SINAMICS S120

Commissioning Manual · 10/2008
CANopen interface

SINAMICS

SIEMENS

SIEMENS

SINAMICS

S120
SINAMICS S120 CANopen interface

Commissioning Manual

Valid for:
Firmware Version 2.6 SP1

(IH2), V2.6, 10/2008 Edition
6SL3097-2AA00-0BP3

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
with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION
without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE
indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

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 adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the 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

Ordernumber: 6SL3097-2AA00-0BP3
℗ 01/2009

Preface

SINAMICS documentation

The SINAMICS documentation is organized in 2 parts:

- General documentation / catalogs
- Manufacturer/Service Documentation

An overview of publications, which is updated on a monthly basis and also provides information about the language versions available, can be found on the Internet at:

http://www.siemens.com/motioncontrol
Select "Support" > "Technical Documentation" > "Overview of Publications".

The Internet version of DOConCD (DOConWEB) is available under:

http://www.automation.siemens.com/doconweb

Information about training courses and FAQs (Frequently Asked Questions) can be found at the following website:

http://www.siemens.com/motioncontrol under menu option "Support"

Utilization phases

Table 1 Usage phases and the available documents/tools

Usage phase	Document / tool
Exploratory	SINAMICS S Sales Documentation
Planning/configuration	• SIZER configuration tool • Configuration Manuals, Motors
Decision/ordering	SINAMICS S Catalogs
Installation/assembly	• SINAMICS S110 Equipment Manual • SINAMICS S120 Equipment Manual for Control Units and Supplementary System Components • SINