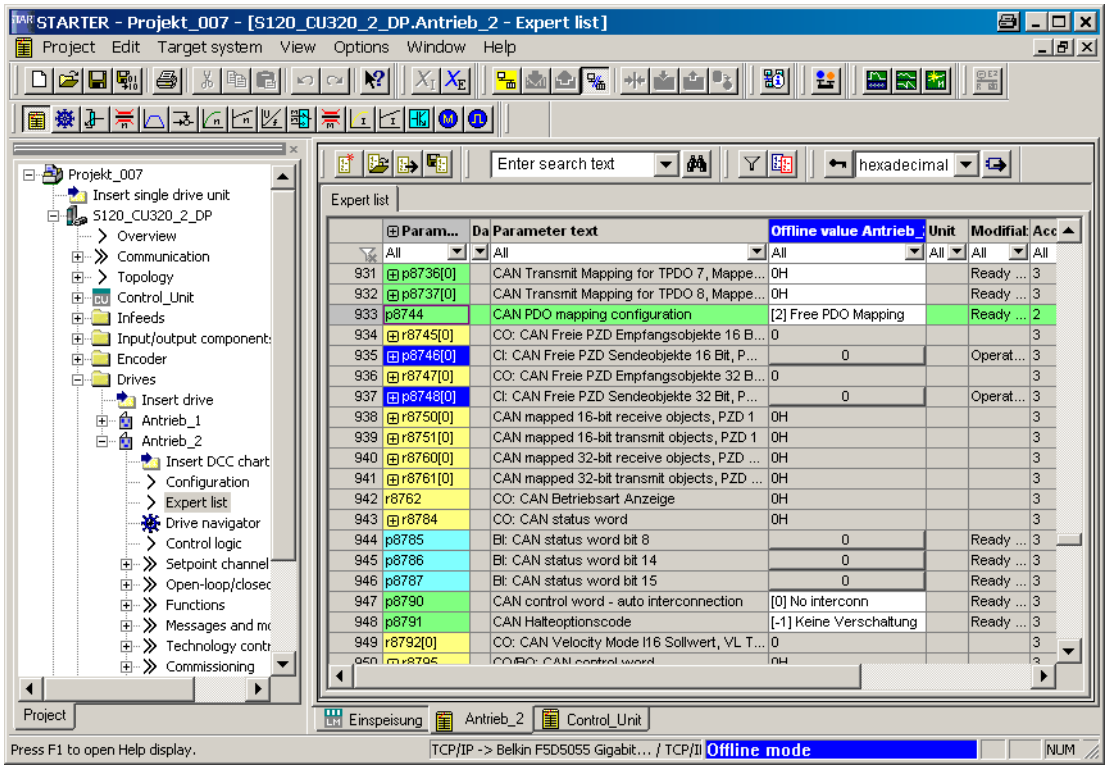


Figure 3-19 Free PDO mapping

4. Select the setting "Free PDO mapping (2)".

3.5.3 Assigning COB IDs and mapping parameters for free PDO mapping

If you are in the ONLINE mode and want to assign free COB IDs and mapping parameters, you have the following options:

- COB IDs
  You can assign a COB as required in the following parameters:
  - Receive (RPDO) in parameters p8700 to p8707
  - Transmit (TPDO) in parameters p8720 to p8727
- Mapping parameters
  You can enter the process data objects for mapping the transmission message frames in the following parameters:
  - Receive message frames (RPDO) starting from parameter p8710
  - Transmit message frames (TPDO) starting from parameter p8730

Note

If mapping parameters are changed in the STARTER commissioning tool in ONLINE mode, the COB ID for the PDO in question must first be set to "invalid" in the communication parameters and, once the parameters have been changed, reset to "valid".

COB-ID, bit 31 = 0 -> COB-ID valid,

bit 31 = 1 -> COB ID invalid

Procedure

To carry out free PDO mapping in the ONLINE mode, proceed as follows:

1. Set the COB ID of the RPDO or TPDO in question to "invalid" (e.g.: p8700[0], COB-ID of the PDO).
2. Enter the process data objects as mapping parameters into the relevant RPDO or TPDO (for e.g.: from p8710[0], mapped object).
3. Set the COB ID of the RPDO or TPDO in question to "valid".
4. Activate the confirmation of the actual mapping configuration using p8741.

Operating steps

An example is used to illustrate the procedure described below (in this example: RPDO1 for drive object 2).

Setting the COB ID to "invalid"

1. Select the drive in the project navigator; e.g. "drive_2".
2. Call the expert list for the Single Motor Module 2 (drive_2) using the shortcut menu "Expert list".
3. Search for COB ID parameter p8700[0] for communication parameter RPDO1.
4. Transfer the hexadecimal value from the STARTER commissioning tool to a calculator and enter **Or** 8000 0000 hex. Transfer the result to the STARTER parameter.

   As a result, bit 31 has now been set to "invalid".

   You can now adjust the mapping parameters.

Adapting mapping parameters

1. Choose the process data object to be mapped (e.g. RPDO1, control word = 6040 hex) from the "Objects of the drive profile DSP402" table (see "Communication objects (Page 152)") .
2. Add a suitable offset for the SINAMICS drive object (e.g. plus 800 hex starting from drive object 2).
   Refer to the column "Values from table "OD index (hex) (e.g. 6840 hex)"" in the parameterization table in the following section.
3. Convert the OD index (bits 31…16), subindex (bits 15…8) and object size (bits 7…0) into a hexadecimal value (32 bits).
   See the column "Resulting hexadecimal value" (e.g. 6840 0010 hex) in the parameterization table in the following section.
4. Enter this value in the STARTER commissioning tool in the corresponding mapping parameter.
   See the column "Mapping parameters in STARTER" (e.g. p8710[0]) in the parameterization table in the following section.

Setting the COB ID to "valid"

1. Select the drive in the project navigator; e.g. "drive_2".
2. Call the expert list for the Single Motor Module 2 using the shortcut menu "Expert list".
3. Search for COB ID parameter p8700[0] for communication parameter RPDO1.
4. Transfer the hexadecimal value from the STARTER commissioning tool to a calculator and enter **And** 7FFF FFFF hex. Transfer the result to the STARTER parameter. As a result, bit 31 has now been set to "valid".
5. Copy the value to STARTER.

Note

The RPDO and TPDO described below serve to illustrate the free PDO mapping process. You can freely select the way in which you map your own process data objects for data transfer message frames.

Mapping RPDO1

When you choose the process data object for the control word from the "Objects of the drive profile DSP402" table, this yields the following values, which you enter in parameters p8710[0] to p8710[3] for RPDO1 mapping (see the "Resulting hexadecimal value" column in the following table):

Note

Since it is SINAMICS drive object 2, 800 hex is added to the index.

Table 3- 2 Values for RPDO1

Object name	Values from "Objects of the drive profile DSP402" table			Mapping parameters in STARTER	
	OD index (hex)	Subindex (hex)	Object size	Resulting hexadecimal value	Parameter RPDO1
Control word	6840	00	10 hex (16 bit)	6840 0010	p8710[0]
No object	-	-	-	0	p8710[1...3]

Mapping RPDO2

When you choose the process data object for the control word and the setpoint velocity from the "Objects of the drive profile DSP402" table, this yields the following values, which you enter in parameters p8711[0...3] for RPDO2 mapping (see the "Resulting hexadecimal value" column in the following table):

Table 3- 3 Values for RPDO2

Object name	Values from "Objects of the drive profile DSP402" table			Mapping parameters in STARTER	
	OD index (hex)	Subindex (hex)	Object size	Resulting hexadecimal value	Parameter RPDO2
Control word	6840	00	10 hex (16 bit)	6840 0010	p8711[0]
Setpoint velocity	68FF	00	20 hex (32 bit)	68FF 0020	p8711[1]
No object	-	-	-	0	p8711[2...3]

Mapping RPDO3

When you choose the process data object for the control word and the setpoint torque from the "Objects of the drive profile DSP402" table, this yields the following values, which you enter in parameters p8712[0...3] for RPDO3 mapping (see the "Resulting hexadecimal value" column in the following table):

Table 3- 4 Values for RPDO3

Object name	Values from "Objects of the drive profile DSP402" table			Mapping parameters in STARTER	
	OD index (hex)	Subindex (hex)	Object size	Resulting hexadecimal value	Parameter RPDO3
Control word	6840	00	10 hex (16 bit)	6840 0010	p8712[0]
Setpoint torque	6871	00	10 hex (16 bit)	6871 0010	p8712[1]
No object	-	-	-	0	p8712[2...3]

Mapping RPDO4

When you choose the process data object for the control word, setpoint velocity and the setpoint torque from the "Objects of the drive profile DSP402" table, this yields the following values, which you enter in parameter p8713[0...3] for RPDO4 mapping (see the "Resulting hexadecimal value" column in the following table):

Table 3- 5 Values for RPDO4

Object name	Values from "Objects of the drive profile DSP402" table			Mapping parameters in STARTER	
	OD index (hex)	Subindex (hex)	Object size	Resulting hexadecimal value	Parameter RPDO4
Control word	6840	00	10 hex (16 bit)	6840 0010	p8713[0]
Setpoint velocity	68FF	00	20 hex (32 bit)	68FF 0020	p8713[1]
Setpoint torque	6871	00	10 hex (16 bit)	6871 0010	p8713[2]
No object	-	-	-	0	P8713[3]

Mapping TPDO1

When you choose the process data object for the CBC status word from the "Objects of the drive profile DSP402" table, this yields the following values, which you enter in parameter p8730[0...3] for TPDO1 mapping (see the "Resulting hexadecimal value" column in the following table):

Table 3- 6 Values for TPDO1

Object name	Values from "Objects of the drive profile DSP402" table			Mapping parameters in STARTER	
	OD index (hex)	Subindex (hex)	Object size	Resulting hexadecimal value	Parameter TPDO1
CBC status word	6841	00	10 hex (16 bit)	6841 0010	p8730[0]
No object	-	-	-	0	p8730[1...3]

Mapping TPDO2

When you choose the process data object for the CBC status word and the setpoint velocity from the "Objects of the drive profile DSP402" table, this yields the following values, which you enter in parameter p8731[0...3] for TPDO2 mapping (see the "Resulting hexadecimal value" column in the following table):

Table 3- 7 Values for TPDO2

Object name	Values from "Objects of the drive profile DSP402" table			Mapping parameters in STARTER	
	OD index (hex)	Subindex (hex)	Object size	Resulting hexadecimal value	Parameter TPDO2
CBC status word	6841	00	10 hex (16 bit)	6841 0010	p8731[0]
Actual velocity	686C	00	20 hex (32 bit)	686C 0020	p8731[1]
No object	-	-	-	0	p8731[2...3]

Mapping TPDO3

When you choose the process data object for the CBC status word and the actual torque from the "Objects of the drive profile DSP402" table, this yields the following values, which you enter in parameter p8732[0...3] for TPDO3 mapping (see the "Resulting hexadecimal value" column in the following table):

Table 3- 8 Values for TPDO3

Object name	Values from "Objects of the drive profile DSP402" table			Mapping parameters in STARTER	
	OD index (hex)	Subindex (hex)	Object size	Resulting hexadecimal value	Parameter TPDO3
CBC status word	6841	00	10 hex (16 bit)	6841 0010	p8732[0]
Actual torque	6874	00	10 hex (16 bit)	6874 0010	p8732[1]
No object	-	-	-	0	p8732[2...3]

Mapping TPDO4

When you choose the process data object for the CBC status word and the encoder position actual value from the "Objects of the drive profile DSP402" table, this yields the following values, which you enter in parameter p8733[0...3] for TPDO4 mapping (see the "Resulting hexadecimal value" column in the following table):

Table 3- 9 Values for TPDO4

Object name	Values from "Objects of the drive profile DSP402" table			Mapping parameters in STARTER	
	OD index (hex)	Subindex (hex)	Object size	Resulting hexadecimal value	Parameter TPDO4
CBC status word	6841	00	10 hex (16 bit)	6841 0010	p8733[0]
Actual position value	6863	00	20 hex (32 bit)	6863 0020	p8733[1]
No object	-	-	-	0	p8733[2...3]

3.6 Interconnecting process data in the receive and transmit buffers

The process data objects for the transmission message frames must be interconnected by means of BICO interconnection for the interface between SINAMICS and the CAN bus. During initial commissioning, proceed as follows:

1. Read out the image of the process data objects to the PZD receive and PZD transmit words for the receive and transmit buffers.
2. Interconnect the source parameter of the PZD receive word at the receive buffer with the SINAMICS target parameters of the process data object.
3. Interconnect the SINAMICS source parameter of the process data object with the target parameters of the PZD transmit word at the transmit buffer.

Note

The STARTER commissioning tool is in the ONLINE mode.

3.6.1 Reading the image of the individual process data objects for the receive and transmit buffers

Once you have parameterized PDO mapping, the device automatically recognizes how the individual process data objects have to be distributed to the PZD receive and PZD transmit words.

To read out the image of the individual process data objects to the PZD receive and PZD transmit words for the receive and transmit buffers, carry out the following steps.

Example

Note

Each drive object has a separate receive and transmit buffer for transferring message frames.

The expert list for each of the drive objects contains the images for the following parameters:

- 16-bit process data objects for
  - Receive, starting from parameter r8750[0]
  - Transmit, starting from parameter r8751[0]
- 32-bit process data objects for
  - Receive, starting from parameter r8760[0]
  - Transmit, starting from parameter r8761[0]

Procedure using the example of drive object 2

1. Select the drive in the project navigator; e.g. "drive_2".
2. In the "Expert list" shortcut menu, call the expert list of drive object 2.
3. Search for parameter r8750[0] with the image of the 16-bit process data objects in the receive buffer. In this example, the following are mapped (see also screenshot below):
   - Control word (6840 hex) in PZD 1
   - Setpoint torque (6871 hex) in PZD 4
4. Search for the r parameter r8760[0] with the image of the 32-bit process data objects in the receive buffer. In this example, the following are mapped (see also screenshot below):
   - Setpoint velocity (68FF hex) in PZD 2+3

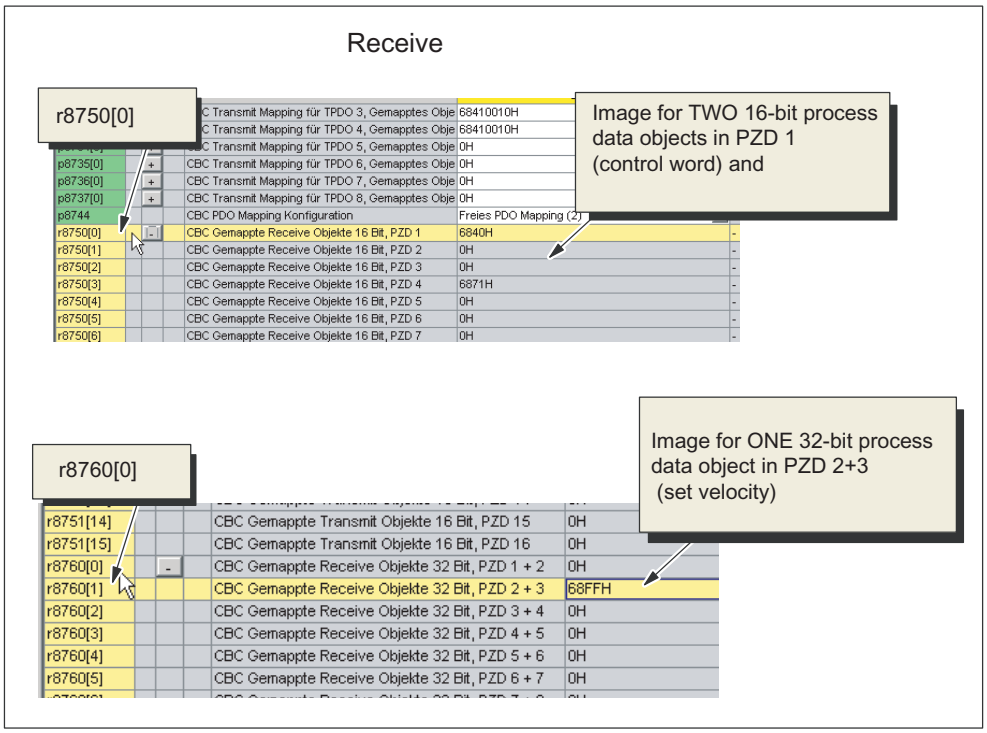


Figure 3-20 Image of process data objects in receive buffer

5. Search for parameter r8751[0] with the image of the 16-bit process data objects in the transmit buffer. In this example, the following are mapped (see also screenshot below):
   - CBC status word (6841 hex) in PZD 1
   - Actual torque (6874 hex) in PZD 4
6. Search for parameter r8761[0] with the image of the 32-bit process data objects in the transmit buffer. In this example, the following are mapped (see also screenshot below):
   - Actual velocity (686C hex) in PZD 2+3
   - Actual position value (6863 hex) in PZD 5+6

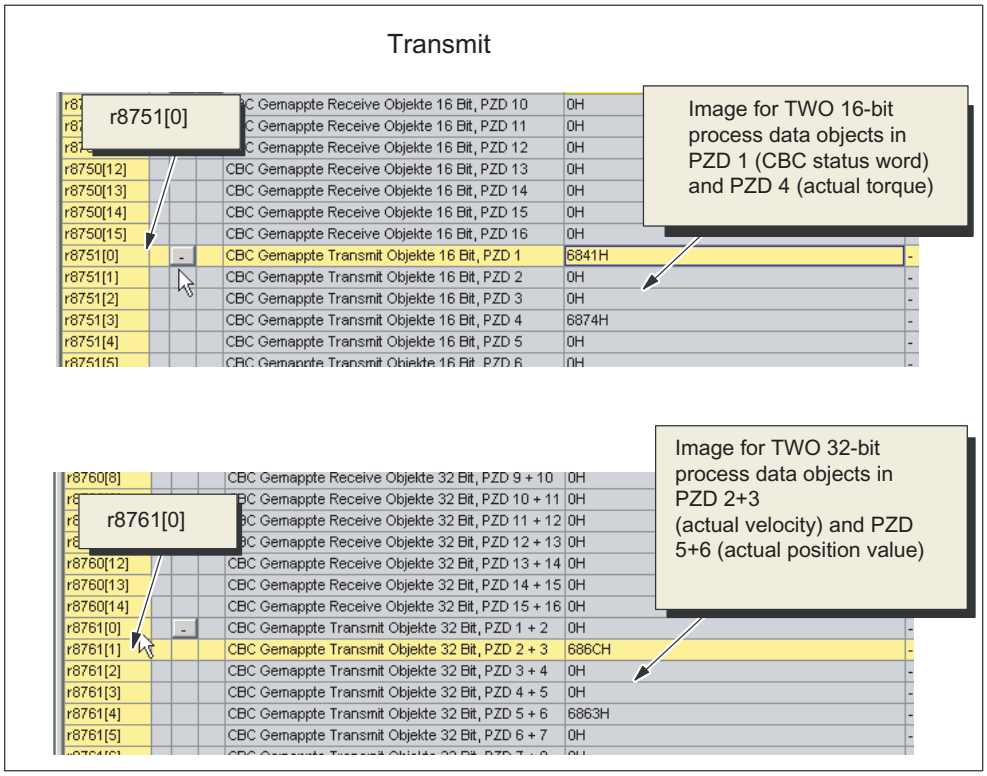


Figure 3-21 Image of process data objects in transmit buffer

7. Select the drive in the project navigator; e.g. "drive_1".
8. In the "Expert list" shortcut menu, call the expert list of drive object 1.

   For receiving, the 16-bit process data objects starting from parameter r8750[0] and the 32-bit process data objects starting from parameter r8760[0] are also available here. Likewise, the 16-bit process data objects starting from parameter r8751[0] and the 32-bit process data objects starting from parameter r8761[0] can be read for transmission purposes.
9. Once you have read the images for both drive objects, you can now interconnect the process data objects in the receive and transmit buffers (see below).

3.6.2 Interconnecting process data for transmission message frames

Precondition

On the basis of the image read in the previous section, note the following:

The following objects are interconnected with each other:

- SINAMICS source and/or target parameters for the process data objects
- The receive or transmit words in the receive/transmit buffer

The excerpts of function diagrams for the receive/transmit buffer, which are shown in the following sections, illustrate:

- How the process data objects in the receive and transmit buffers are distributed to the receive and transmit words.
- Which associated target and source parameters for the receive and transmit words have to be interconnected (highlighted in color).

Note

The SINAMICS source and target parameters, which are interconnected with the receive and transmit words, are listed in the "SINAMICS parameter" column in the "Objects of the drive profile DSP 402" table.

3.6.2.1 Interconnecting the receive buffer

In the receive buffer, interconnect the following for the transmission message frames:

- Control word (PZD 1)
- Setpoint velocity (PZD 2+3)
- Set torque (PZD 4)

Operating steps

For instance, if you interconnect the setpoint velocity process data object in the PZD receive word 2+3 (32 bit), proceed as follows. interconnect the following target and source parameters:

- SINAMICS target parameter for the setpoint velocity (p1155[0] > 32 bit, see Table "CANopen objects of the drive profile DSP402 (Page 152)")

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameters	Trans-mission	Data type	Pre-set values:	Can be read/ written to
...	...	...	...	...	...	...	...
60FF		Target velocity Setpoint velocity	Without ramp-function generator -> p1155[0] With ramp-function generator -> p1070	SDO/PDO	Integer32	-	rw

- Source parameter r8860[1] => 32 bit in the receive buffer (see the following diagram; the source parameter for the setpoint velocity is highlighted in color).

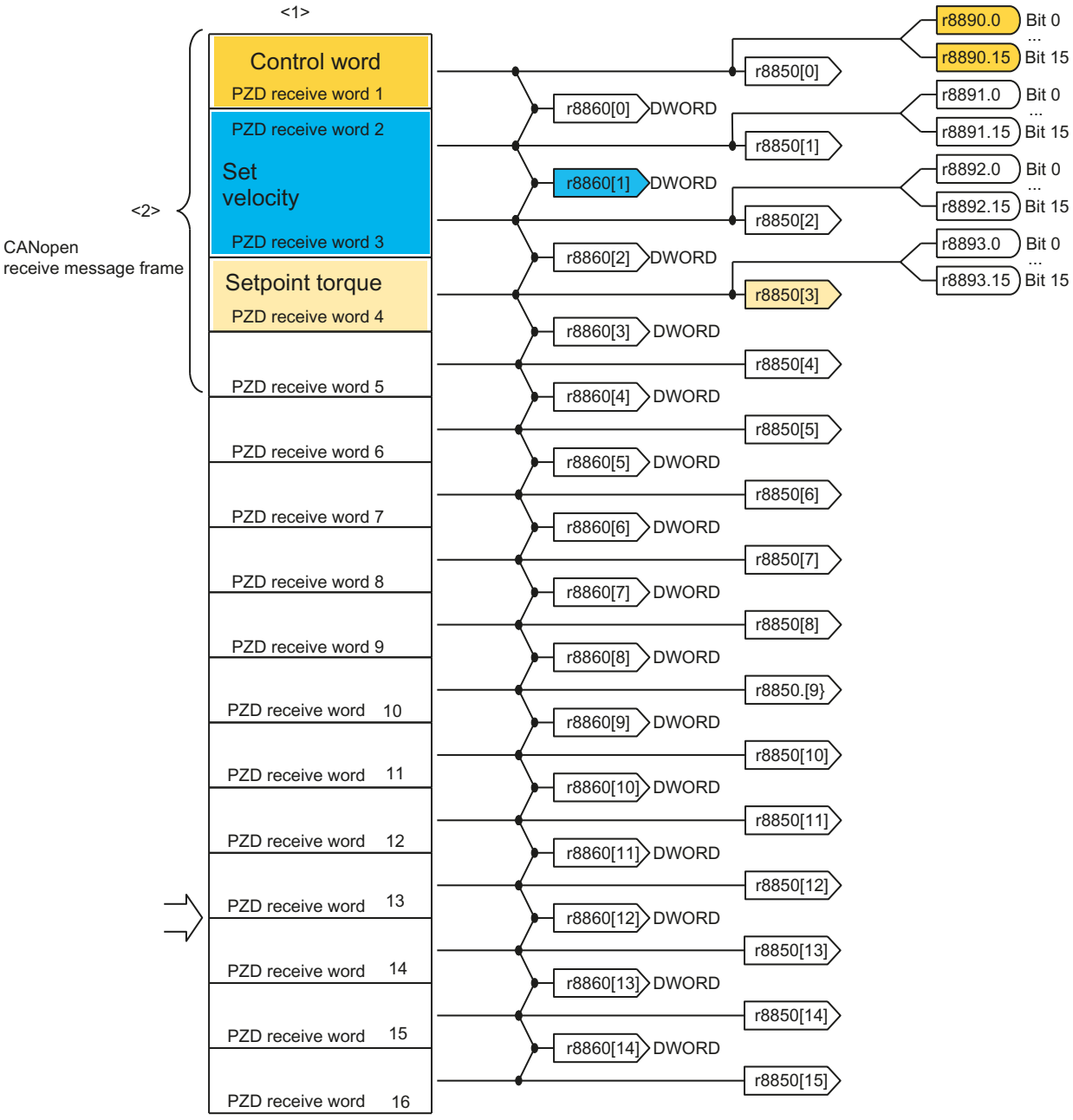


Figure 3-22 Excerpt of function diagram: receive buffer

Note

You can now interconnect the process data objects listed below.

Control word (PZD 1 16 bit)

You automatically interconnect the control word as SINAMICS target parameter with the source parameter r8890 when first commissioning the system.

Example:

1. Select the drive in the project navigator; e.g. "Drive_1".
2. In the "Expert list" shortcut menu, call the expert list of the drive.
3. Search for parameter p8790.
4. Activate option "Interconnection (1)".

   The interconnection is established when you exit the field.

Setpoint velocity (PZD 2+3, 32 bit)

The following table shows the parameters for the setpoint velocity, which has to be interconnected with the corresponding source.

Table 3- 10 Interconnect setpoint velocity

Target (sink)	Source PZD 2+3	Meaning
p1155[0]	r8860[1]	Speed setpoint 1

For the setpoint velocity, data type 32 bit, this is how you interconnect parameter p1155[0] with parameter r8860[1]:

1. In the project navigator, double-click "Drive_1" > "Open-loop/closed-loop control" > "Setpoint addition".
2. You click in the field "Speed setpoint 1" and select the shortcut menu "Further interconnections... ".
3. Interconnect parameter p1155[0] with r8860[1] = PZD 2+3.

Setpoint torque (PZD 4, 16 bit)

For the setpoint torque, the target parameter p1513[0] must be interconnected with the source parameter r8850[3]:

Table 3- 11 Interconnect the setpoint torque

Target (sink)	Source PZD 4	Meaning
p1513[0]	r8850[3]	Additional torque

1. In the project navigator, double-click "Drive_1" > "Open-loop/closed-loop control" > "Torque setpoints".
2. Activate the "Additional torques" tab.
3. You click in the field "Additional torque 2" and select the shortcut menu "Further interconnections... ".
4. Interconnect parameter p1513[0] with r8850[3] = PZD 4.

   To ensure that the setpoint torque becomes active, additional interconnections must be established under the "Torque setpoint" tab.
5. Activate the "Torque setpoints" tab.
6. You click in the field "Speed/torque control" and select the shortcut menu "Further interconnections... ".
7. interconnect p1501[0] with a free bit (e.g. bit 14) from control word (r8890).

3.6.2.2 Interconnecting the transmit buffer

In the transmit buffer, interconnect the following for the transmission message frames:

- CBC status word (PZD 1)
- Actual velocity (PZD 2+3)
- Actual torque (PZD 4)

Operating steps

For instance, if you interconnect CBC status word in PZD transmit word 1 (16 bit), proceed as follows. Interconnect the following source and target parameters:

- SINAMICS source parameters for the CBC status word (r8784 > 16 bit, see Table "CANopen objects of the drive profile DSP402 (Page 152)")

OD index (hex)	Subi ndex (hex)	Object name	SINAMICS parameters	Transmis sion	Data type	Default values	Can be read/ written to
6041		Status word	r8784	PDO/SD O	Unsigned1 6	-	ro

- Target parameter p8851[0] => 16 bit in the transmit buffer (see the following diagram; the target parameter for the CBC status word is color highlighted).

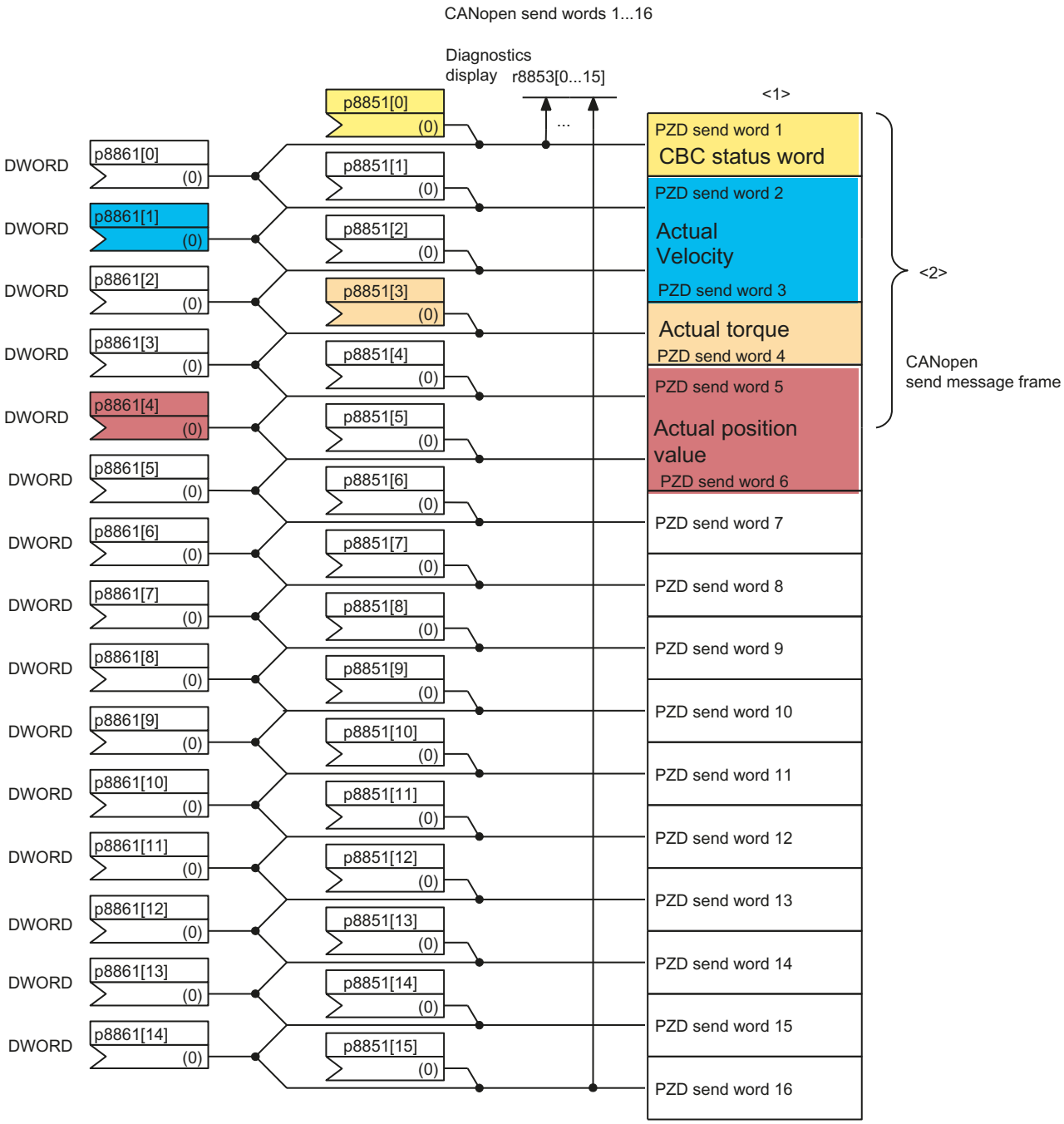


Figure 3-23 Excerpt of function diagram: transmit buffer

Note

You can now interconnect the process data objects listed below.

CBC status word (PZD, 1 16 bit)

For the CBC status word, the target parameter p8851[0] must be interconnected with the source parameter r8784:

Table 3- 12 Interconnecting the CBC status word

Target (sink) PZD 1	Source	Meaning
p8851[0]	r8784	CBC status word

1. Select the drive in the project navigator; e.g. "Drive_1".
2. In the "Expert list" shortcut menu, call the expert list of the drive.
3. Search for parameter p8851[0].
4. Interconnect parameter p8851[0]= PZD 1 with the associated parameter r8784.

Actual velocity (PZD 2+3, 32 bit)

The target parameter p8861[1] must be interconnected with the source parameter r0063 for the actual velocity:

Table 3- 13 Interconnect the actual velocity

Target (sink) PZD 2+3	Source	Meaning
p8861[1]	r0063	Speed actual value

1. Select the drive in the project navigator; e.g. "Drive_1".
2. In the "Expert list" shortcut menu, call the expert list of the drive.
3. Search for parameter p8861[0].
4. Interconnect parameter p8861[1] = PZD 2 + 3 with the associated parameters r0063.

Actual torque (PZD 4, 16 bits)

The target parameter p8851[3] must be interconnected with the source parameter r0080 for the actual torque:

Table 3- 14 Interconnect the actual torque

Target (sink) PZD 4	Source	Meaning
p8851[3]	r0080	Actual torque value

1. Select the drive in the project navigator; e.g. "Drive_1".
2. In the "Expert list" shortcut menu, call the expert list of the drive.
3. Search for parameter p8851[3].
4. Interconnect parameter p8851[3]= PZD 4 with the associated parameter r0080.

Actual position value (PZD 5+6, 32 bit)

The target parameter p8851[3] must be interconnected with the source parameter r0482 for the position actual value:

Table 3- 15 Interconnect the actual position value

Target (sink) PZD 5+6	Source	Meaning
p8861[4]	r0482	Actual position value

1. Select the drive in the project navigator; e.g. "Drive_1".
2. In the "Expert list" shortcut menu, call the expert list of the drive.
3. Search for parameter p8861[4].
4. Interconnect parameter p8861[4] = PZD 5+6 with the associated parameters r0482[0…2] for the corresponding encoder.

3.6.2.3 Interconnecting an additional drive object

To interconnect the next drive object (Single Motor Module 2), repeat the steps described in sections "Interconnecting the receive buffer (Page 104)" and "Interconnecting the transmit buffer (Page 107)".

When carrying out the steps, choose "drive_2" etc. in the project navigator.

3.7 In ONLINE mode, load the projects from the drive unit to the PC/PG and save

Precondition

You are in the ONLINE mode in the STARTER commissioning tool and have completed the initial commissioning procedure.

Procedure

To store the data configured ONLINE in the STARTER project on the PG/PC, proceed as follows:

1. Select the drive "Drive_unit_126" in the project navigator.
2. Click on the symbol "Load CPU/drive unit to PG" .

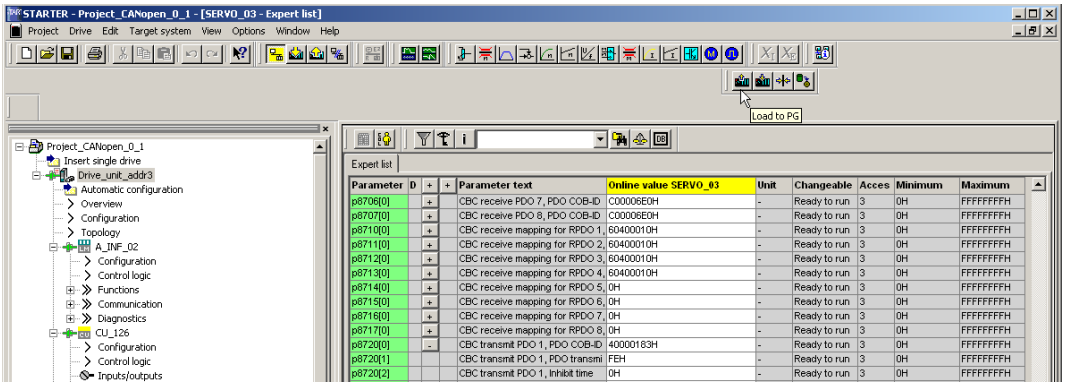


Figure 3-24 Load to PG

3. Click "Yes" at the prompt "Start the load process?".
4. Click in the "The data was successfully loaded" dialog.
5. Click on the symbol "Disconnect from target system" .

6. If prompts are displayed, then click on the following one after the other:
   - Click on "Changes in the drive unit... ".
   - Click on "Save data", for SERVO_3
   - For "Copying from RAM to ROM has been carried out successfully" click on "OK".
   - For "Data will be loaded to the PG, are you sure?" click on "Yes".
   - For "The data was successfully loaded to the PG" click on "OK".
   - Click on "Save data", for SERVO_4.
   - For "Copying from RAM to ROM has been carried out successfully" click on "OK".
   - For "Data will be loaded to the PG, are you sure?" click on "Yes".
   - For "The data was successfully loaded to the PG" click on "OK".

   The STARTER commissioning tool is in the OFFLINE mode.
7. Call the menu "Project > Save under...".

Note

This completes initial commissioning for the CANopen interface.

3.8 SDO access to CANopen process data objects

3.8.1 Standardized CANopen PZD objects

Access to standardized objects

When commissioning, as user you have the responsibility for selecting the appropriate access type and the BICO interconnection.

PDO access

PDO access to mapped standardized CANopen-PZD objects is realized via the BICO interconnection of the corresponding PZD interface parameter. Standardized CANopen-PZD objects are objects of the vendor-specific number range 0x6000 - 0x67FF.

SDO access

The SDO access depends on whether an object is mapped to a PDO:

- Object is mapped in a PDO

  SDO access acts consistently on the corresponding PZD interface parameters
- SETPOINT object not mapped in the PDO

  SDO access acts consistently on the corresponding PZD interface parameters
- ACTUAL VALUE object not mapped in the PDO

  SDO access is realized directly at an actual value parameter.

The following diagrams should clearly show the access operations to standardized CANopen-PZD objects.

SETPOINT

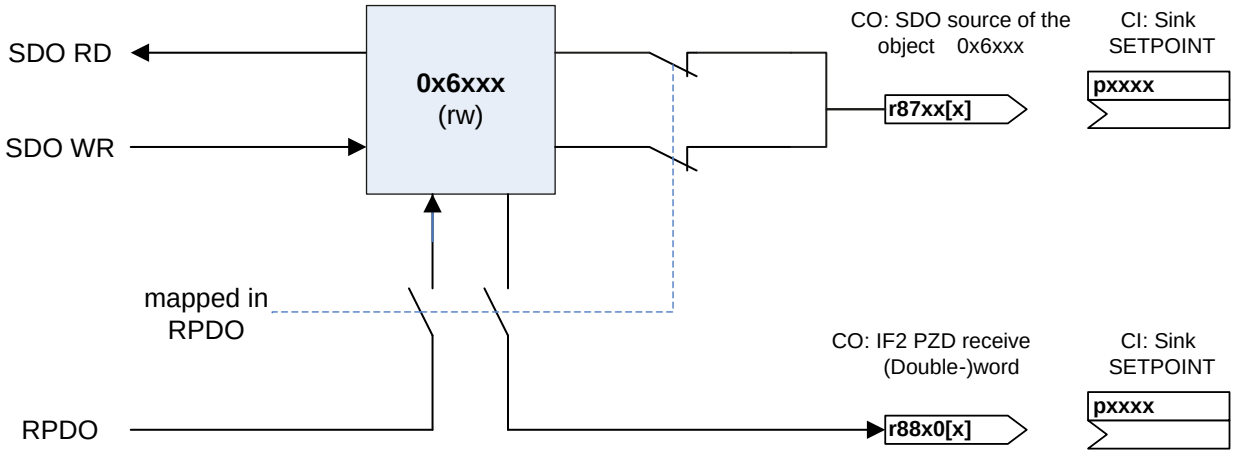


Figure 3-25 Access to standardized CANopen-PZD setpoint objects

ACTUAL VALUE

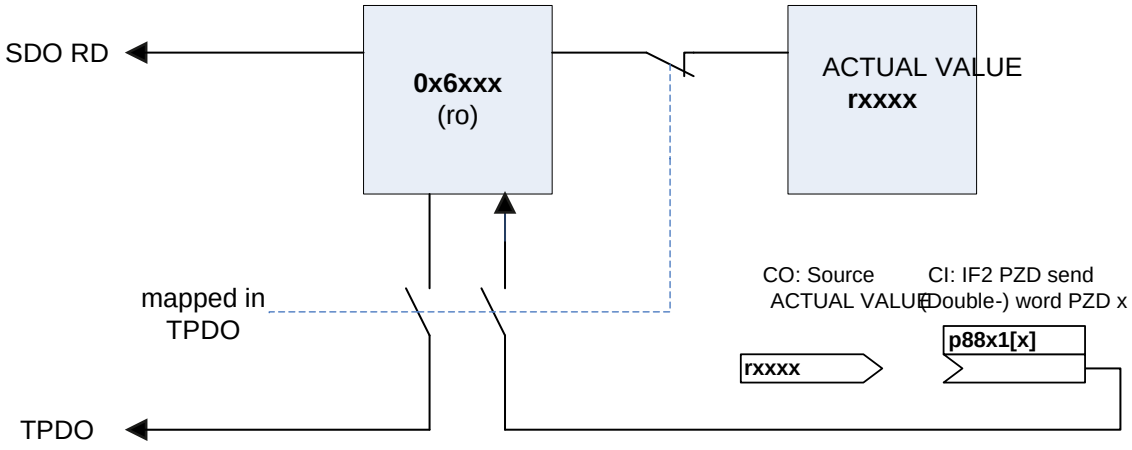


Figure 3-26 Access to standardized CANopen-PZD actual value objects

3.8.2 Free CANopen PZD objects

Access to free CANopen-PZD objects

When commissioning, as user you have the responsibility for selecting the appropriate access type and the BICO interconnection.

PDO access

PDO access to mapped free CANopen-PZD objects is realized via the BICO interconnection of the corresponding PZD interface parameter. Free CANopen-PZD objects are objects of the vendor-specific number range 0x5800 – 0x58xx.

SDO access

The SDO access depends on whether an object is mapped to a PDO:

- Object is mapped in a PDO

  SDO access acts consistently on the corresponding PZD interface parameters

- SETPOINT object not mapped in the PDO

  The SDO access is realized via CO parameters that must be interconnected in the corresponding setpoint sinks. For this reason, each PZD setpoint object requires a CO parameter. If possible, they are grouped in CANopen operating modes.

- ACTUAL VALUE object not mapped in the PDO

  The SDO access is realized via CI parameters that must be interconnected in the corresponding actual value sources. As a consequence, a CI parameter is required for each free PZD actual value object.

The following diagrams clearly show access operations to free PZD objects.

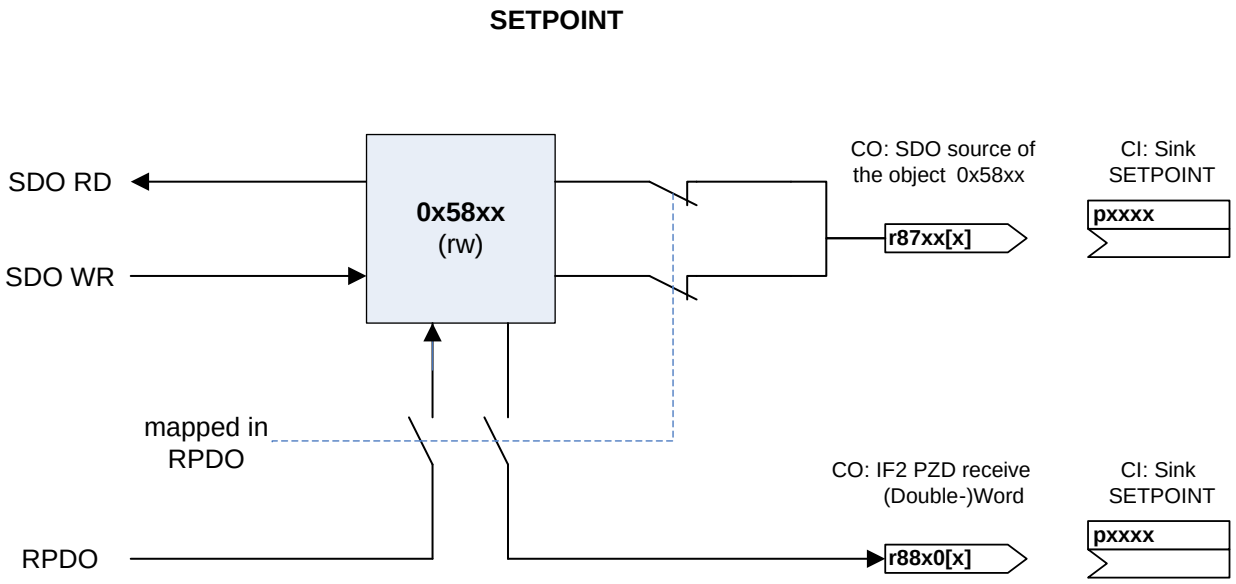


Figure 3-27 Access to free PZD setpoint objects

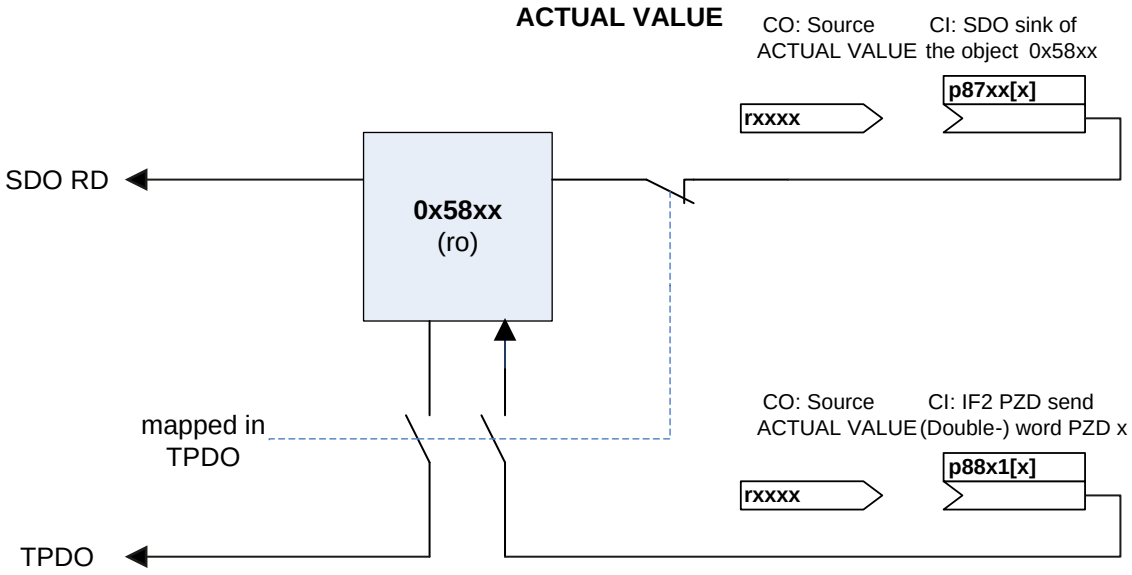


Figure 3-28 Access to free PZD actual value objects

Calculation/conversion setpoints/actual values 4

4.1 Speed setpoint input and evaluation of the speed actual value

The CANopen object "Target Velocity 0x60FF" or "Velocity actual value 0x606C" is accessed using PDO or SDO transfer.

By default, this CANopen object is displayed in the unit "increments/second".

Note

The CANopen object index of a specific drive object is calculated as follows:

- 60FF hex + 800 hex * x (x: drive number 0 ... 7)
- 606C hex + 800 hex * x (x: drive number 0 ... 7)

The following parameters must be taken into account to enter the speed setpoint and evaluate the speed actual value:

- p8798 = CBC speed conversion factor

  The parameter corresponds to the CANopen object 6094 hex.

  The factor converts the required velocity units into the internal velocity units (U/s).

  Factory setting of the velocity unit for CANopen:

  – with encoder -> increments/second

  – without encoder -> revolutions/minute

The internal velocity is calculated as follows:

Speed setpoint

$$n\_set\_intern\ [U/s] = \frac{n\_set\_bus}{p0408 * 2^{p0418}} * \frac{p8798[0]}{p8798[1]}$$

- p0408 = pulse number
- p0418 = fine resolution

Speed setpoint with/without encoder

- With encoder

  The value to be sent to the bus is calculated as follows:

$$n\_set\_bus = n\_set\,[rpm] * \frac{1}{60s} * p0408 * 2^{p0418} * \frac{p8798[1]}{p8798[0]}$$

  If the target velocity should be, e.g.: 3000 rpm, then the following must be sent to the bus to SINAMICS, for an encoder with a pulse number 2048 and a fine resolution of 11:

$$C800000\ hex = \frac{3000\,[rpm]}{60s} * 2048 * 2^{11} * \frac{1}{1}$$

- Without encoder

$$n\_set\_bus = n\_set[rpm] * \frac{p8798[1]}{p8798[0]}$$

Process data interface

If the incoming speed setpoint is read in parameter r8860[x], this is formed from the following calculation:

$$r8860[x] = \frac{n\_set\_bus}{p0408 * 2^{p0418}} * 60s * \frac{4000\ 0000\ hex}{p2000}$$

$$r8860[x] = n\_set\,[rpm] * \frac{4000\ 0000\ hex}{p2000}$$

In order to see the process value specified via the bus referred to 4000 000 hex in r8860[x], then p2000 must be set to p0311.

Speed actual value

Speed actual value with/without encoder

- With encoder

$$n\_act[rpm] = n\_act\_bus * 60s * \frac{1}{p0408 * 2^{p0418}} * \frac{p8798[0]}{p8798[1]}$$

- Without encoder

$$n\_act[rpm] = n\_act\_bus * \frac{p8798[0]}{p8798[1]}$$

Process data interface

If the incoming speed setpoint is read in parameter r8863[x], this is formed from the following calculation:

$$r8863[x] = \frac{n\_act\_bus}{p0408 * 2^{p0418}} * 60s * \frac{4000\ 0000\ hex}{p2000}$$

$$r8863[x] = n\_act\ [rpm] * \frac{4000\ 0000\ hex}{p2000}$$

In order to see the process value specified via the bus referred to 4000 000 hex in r8863[x], then p2000 must be set to p0311.

Note

For some operating modes, an encoder is commissioned anyway, for example: Encoderless closed-loop speed control (p1300 = 20) or encoderless closed-loop torque control (p1300 = 22). For these operating modes, the actual velocity is calculated using the formula "with encoder".

4.2 Torque setpoint input and evaluation of the torque actual value

The CANopen object "Target Torque 0x6071" or "Torque actual value 0x6077" is accessed using PDO or SDO transfer.

The CANopen objects are displayed as default as per mille (1/1000).

Note

The CANopen object index of a specific drive object is calculated as follows:

6071 hex + 800 hex * x (x: drive number 0 ... 7)

6077 hex + 800 hex * x (x: drive number 0 ... 7)

The following parameters must be taken into account to enter the torque setpoint and evaluate the torque actual value:

- r0333 = rated motor torque

Torque setpoint

The value to be sent to the bus should be calculated as follows:

$$\text{m\_set\_bus [per mille]} = \frac{\text{m\_set [Nm]}}{\text{r0333 [Nm]}} * 1000$$

Process data interface

If the incoming torque setpoint is read in parameter r8850[x], then this is displayed from the calculation per mille input from the bus x the value of r0333 referred to p2003.

Example:

$$\text{r8850[x]} = \frac{\text{m\_set\_bus [per mille]} * \text{r0333 [Nm]}}{1000} * \frac{\text{4000 hex}}{\text{p2003 [Nm]}}$$

In order to see the process value specified via the bus referred to 4000 hex in r8850[x], then p2003 must be set to r0333.

Actual torque value

The torque actual value is calculated as follows:

$$\text{m\_act [Nm]} = \frac{\text{m\_act\_bus [per mille]}}{1000} * \text{r0333 [Nm]}$$

Process data interface

If the outgoing torque actual value is read in parameter r8853[x], this is formed from the following calculation:

$$\text{r8853[x]} = \frac{\text{m\_act\_bus [per mille]} * \text{r0333 [Nm]}}{1000} * \frac{\text{4000 hex}}{\text{p2003 [Nm]}}$$

$$\text{r8853[x]} = \frac{\text{m\_ist [Nm]} * 1000}{\text{r0333 [Nm]}} * \frac{\text{4000 hex}}{\text{p2003 [Nm]}}$$

In order to see the process value specified via the bus referred to 4000 hex in r8853[x], then p2003 must be set to r0333.

Diagnosis

5

CANopen supports a standardized system for detecting, describing, and signaling device errors with the following equipment:

- An alarm object for each drive unit ("Emergency object")
- A device-internal error list ("predefined error field")
- An error register

Note

See also the following standards:

- CiA 301 (Application Layer and Communication Profile)
- CiA 402 (Device Profile for Drives and Motion Control)

5.1 Alarm object (Emergency Object)

Error statuses for each drive unit are signaled via the high-priority 8-byte emergency object (error message).

You can find the relevant parameters here:

- in the object directory index 1014 hex (COB ID EMCY) and 1015 hex (inhibit time EMCY)
- for SINAMICS in parameter p8603 of the Control Unit.

When an error occurs, an error message frame (emergency message frame) is sent to the identifier set in object "1014 hex".

Every error is assigned an error code in CANopen, with the error codes being further subdivided into, for example, current errors, voltage errors, etc.

Emergency message frame

When an error occurs, the CANopen drive unit automatically transmits an emergency message frame asynchronously. The emergency message frame is structured as follows (see table below).

Table 5- 1 Structure of the emergency message frame

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
CANopen Errorcode		CANopen Error Register	SINAMICS fault number		Drive object number	Reserved	Reserved

The CANopen error code is in byte 0 and 1 (refer to the following section).

The codings for the CANopen error register are located in byte 2 (refer to the following section).

Byte 5 contains the number of the drive object from which the error originated.

Alarms that do not result in a shutdown are indicated only by the alarm bit or fault bit in the status word and do not trigger an emergency message frame. Faults trigger an emergency message frame and cause the drives to be shut down. The master can then read the alarm or fault during a period of free bus time. The fault message frame can be suppressed by setting bit 31 in object 1014 hex.

5.2 Drive-unit-internal error list ("predefined error field")

The drive unit internal error list ("predefined error field") can be read via the following objects:

- Object directory index 1003 hex
- SINAMICS parameter p8611 of the Control Unit

Fault list

This list contains the faults diagnosed in a drive unit, which have not been acknowledged, and pending alarms in the CANopen alarm number range 8700-8799. The faults are defined using a fault code (error code) and device-specific supplementary information in the sequence that they occurred. As soon as a fault is acknowledged or an alarm is resolved, they are deleted from the drive device internal error list.

All drive objects in the selected Control Unit are acknowledged by writing the subindex 0, as well as for object directory index 1003 hex and for the parameter p8611 with value 0.

The following table describes the CANopen error code that is evaluated with SINAMICS (in the emergency message frame, byte 0/1).

Table 5- 2 CANopen Errorcode

CANopen Errorcode	Meaning	Triggered by SINAMICS
0000 hex	No error present	Successful acknowledgment of all faults or all the alarms are gone
1000 hex	CAN Error 1	All other SINAMICS faults
1001 hex	CAN Error 2	All other CANopen alarms in the alarm number range F08700 up to F08799
8110 hex	CAN overflow, message lost	CBC: Message frame loss (A(N)08751) [alarm]
8120 hex	CAN Error Passive	CBC: Error number for Error Passive exceeded (A08752) [alarm]
8130 hex	CAN Life Guard Error	CBC: Communications error, alarm value 2 F08700(A) [fault/alarm]

Note

For other SINAMICS alarms, an emergency message frame is not sent.

5.3 Error register

The 1-byte error register can be read via the following objects:

- Object directory index 1001 hex
- SINAMICS parameter r8601 of the Control Unit

Error register

The register displays any drive unit errors that have occurred and their type.

The following table describes the CANopen error register that is evaluated with SINAMICS (in the emergency message frame, bytes 1 and 2).

Table 5- 3 Error register

Error register	Meaning	Triggered by SINAMICS
Bit 0	generic error	Set for every alarm that CAN identifies.
Bit 4	communication error	For CAN communication alarms set, i.e. for alarms in the number range 08700-08799.
Bit 7	manufacturer error	Set for all SINAMICS errors outside the CAN communication alarm number range.

Operating modes

6

CANopen supports velocity-related and torque-related operating modes:

- Velocity Mode (velocity-related)
  Simple velocity control with ramps and the relevant objects (e.g. frequency converters or current converters)
- Profile Velocity Mode (velocity-related)
  Velocity control, closed-loop speed control and the relevant objects.
- Profile Torque Mode (torque-related)
  Torque control and the relevant objects

Overview of CANopen objects and operating modes

Table 6- 1 CANopen objects and operating modes

Operating modes	Preferred access via objects of the CANopen operating mode	0x6060 - operating mode	0x6061 - operating mode display	0x6502 - supported drive type	p1300 - open-loop/closed-loop control operating mode
Velocity Mode	Velocity mode	2	2	Bit 1	0
M* 1		-1	-1	Bit16	1
M* 2		-2	-2	Bit17	2
M* 3		-3	-3	Bit18	3
M* 4		-4	-4	Bit19	4
M* 5		-5	-5	Bit20	5
M* 6		-6	-6	Bit21	6
M* 7		-7	-7	Bit22	7
M* 8		-15	-15	Bit23	15
M* 9	M*	-18	-18	Bit22	18
M* 10	Velocity mode	-19	-19	Bit25	19
M* 11	Profile Velocity Mode	-20	-20	Bit26	20
Profile velocity mode		3	3	Bit2	21
M* 12	Profile Torque Mode	-22	-22	Bit27	22
Profile torque mode		4	4	Bit3	23

Operating modes	Preferred access via objects of the CANopen operating mode	0x6060 - operating mode	0x6061 - operating mode display	0x6502 - supported drive type	p1300 - open-loop/closed-loop control operating mode
No mode change/no mode assigned	-	0	0	-	-
M* = Manufacturer-specific Operation Mode					

CANopen object 0x6502

The CANopen object "0x6502 Supported drive modes" indicates which CANopen operating modes can be selected in the corresponding drive units and in the existing commissioning state via the object "0x6060", via SDO and/or via PDO access.

CANopen object 0x6060

By writing the CANopen object "0x6060 Modes of operation", the required operating mode can be selected.

The object always displays the requested mode, even if this is not effective, because it is not supported for example from the drive object.

Also writing via SDO is always successful, if a value is written within its data type.

If an operating mode cannot be selected, for example, because it is not supported, then the old operating mode remains effective.

By reading the object "0x6061" the controller itself must check as to whether the requested operating mode was able to be successfully selected, and is therefore effective.

Access via RPDO

The object can be mapped, and therefore transferred via PDO unconfirmed. Presently, for SINAMICS, this process data access within the scope of the CAN sampling time only makes sense for switching between speed and torque control; this is because only these operating modes are supported by CANopen, and for SINAMICS, can presently be switched over in operation.

All switchover operations of the known operating modes, lead to the SINAMICS parameter channel that runs in the background.

If the object is mapped in an RPDO, then also an SDO access consistently acts on the corresponding PZD interface parameters and no longer on p1300. The diagram below should clearly illustrate this.

All switchover operations of the other operating modes, lead to the SINAMICS parameter channel, which runs in the background. It does not make sense to create a PDO access for this purpose. If the object is mapped in an RPDO, then also an SDO access consistently acts on the corresponding PZD interface parameters and no longer on p1300. The diagram below should clearly illustrate this.

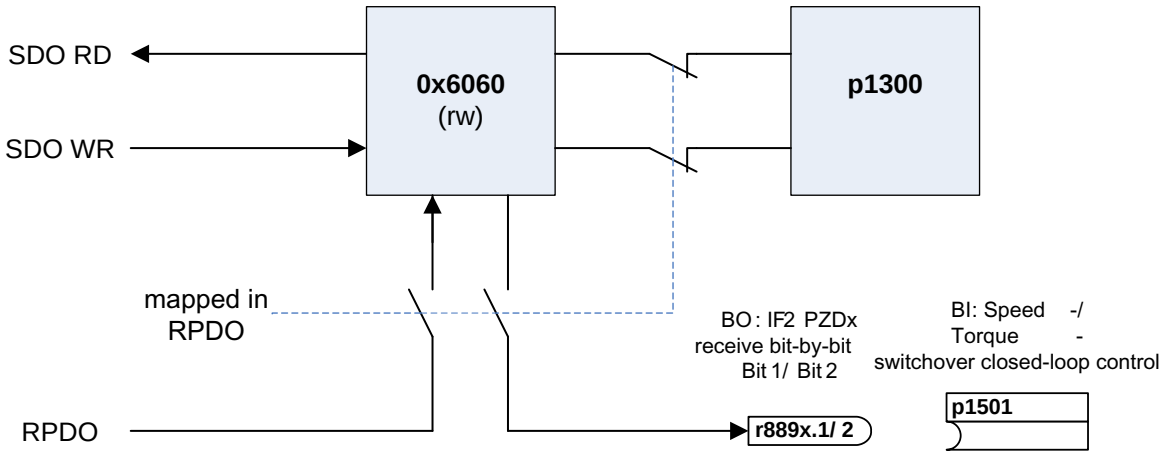


Figure 6-1 Accessing operating modes

As user, you are responsible when commissioning for selecting the appropriate access type.

Access via SDO

For an SDO access, a value is written to the object 0x6060 in parameter p1300. You can take the corresponding value from the Table 6-1 CANopen objects and operating modes (Page 129).

CANopen object 0x6061

The CANopen object "0x6061 modes of operation display" indicates the currently active operating mode.

The object can be mapped, and therefore transferred via PDO unconfirmed.

If the object is mapped in a TPDO, then an SDO access consistently acts on the corresponding PZD interface parameters. When commissioning, the new CO parameter "r8762 CAN operating mode display" can be interconnected in the corresponding sink of the PZD interface IF2.

The value of parameter r8762 is formed as follows:

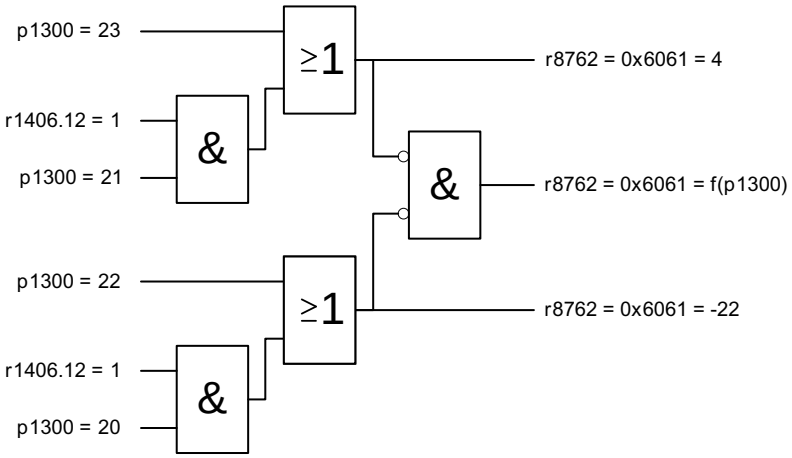


Figure 6-2 Operating mode display

If the transmission mode of the corresponding TPDO is configured to be asynchronous, when the operating mode has been successfully changed, it is immediately sent, therefore informing the control that the operating mode has been successfully changed.

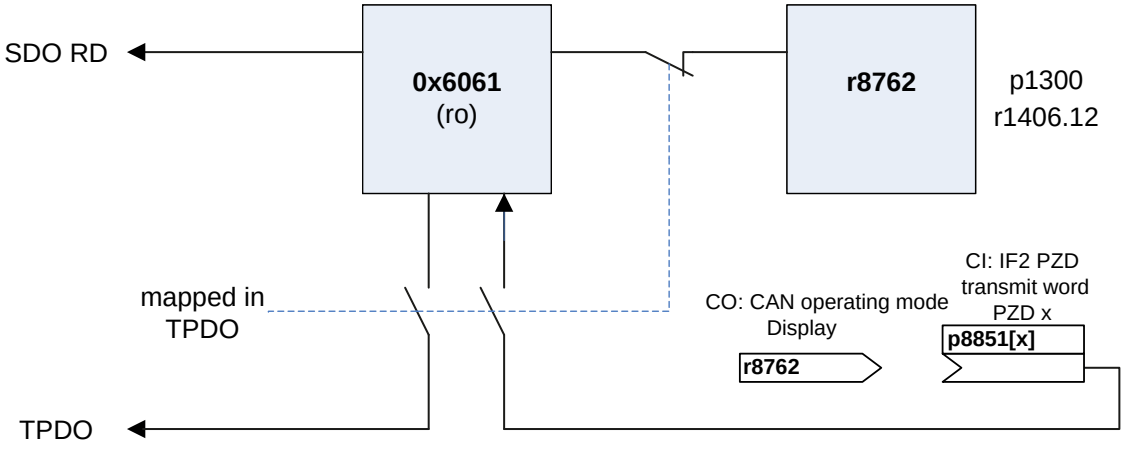


Figure 6-3 Accessing the operating mode display

Communication objects

7

This section contains a table of the objects (data values) that are used in SINAMICS for communication via the CANopen interface.

- Control Unit communication objects independent of the drive
- Drive-dependent communication objects
- Manufacturer-specific objects
- Objects in drive profile DSP402

The objects are stored in an object directory.

7.1 Control Unit communication objects independent of the drive

The following table lists the object directory with the index of the individual drive-independent Control Unit communication objects. The column "SINAMICS parameter" shows the parameter numbers and their allocation for SINAMICS.

Table 7- 1 Control Unit communication objects independent of the drive

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Default values	Can be read/ written to
1000		Device type	r8600	SDO	Unsigned32	–	ro
1001		Error register	r8601	SDO	Unsigned8	–	ro
1003	0...52 hex	Predefined error field	p8611[0...82]	SDO	Unsigned32	0	ro/rw
	0	Number of errors	p8611.0	SDO	Unsigned32	0	rw
	1	Number of module	p8611.1	SDO	Unsigned32	0	ro
	2	Number of errors: module 1	p8611.2	SDO	Unsigned32	0	ro
	3-A	Standard error field: module 1	p8611.3-p8611.10	SDO	Unsigned32	0	ro
	B	Number of errors: module 2	p8611.11	SDO	Unsigned32	0	ro
	C-13	Standard error field: module 2	p8611.12-p8611.19	SDO	Unsigned32	0	ro
	14	Number of errors: module 3	p8611.20	SDO	Unsigned32	0	ro
	15-1C	Standard error field: module 3	p8611.21-p8611.28	SDO	Unsigned32	0	ro
	1D	Number of errors: module 4	p8611.29	SDO	Unsigned32	0	ro
	1E-25	Standard error field: module 4	p8611.30-p8611.37	SDO	Unsigned32	0	ro
	26	Number of errors: module 5	p8611.38	SDO	Unsigned32	0	ro
	27-2E	Standard error field: module 5	p8611.39-p8611.46	SDO	Unsigned32	0	ro
	2F	Number of errors: module 6	p8611.47	SDO	Unsigned32	0	ro
	30-37	Standard error field: module 6	p8611.48-p8611.55	SDO	Unsigned32	0	ro
	38	Number of errors: module 7	p8611.56	SDO	Unsigned32	0	ro
	39-40	Standard error field: module 7	p8611.57-p8611.64	SDO	Unsigned32	0	ro
	41	Number of errors: module 8	p8611.65	SDO	Unsigned32	0	ro
	42-49	Standard error field: module 8	p8611.66-p8611.73	SDO	Unsigned32	0	ro
	4A	Number of Control Unit faults	p8611.74	SDO	Unsigned32	0	ro

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Default values	Can be read/ written to
	4B-52	Standard Control Unit Error Field	p8611.75-p8611.82	SDO	Unsigned32	0	ro
1005		SYNCH COB ID	p8602	SDO	Unsigned32	128	rw
1008		Manufacturer device name	–	SDO	–	–	–
100A		Manufacturer software version	r0018	SDO	Unsigned32	–	ro
100C		Guard time	p8604.0	SDO	Unsigned16	0	rw
100D		Lifetime factor	p8604.1	SDO	Unsigned16	0	rw
1010		Store parameters	p0977	SDO	Unsigned16	0	rw
	0	Largest subindex supported	–	SDO	–	–	–
	1	Save all parameters	p0977	SDO	Unsigned16	0	rw
	2	Save communication parameters (0x1000-0x1fff)	p0977	SDO	Unsigned16	0	rw
	3	Save application-related parameters (0x6000-0x9fff)	p0977	SDO	Unsigned16	0	rw
1011		Restore default parameters	p0976	SDO	Unsigned16	0	rw
	0	Largest subindex supported		SDO			
	1	Restore all default parameters	p0976	SDO	Unsigned16	0	rw
	2	Restore communication default parameters (0x1000-0x1fff)	p0976	SDO	Unsigned16	0	rw
	3	Restore application default parameters (0x6000-0x9fff)	p0976	SDO	Unsigned16	0	rw
1014		COB ID emergency	p8603	SDO	Unsigned32	0	rw
1017		Producer heartbeat time	p8606	SDO	Unsigned16	0	rw
1018		Identy Object	r8607[0...3]		Unsigned32	–	ro
	0	Number of entries	–	SDO	–	–	–
	1	Vendor ID	r8607.0	SDO	Unsigned32	–	ro
	2	Product code	r8607.1	SDO	Unsigned32	–	ro
	3	Revision number	r8607.2	SDO	Unsigned32	–	ro
	4	Serial number	r8607.3	SDO	Unsigned32	0	ro
1027		Module list	–	–	–	–	–
	0	Number of entries	r0102	SDO	Unsigned16	–	ro
	1-8	Module ID	p0107[0...15]	SDO	Integer16	0	rw
1029		Error behavior	–	–	–	–	–

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Default values	Can be read/ written to
	0	No. of error classes	–	SDO	–	–	–
	1	Communication Error	p8609.0	SDO	Unsigned32	1	rw
	2	Device profile or manufacturer-specific error	p8609.1	SDO	Unsigned32	1	rw
1200		1st server SDO parameter	–	–	–	–	–
	0	Number of entries	–	SDO	–	–	–
	1	COB ID client -> server (rx)	r8610.0	SDO	Unsigned32	–	ro
	2	COB ID server -> client (tx)	r8610.1	SDO	Unsigned32	–	ro
1201		Drive object server SDO parameter	–	–	–	–	–
	0	Number of entries	–	SDO	–	–	ro
	1	COB ID client -> server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB ID server -> client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw

7.2 Drive-dependent communication objects

Eight transmit/receive PDO can be parameterized for each drive, whereby a total of 25 receive PDO must not be exceeded.

Each PDO contains:

- Communication parameters
- Mapping parameters (max. 8 bytes/4 words/64 bits).

Rule

In the following tables, the first PDO is highlighted in **bold** to indicate that the communication and mapping parameters for one PDO are related.

The "predefined connection set" column contains the predefined values for the "predefined connection set".

7.2.1 Communication objects receive PDO

Overview

The following table lists the object directory with the index of the individual drive-dependent communication objects for the receive PDO of the first drive object:

Table 7- 2 Drive-dependent communication objects for receive PDO

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
1400		**Receive PDO 1 communication parameter**	–	–	–	–	–
	0	**Largest subindex supported**	–	**SDO**	**Unsigned8**	**2**	**ro**
	1	**COB ID used by PDO**	**p8700.0**	**SDO**	**Unsigned32**	**200 hex + node ID**	**rw**
	2	**Transmission type**	**p8700.1**	**SDO**	**Unsigned8**	**FE hex**	**rw**
1401		Receive PDO 2 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	2	ro
	1	COB ID used by PDO	p8701.0	SDO	Unsigned32	300 hex + node ID	rw
	2	Transmission type	p8701.1	SDO	Unsigned8	FE hex	rw
1402		Receive PDO 3 communication parameter	–	–	–	–	–

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
	0	Largest subindex supported	–	SDO	Unsigned8	2	ro
	1	COB ID used by PDO	p8702.0	SDO	Unsigned32	400 hex + node ID	rw
	2	Transmission type	p8702.1	SDO	Unsigned8	FE hex	rw
1403		Receive PDO 4 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	2	ro
	1	COB ID used by PDO	p8703.0	SDO	Unsigned32	500 hex + node ID	rw
	2	Transmission type	p8703.1	SDO	Unsigned8	FE hex	rw
1404		Receive PDO 5 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	2	ro
	1	COB ID used by PDO	p8704.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	Transmission type	p8704.1	SDO	Unsigned8	FE hex	rw
1405		Receive PDO 6 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	2	ro
	1	COB ID used by PDO	p8705.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	Transmission type	p8705.1	SDO	Unsigned8	FE hex	rw
1406		Receive PDO 7 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	2	ro
	1	COB ID used by PDO	p8706.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	Transmission type	p8706.1	SDO	Unsigned8	FE hex	rw
1407		Receive PDO 8 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	2	ro
	1	COB ID used by PDO	p8707.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	Transmission type	p8707.1	SDO	Unsigned8	FE hex	rw
1600		**Receive PDO 1 mapping parameter**	–	–	–	–	–

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
	0	**Number of mapped application objects in PDO**	–	**SDO**	**Unsigned8**	**1**	**ro**
	1	**PDO mapping for the first application object to be mapped**	**p8710.0**	**SDO**	**Unsigned32**	**6040 hex**	**rw**
	2	**PDO mapping for the second application object to be mapped**	**p8710.1**	**SDO**	**Unsigned32**	**0**	**rw**
	3	**PDO mapping for the third application object to be mapped**	**p8710.2**	**SDO**	**Unsigned32**	**0**	**rw**
	4	**PDO mapping for the fourth application object to be mapped**	**p8710.3**	**SDO**	**Unsigned32**	**0**	**rw**
1601		Receive PDO 2 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	2	ro
	1	PDO mapping for the first application object to be mapped	p8711.0	SDO	Unsigned32	6040 hex	rw
	2	PDO mapping for the second application object to be mapped	p8711.1	SDO	Unsigned32	60FF hex	rw
	3	PDO mapping for the third application object to be mapped	p8711.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8711.3	SDO	Unsigned32	0	rw
1602		Receive PDO 3 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	2	ro
	1	PDO mapping for the first application object to be mapped	p8712.0	SDO	Unsigned32	6040 hex	rw

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
	2	PDO mapping for the second application object to be mapped	p8712.1	SDO	Unsigned32	6071 hex	rw
	3	PDO mapping for the third application object to be mapped	p8712.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8712.3	SDO	Unsigned32	0	rw
1603		Receive PDO 4 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	3	ro
	1	PDO mapping for the first application object to be mapped	p8713.0	SDO	Unsigned32	6040 hex	rw
	2	PDO mapping for the second application object to be mapped	p8713.1	SDO	Unsigned32	60FF hex	rw
	3	PDO mapping for the third application object to be mapped	p8713.2	SDO	Unsigned32	6071 hex	rw
	4	PDO mapping for the fourth application object to be mapped	p8713.3	SDO	Unsigned32	0	rw
1604		Receive PDO 5 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	0	ro
	1	PDO mapping for the first application object to be mapped	p8714.0	SDO	Unsigned32	0	rw
	2	PDO mapping for the second application object to be mapped	p8714.1	SDO	Unsigned32	0	rw
	3	PDO mapping for the third application object to be mapped	p8714.2	SDO	Unsigned32	0	rw

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
	4	PDO mapping for the fourth application object to be mapped	p8714.3	SDO	Unsigned32	0	rw
1605		Receive PDO 6 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	0	ro
	1	PDO mapping for the first application object to be mapped	p8715.0	SDO	Unsigned32	0	rw
	2	PDO mapping for the second application object to be mapped	p8715.1	SDO	Unsigned32	0	rw
	3	PDO mapping for the third application object to be mapped	p8715.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8715.3	SDO	Unsigned32	0	rw
1606		Receive PDO 7 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	0	ro
	1	PDO mapping for the first application object to be mapped	p8716.0	SDO	Unsigned32	0	rw
	2	PDO mapping for the second application object to be mapped	p8716.1	SDO	Unsigned32	0	rw
	3	PDO mapping for the third application object to be mapped	p8716.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8716.3	SDO	Unsigned32	0	rw

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
1607		Receive PDO 8 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	0	ro
	1	PDO mapping for the first application object to be mapped	p8717.0	SDO	Unsigned32	0	rw
	2	PDO mapping for the second application object to be mapped	p8717.1	SDO	Unsigned32	0	rw
	3	PDO mapping for the third application object to be mapped	p8717.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8717.3	SDO	Unsigned32	0	rw

Note

Each additional drive begins with the description in an offset of 40 hex.

7.2.2 Communication objects send PDO

Overview

The following table lists the object directory with the index of the individual drive-dependent communication objects for the transmit PDO of the first drive object:

Table 7- 3 Drive-dependent communication objects for transmit PDO

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
1800		Transmit PDO 1 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	5	ro

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
	1	**COB ID used by PDO**	**p8720.0**	**SDO**	**Unsigned32**	**180 hex + node ID**	**rw**
	2	**Transmission type**	**p8720.1**	**SDO**	**Unsigned8**	**FE hex**	**rw**
	3	**Inhibit time**	**p8720.2**	**SDO**	**Unsigned16**	**0**	**rw**
	4	**Compatibility entry**	**p8720.3**	**SDO**	**Unsigned8**	**3**	**rw**
	5	**Event timer**	**p8720.4**	**SDO**	**Unsigned16**	**0**	**rw**
1801		Transmit PDO 2 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	5	ro
	1	COB ID used by PDO	p8721.0	SDO	Unsigned32	280 hex + node ID	rw
	2	Transmission type	p8721.1	SDO	Unsigned8	FE hex	rw
	3	Inhibit time	p8721.2	SDO	Unsigned16	0	rw
	4	Compatibility entry	p8721.3	SDO	Unsigned8	0	rw
	5	Event timer	p8721.4	SDO	Unsigned16	0	rw
1802		Transmit PDO 3 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	5	–
	1	COB ID used by PDO	p8722.0	SDO	Unsigned32	380 hex + node ID	rw
	2	Transmission type	p8722.1	SDO	Unsigned8	FE hex	rw
	3	Inhibit time	p8722.2	SDO	Unsigned16	0	rw
	4	Compatibility entry	p8722.3	SDO	Unsigned8	0	rw
	5	Event timer	p8722.4	SDO	Unsigned16	0	rw
1803		Transmit PDO 4 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	5	ro
	1	COB ID used by PDO	p8723.0	SDO	Unsigned32	480 hex + node ID	rw
	2	Transmission type	p8723.1	SDO	Unsigned8	FE hex	rw
	3	Inhibit time	p8723.2	SDO	Unsigned16	0	rw
	4	Compatibility entry	p8723.3	SDO	Unsigned8	0	rw
	5	Event timer	p8723.4	SDO	Unsigned16	0	rw
1804		Transmit PDO 5 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	5	ro

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
	1	COB ID used by PDO	p8724.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	Transmission type	p8724.1	SDO	Unsigned8	FE hex	rw
	3	Inhibit time	p8724.2	SDO	Unsigned16	0	rw
	4	Compatibility entry	p8724.3	SDO	Unsigned8	0	rw
	5	Event timer	p8724.4	SDO	Unsigned16	0	rw
1805		Transmit PDO 6 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	5	ro
	1	COB ID used by PDO	p8725.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	Transmission type	p8725.1	SDO	Unsigned8	FE hex	rw
	3	Inhibit time	p8725.2	SDO	Unsigned16	0	rw
	4	Compatibility entry	p8725.3	SDO	Unsigned8	0	rw
	5	Event timer	p8725.4	SDO	Unsigned16	0	rw
1806		Transmit PDO 7 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	5	ro
	1	COB ID used by PDO	p8726.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	Transmission type	p8726.1	SDO	Unsigned8	FE hex	rw
	3	Inhibit time	p8726.2	SDO	Unsigned16	0	rw
	4	Compatibility entry	p8726.3	SDO	Unsigned8	0	rw
	5	Event timer	p8726.4	SDO	Unsigned16	0	rw
1807		Transmit PDO 8 communication parameter	–	–	–	–	–
	0	Largest subindex supported	–	SDO	Unsigned8	5	ro
	1	COB ID used by PDO	p8727.0	SDO	Unsigned32	C000 06E0 hex	rw
	2	Transmission type	p8727.1	SDO	Unsigned8	FE hex	rw
	3	Inhibit time	p8727.2	SDO	Unsigned16	0	rw
	4	Compatibility entry	p8727.3	SDO	Unsigned8	0	rw
	5	Event timer	p8727.4	SDO	Unsigned16	0	rw
1A00		**Transmit PDO 1 mapping parameter**	–	–	–	–	–

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
	0	**Number of mapped application objects in PDO**	–	**SDO**	**Unsigned8**	**1**	**ro**
	1	**PDO mapping for the first application object to be mapped**	**p8730.0**	**SDO**	**Unsigned32**	**6041 hex**	**rw**
	2	**PDO mapping for the second application object to be mapped**	**p8730.1**	**SDO**	**Unsigned32**	**0**	**rw**
	3	**PDO mapping for the third application object to be mapped**	**p8730.2**	**SDO**	**Unsigned32**	**0**	**rw**
	4	**PDO mapping for the fourth application object to be mapped**	**p8730.3**	**SDO**	**Unsigned32**	**0**	**rw**
1A01		Transmit PDO 2 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	2	ro
	1	PDO mapping for the first application object to be mapped	p8731.0	SDO	Unsigned32	6041 hex	rw
	2	PDO mapping for the second application object to be mapped	p8731.1	SDO	Unsigned32	606C hex	rw
	3	PDO mapping for the third application object to be mapped	p8731.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8731.3	SDO	Unsigned32	0	rw
1A02		Transmit PDO 3 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	2	ro

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
	1	PDO mapping for the first application object to be mapped	p8732.0	SDO	Unsigned32	6041 hex	rw
	2	PDO mapping for the second application object to be mapped	p8732.1	SDO	Unsigned32	6077 hex	rw
	3	PDO mapping for the third application object to be mapped	p8732.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8732.3	SDO	Unsigned32	0	rw
1A03		Transmit PDO 4 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	2	ro
	1	PDO mapping for the first application object to be mapped	p8733.0	SDO	Unsigned32	6041 hex	rw
	2	PDO mapping for the second application object to be mapped	p8733.1	SDO	Unsigned32	6063 hex	rw
	3	PDO mapping for the third application object to be mapped	p8733.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8733.3	SDO	Unsigned32	0	rw
1A04		Transmit PDO 5 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	0	ro
	1	PDO mapping for the first application object to be mapped	p8742.0	SDO	Unsigned32	0	rw

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
	2	PDO mapping for the second application object to be mapped	p8742.1	SDO	Unsigned32	0	rw
	3	PDO mapping for the third application object to be mapped	p8742.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8742.3	SDO	Unsigned32	0	rw
1A05		Transmit PDO 6 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	0	ro
	1	PDO mapping for the first application object to be mapped	p8752.0	SDO	Unsigned32	0	rw
	2	PDO mapping for the second application object to be mapped	p8752.1	SDO	Unsigned32	0	rw
	3	PDO mapping for the third application object to be mapped	p8752.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8752.3	SDO	Unsigned32	0	rw
1A06		Transmit PDO 7 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	0	ro
	1	PDO mapping for the first application object to be mapped	p8752.0	SDO	Unsigned32	0	rw
	2	PDO mapping for the second application object to be mapped	p8752.1	SDO	Unsigned32	0	rw

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transmission	Data type	Predefined connection set	Can be read/ written to
	3	PDO mapping for the third application object to be mapped	p8752.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8752.3	SDO	Unsigned32	0	rw
1A07		Transmit PDO 8 mapping parameter	–	–	–	–	–
	0	Number of mapped application objects in PDO	–	SDO	Unsigned8	0	ro
	1	PDO mapping for the first application object to be mapped	p8752.0	SDO	Unsigned32	0	rw
	2	PDO mapping for the second application object to be mapped	p8752.1	SDO	Unsigned32	0	rw
	3	PDO mapping for the third application object to be mapped	p8752.2	SDO	Unsigned32	0	rw
	4	PDO mapping for the fourth application object to be mapped	p8752.3	SDO	Unsigned32	0	rw

Note

Each additional drive begins with the description in an offset of 40 hex.

7.2.3 Other communication objects

Overview

The following table lists the object directory with the index of all other communication objects dependent on the drive:

Table 7- 4 Other communication objects

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transfer	Data type	Pre-set values:	Can be read/ written to
1202		Drive object server SDO parameter	–	–	–	–	–
	0	Number of entries	–	SDO	–	–	ro
	1	COB ID client -> server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB ID server -> client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1203		Drive object server SDO parameter	–	–	–	–	–
	0	Number of entries	–	SDO	–	–	ro
	1	COB ID client -> server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB ID server -> client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1204		Drive object server SDO parameter	–	–	–	–	–
	0	Number of entries	–	SDO	–	–	ro
	1	COB ID client -> server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB ID server -> client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1205		Drive object server SDO parameter	–	–	–	–	–
	0	Number of entries	–	SDO	–	–	ro
	1	COB ID client -> server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB ID server -> client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1206		Drive object server SDO parameter	–	–	–	–	–
	0	Number of entries	–	SDO	–	–	ro
	1	COB ID client -> server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB ID server -> client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1207		Drive object server SDO parameter	–	–	–	–	–

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transfer	Data type	Pre-set values:	Can be read/ written to
	0	Number of entries	–	SDO	–	–	ro
	1	COB ID client -> server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB ID server -> client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1208		Drive object server SDO parameter	–	–	–	–	–
	0	Number of entries	–	SDO	–	–	ro
	1	COB ID client -> server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB ID server -> client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw
1209		Drive object server SDO parameter	–	–	–	–	–
	0	Number of entries	–	SDO	–	–	ro
	1	COB ID client -> server (rx)	p8612.0	SDO	Unsigned32	0x80000000	rw
	2	COB ID server -> client (tx)	p8612.1	SDO	Unsigned32	0x80000000	rw

7.3 Free objects

OD index (hex)	Description	Data type per PZD	Pre-set values:	Can be read/written to	SINAMICS parameters
5800 to 580F	16 freely-interconnectable receive process data	Integer16	0	rw	r8745[x]
5810 to 581F	16 freely-interconnectable transmit process data	Integer16	0	ro	r8746[x]
5820 to 5827	8 freely-interconnectable receive process data	Integer32	0	rw	r8747[x]
5828 to 582F	Reserved	–	–	–	–
5830 to 5837	8 freely-interconnectable transmit process data	Integer32	0	ro	r8748[x]
5838 to 5879	Reserved	–	–	–	–

Note

The free objects for additional drive objects are formed by adding the offset 80 hex to the object number of the freely interconnectable objects.

For example, drive object 2 begins at 5880 hex.

You can interconnect any process data objects using receive/transmit words/double words of the receive and transmit buffer.

Scaling the process data of the free objects:

- 16-bit (word): 4000 hex corresponds to 100%
- 32 bit (word): 4000000 hex corresponds to 100%

If the process data is a temperature value, the scaling of the free objects appears as follows:

- 16-bit (word): 4000 hex corresponds to 100 °C
- 32 bit (word): 4000000 hex corresponds to 100 °C

7.4 CANopen objects of the drive profile DSP402

Overview

The following table lists the object directory with the index of the individual objects for the drives.

CANopen for SINAMICS S120 supports the "Profile Velocity Mode", the "Profile Torque Mode" and the "Velocity Mode".

Table 7- 5 Objects in drive profile DSP402

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transfer	Data type	Default values	Can be read/ written to
Device control							
6007	–	Abort connection option code	p8641	SDO	Integer16	3	rw
6040	–	Controlword	r8795	PDO/SDO	Unsigned16	–	rw
6041	–	Status word	r8784	PDO/SDO	Unsigned16	–	ro
605D	–	Stop option code	p8791	SDO	Integer16	–	rw
6060	–	Modes of operation	p1300	PDO/SDO	Integer8	–	rw
6061	–	Modes of operation display	r8762	PDO/SDO	Integer8	–	ro
6502	–	Supported drive modes	–	SDO	Unsigned32	–	ro
6504	–	Drive manufacturer	–	SDO	String	–	ro
67FF	–	Single device type	–	SDO	Unsigned32	–	ro
Factor group							
6094	00	Velocity encoder factor	–	SDO	Unsigned8	2	ro
	01	Velocity encoder factor numerator	p8798[0]	SDO	Unsigned32	1	rw
	02	Velocity encoder factor denumerator	p8798[1]	SDO	Unsigned32	1	rw
Profile velocity mode							
6063	–	Actual position value	r0482	SDO/PDO	Integer32	–	ro
6069	–	Velocity sensor actual value	r0061	SDO/PDO	Integer32	–	ro
606B	–	Velocity demand value	r1170	SDO/PDO	Integer32	–	ro
606C	–	Velocity actual value Actual velocity	r0063	SDO/PDO	Integer32	–	ro
6083	–	Profile acceleration	p1082/ p1120	SDO	Unsigned32	–	rw
6084	–	Profile deceleration	p1082/ p1121	SDO	Unsigned32	–	rw
6085	–	Quick stop deceleration	p1082/ p1135	SDO	Unsigned32	–	rw
6086	–	Motion profile type	p1115/ p1134	SDO	Integer16	0	rw

OD index (hex)	Subindex (hex)	Object name	SINAMICS parameters	Transfer	Data type	Default values	Can be read/ written to
60FF	–	Target velocity Setpoint velocity	Without ramp-function generator -> p1155[0] With ramp-function generator -> p1070	SDO/PDO	Integer32	–	rw
Profile torque mode							
6071	–	Target torque Setpoint torque	r8797	SDO/PDO	Integer16	–	rw
6072	–	Max torque	p1520	SDO	0	0	0
6074	–	Torque demand value Setpoint torque total	r0079	SDO/PDO	Integer16	–	ro
6077	–	Torque actual value	r0080	SDO/PDO	Integer16	–	ro
Velocity Mode							
6042	–	vl target velocity	r8792	SDO/PDO	Integer16	–	rw
6043	–	vl velocity demand	r1170	SDO/PDO	Integer16	–	ro
6044	–	vl velocity actual value	r0063	SDO/PDO	Integer16	–	ro
6046	0	vl velocity min./max. amount	–	SDO	Unsigned8	–	ro
	1	vl velocity min. amount	p1080	SDO	Unsigned32	–	rw
	2	vl velocity max. amount	p1082	SDO	Unsigned32	–	rw
6048	0	vl velocity acceleration	–	SDO	Unsigned8	–	ro
	1	Delta speed	p1082	SDO	Unsigned32	–	rw
	2	Delta time	p1120	SDO	Unsigned16	–	rw

Note

The drive objects for further SINAMICS drive objects in the drive profile are described with an offset of 800 hex.

SDO server for each drive object

8

8.1 Mapping models

Several mapping models are supported:

- Multi-DO SINAMICS drive unit with one SDO server
- Multi-DO SINAMICS drive unit with several EDS files and SDO servers

Multi-DO SINAMICS drive unit with one SDO server

With this mapping model, the multi-DO SINAMICS drive unit is mapped via modules with an SDO server (default SDO server). A modular EDS file, without manufacturer-specific CANopen objects, exists for every SINAMICS drive unit.

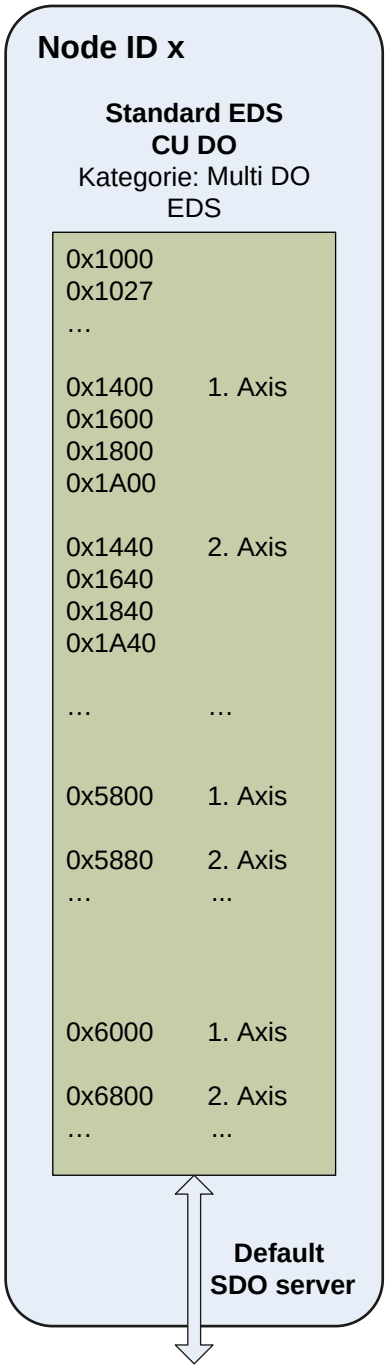


Figure 8-1 Modeling a multi-DO SINAMICS drive unit

Multi-DO SINAMICS drive unit with several EDS files and SDO servers

For this mapping model, the multi-DO SINAMICS drive unit is mapped from n+2 EDS files and n+2 SDO servers.

Components	Explanation
n+2 EDS files	1 standard EDS file for CU drive object 1 manufacturer-specific EDS file for the drive object of the Control Unit n manufacturer-specific EDS files for n axes, which support CANopen (with n = 1 … 8 axes)
n+2 SDO channels	1 default SDO server for CU-DO 1 manufacturer-specific SDO server for the drive object of the Control Unit n manufacturer-specific SDO servers for n axes (for each drive object, which CANopen supports, with n = 1 … 8 axes).

Each additional SDO server is responsible for a drive object. In particular, it is responsible for the manufacturer-specific objects of this drive object. The default SDO server has access to the standard EDS file.

This means that for each CANopen drive object (including the drive object of the Control Unit), there is an additional dedicated manufacturer-specific EDS file.

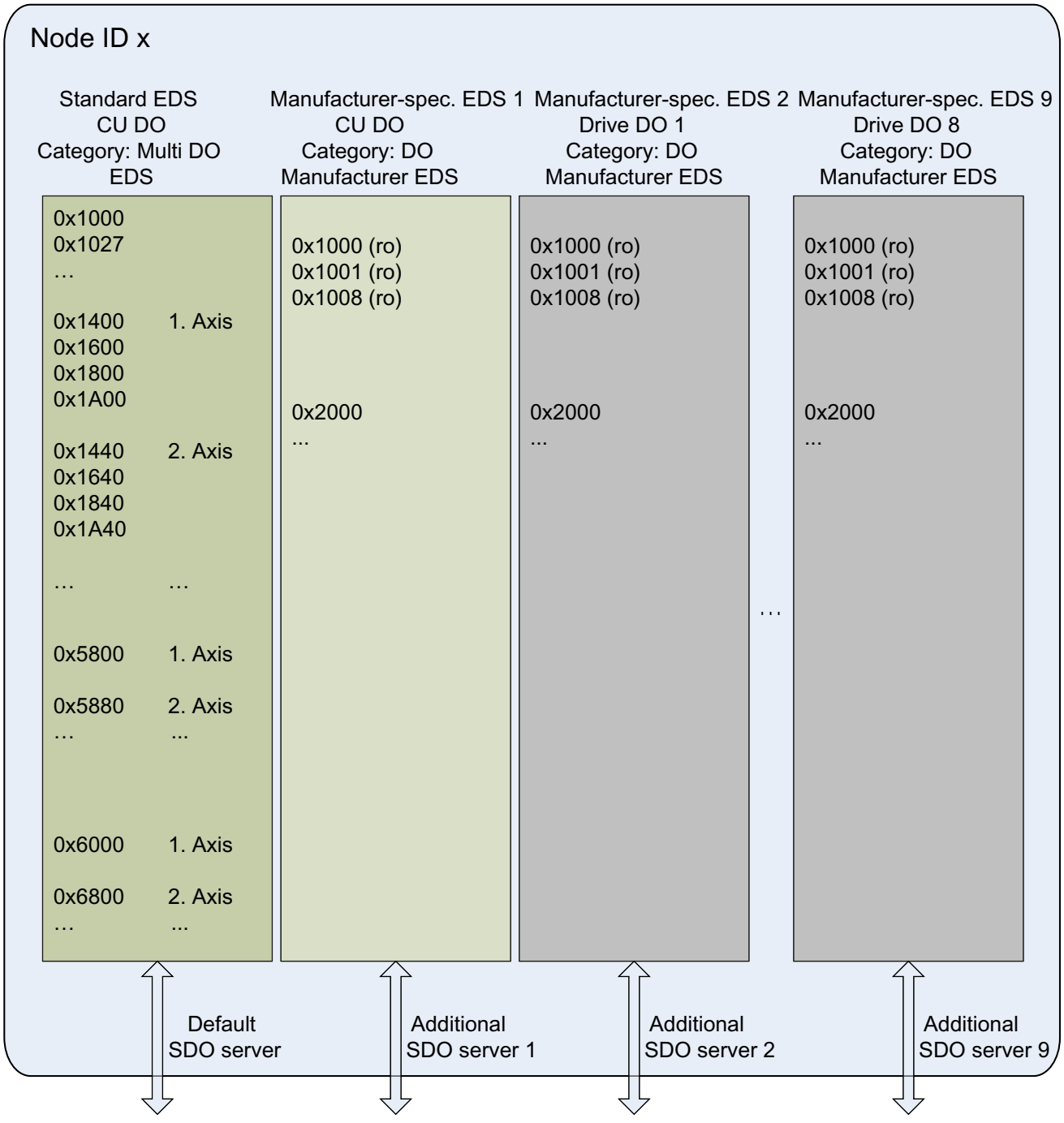


Figure 8-2 Modeling a multi-DO SINAMICS drive unit

8.2 EDS files

In addition to the standard EDS files, there are also standardized CANopen objects (SDO server parameter), which were extended for the additional SDO server.

In addition, you can use the following EDS:

- modular EDS file without manufacturer-specific objects
- EDS files of both categories (modular and manufacturer-specific) for the new modeling

There is then an EDS file for each SINAMICS drive unit and one for each SINAMICS drive object, which supports CANopen.

EDS files of all SINAMICS devices

Name	G120P	G120C	G120S_SERVO	G120S_VECTOR	S110	S120	S150	G130	G150	GL150	GM150	SL150	SM120	MV	Kategorie
SINAMICS_G120P.eds	x														Single DO EDS
SINAMICS_G120C.eds		x													Single DO EDS
SINAMICS_G120S_SERVO.eds			x												Single DO EDS
SINAMICS_G120S_VECTOR.eds				x											Single DO EDS
SINAMICS_MV.eds														x	Multi DO EDS
SINAMICS_S120.eds						x									Multi DO EDS
SINAMICS_S110.eds					x										Multi DO EDS
SINAMICS_S150.eds							x								Multi DO EDS
SINAMICS_SL150.eds												x			Multi DO EDS
SINAMICS_SM120.eds													x		Multi DO EDS
SINAMICS_G130.eds								x							Multi DO EDS
SINAMICS_G150.eds									x						Multi DO EDS
SINAMICS_GL150.eds										x					Multi DO EDS
SINAMICS_GM150.eds											x				Multi DO EDS
SINAMICS_S120_CU.eds						x									DO Manufacturer EDS
SINAMICS_S150_CU.eds							x								DO Manufacturer EDS
SINAMICS_G130_CU.eds								x							DO Manufacturer EDS
SINAMICS_G150_CU.eds									x						DO Manufacturer EDS
SINAMICS_MV_CU.eds														x	DO Manufacturer EDS
SINAMICS_SERVO.eds						x									DO Manufacturer EDS
SINAMICS_VECTOR.eds						x	x	x	x						DO Manufacturer EDS
SINAMICS_VECTORMV.eds											x	x		x	DO Manufacturer EDS
SINAMICS_A_INFM2C.eds														x	DO Manufacturer EDS

Figure 8-3 EDS files of the SIMATICS devices

Multi DO EDS

Device-specific EDS without manufacturer-specific objects.
File name: "SINAMICS_Gerätename.eds"

CANopen objects	Name
Objects from the communication profile - number range 0x1000 - 0x1FFF:	
0x1000	Device Type
0x1001	Error Register
0x1003	Pre-defined error field
0x1005	COB-ID SYNC message
0x1008	Manufacturer device name
0x100A	Manufacturer software version
0x100C	Guard time
0x100D	Life time factor
0x1010	Store parameters
0x1011	Restore default parameters
0x1014	COB-ID EMCY
0x1017	Producer heartbeat time
0x1018	Identity object
0x1027	Module list
0x1029	Error behavior object
0x1200	SDO server parameter Default SDO channel
0x1201	SDO server parameter CU DO channel
0x1200 + (n+1)	with n = 1 … 8 SDO server parameter Drive DO channels
0x1400 + (n-1) * 0x40 bis 0x1407 + (n-1) * 0x40	with n = 1 … 8 axes RPDO communication parameter
0x1600 + (n-1) * 0x40 bis 0x1607 + (n-1) * 0x40	with n = 1 … 8 axes RPDO mapping parameter
0x1800 + (n-1) * 0x40 bis 0x1807 + (n-1) * 0x40	with n = 1 … 8 axes TPDO communication parameter
0x1A00 + (n-1) * 0x40 bis 0x1A07 + (n-1) * 0x40	with n = 1 … 8 axes TPDO mapping parameter
Free process data objects in the manufacturer-specific number range of all n logical devices (n axes)	
0x5800 + (n-1) * 0x80 bis 0x580F + (n-1) * 0x80	with n = 1 … 8 axes 16 Bit fee receive PDOs
0x5810 + (n-1) * 0x80 bis 0x581F + (n-1) * 0x80	with n = 1 … 8 axes 16 Bit free transmit PDOs
0x5820 + (n-1) * 0x80 bis 0x5827 + (n-1) * 0x80	with n = 1 … 8 axes 32 Bit free receive PDOs
0x5830 + (n-1) * 0x80 bis 0x5837 + (n-1) * 0x80	with n = 1 … 8 axes 32 Bit free transmit PDOs
Objects of the standard profile - number range of all n logical devices (n axes)	
0x6000 + (n-1) * 0x800 bis 0x67FF + (n-1) * 0x800	with n = 1 … 8 axes

DO manufacturer EDS

Drive object-specific EDS with manufacturer-specific objects.
File name for CU DO: "SINAMICS_Gerätename_CU.eds"

File name for DO: "SINAMICS_DOClassID.eds"

CANopen objects	Name
Specified objects (mandatory) from the communication profile - number range 0x1000 to 0x1FFF:	
0x1000 (read only)	Device Type
0x1001 (read only)	Error Register
0x1018 (read only)	Identity Object
Manufacturer-specific objects in the number range 0x2000 to 0x57FF: Conversion rule: Manufacturer-specific object index = SINAMICS parameter No. in hex + 0x2000	

8.3 SDO server

Default SDO server

The default SDO server is used to access the standardized objects, i.e. on the EDS file of category "Multi DO EDS".

By changing the default value of the p8630[0] >= 1, manufacturer-specific objects can still be accessed. This is especially important for single DO drives, which only have the default SDO server.

Server default SDO parameters:

0x1200.0 = 2
0x1200.1 = 0x600 + Node ID
0x1200.2 = 0x580 + Node ID

The server SDO parameters of the default SDO server are read-only, and mapped in SINAMICS parameter r8610.

Additional SDO server

The next SDO server after the default SDO server is used to access the manufacturer-specific objects of the CU drive object; i.e. to access the EDS file of the category "DO Manufacturer EDS".

Server SDO parameters of the drive object of the control unit:

0x1201.0 = 2
0x1201.1 = 0x601 + Node ID
0x1201.2 = 0x581 + Node ID

The object in the fixed area is in the EDS file.

The n additional SDO servers are used to access the manufacturer-specific objects of the CAN drive object with n = 1 … 8 axes, i.e. to access the EDS files of the category "DO Manufacturer EDS".

There is no SDO access to drive object categories that are not supported by CAN.

Server SDO parameters of the nth axis:

(0x1201 + n).0 = 2 with n = 1 … 8 axes
(0x1201 + n).1 = 0x601 + Node ID + n with n = 1 … 8 axes
(0x1201 + n).2 = 0x581 + Node ID + n with n = 1 … 8 axes

The objects in the modular area are located in the EDS file.

The additional SDO servers are used to improve the addressing, and not to facilitate parallel SDO access. Response times should be expected when accessing the parameter manager.

In the factory setting, all additional SDO servers are set to invalid using bit 31, and when required, must be manually activated.

The server SDO parameters of the additional SDO servers can be written to, and are mapped in SINAMICS parameter p8612.

The CAN identifiers of the additional SDO servers can be freely selected within the CANopen SDO number range 0x601 - 0x67F for receive SDO or 0x581 - 0x5FF for send SDO.

8.4 Objects of the manufacturer-specific EDS files

There are manufacture-specific CANopen objects for a series of SINAMICS parameters in the corresponding manufacture-specific EDS files:

Parameter name	Parameter reference	CANopen object
Current limit	p0640	0x2280
Technology controller enable	p2200	0x2898
Technology controller actual value filter time constant	p2265	0x28D9
Technology controller differentiation time constant	p2274	0x28E2
Technology controller proportional gain	p2280	0x28E8
Technology controller maximum limiting	p2291	0x28F3
Technology controller minimum limiting	p2292	0x28F4
Output frequency	r0066	0x2042
Speed setpoint smoothed	r0020	0x2014
Output frequency smoothed	r0024	0x2018
Output voltage smoothed	r0025	0x2019
DC link voltage, smoothed	r0026	0x201A
Absolute actual current, smoothed	r0027	0x201B
Actual torque, smoothed	r0031	0x201F
Actual active power, smoothed	r0032	0x2020
Motor temperature	r0035	0x2023
Power unit temperatures	r0037	0x2025
Energy display	r0039	0x2027
Command Data Set CDS effective	r0050	0x2032
Status word 2 (ZSW2) effective	r0053	0x2035
Control word 1 (STW1) effective	r0054	0x2036
CU digital inputs, status	r0722	0x22D2
CU, digital outputs status	r0747	0x22EB
CU analog inputs input voltage/current	r0752	0x22F0
CU analog outputs output voltage/current	r0774	0x2306
Fault number	r0947	0x23B3
Actual pulse frequency	r1801	0x2709
Alarm number	r2110	0x283E
Technology controller setpoint after ramp-function generator	r2260	0x28D4
Technology controller actual value after filter	r2266	0x28DA
Technology controller output signal	r2294	0x28F6
Technology controller ramp-up time	p2257	0x28D1
Technology controller ramp-down time	p2258	0x28D2
Technology controller integral action time	p2285	0x28ED

Manufacturer-specific CANopen object

Appendix A

Note

All the parameters, faults, alarms, and function diagrams for CANopen in the SINAMICS drive line-up are described in the SINAMICS S120/S150 List Manual.

A.1 Glossary

When using a CANopen profile via the CAN bus, you will encounter the following common terms and abbreviations:

CAL (CAN Application Layer)

Communication layer above the CAN bus designed for CAN bus applications in open communication systems. It comprises NMT, DBT, LMT, and CMS elements. Since CAL is very extensive and highly flexible, a subset of CAL functions for automation applications has been defined with the CANopen communication profile CiA DS 301.

CAN (controller area network)

A serial bus system (also known as CAN bus) that was originally designed for use in vehicles but is now also used in automation technology. CANopen (see below) extends the CAN bus protocols to include additional layers.

CAN controller

An electronic module whose hardware processes the CAN bus protocols.

CAN identifier

With the assignment of CAN identifiers to CAN messages (CANopen: PDO, SDO), the relative priority of the CAN messages to one another is specified.

CANopen

A CiA-defined communication model based on the CAN bus and CAL. To make it easier to use devices produced by different manufacturers on a bus, a subset of CAL functions for automation applications has been defined with the CANopen communication profile CiA DS 301. Other profiles are also defined for certain device types (e.g. drives).

CiA (CAN in Automation international users and manufacturers group)

Association of manufacturers and users of devices with a CAN interface.

CMS (CAN message specification)

A part of the CAL that defines different mechanisms for transferring data.

COB (communication object)

On the CAN bus, data is transferred in packages known as communication objects (COB) or CAN message).

Devices connected to the CAN bus can transmit and receive COB.

COB-ID (COB identifier)

Each COB can be uniquely identified by means of an identifier, which is part of the COB. CAN specification 2.0A supports up to 2048 COB, which are identified by means of 11-bit identifiers. In this Commissioning Manual, COB IDs are always specified as hexadecimal values.

A list of COB identifiers, which contains all the COB that can be accessed via CAN, is available in the object directory for the relevant drive unit.

DRIVECOM

Association of drive manufacturers that has developed standards for networking drives (profiles). DRIVECOM profile 22 for positioning drives, which is implemented in the servo amplifier, was used by CiA as a basis for developing CANopen drive profile CiA DSP 402.

EDS (Electronic Data Sheet)

Electronic data sheet

EMCY (Emergency)

SINAMICS features an emergency object to inform other nodes on the CANopen bus of internal device faults or CAN bus faults. It is assigned a high priority and provides important information about the status of the drive unit.

NMT (network management)

A part of CAL used for initialization, configuration, and troubleshooting purposes.

Node ID (node identification)

Uniquely identifies a device in the CANopen network. For this reason, all the devices must have a unique node ID (bus address). The default distribution (standard setting) of the COB IDs is derived from the node ID. In this Commissioning Manual, node IDs are always specified as hexadecimal values.

OD (object directory)

A "database" – or object directory – containing all the objects supported by a drive is defined for each drive unit. The object directory contains:

- Type, description, and serial number of the device
- Name, format, description + index for each object
- Lists of PDO, SDO
- Which data is assigned to the PDO?
- When are PDO transmitted? (SYNC, change in object, etc.)

- The time at which emergency messages are transmitted
- ...

All the drive unit variables are accessed via objects. The SDO and PDO communication services access the object directory of the drive unit.

PDO (process data object)

Used for accessing selected data rapidly and in real time. For certain variables, mappings to certain PDOs are preconfigured.

The SDO is used to access all the other variables.

Profile

In the case of communication with bus systems, profiles are documents used for device standardization purposes, whereby communication functions (in a communication profile), device functions (in a device profile), or drive functions (in a drive profile) are described from the point of view of the communication interface.

RPDO (receive PDO)

PDO is received by the device (contains the final position, for example).

SDO (service data object)

The SDO provides access to all variables in a CANopen device (in the case of drives: drive and CANopen variables).

The SDO is generally used for configuration purposes. PDO provide fast, real-time access to selected variables.

SYNC (synchronization)

SYNC is a special message frame that synchronizes the CAN devices with each other. This message frame has a very high priority.

TPDO (transmit PDO)

PDO transmitted by the drive (contains the actual position value, for example).

Variable

All the drive and CANopen functions can be accessed via variables.

Variables can be accessed via SDO or PDO.

Index

D

E

F

G

H

I

L

M

N

O

P

R

S

T

V

Siemens AG
Industry Sector
Drive Technologies
Motion Control Systems
P.O. Box 3180
91050 ERLANGEN
GERMANY

www.siemens.com/motioncontrol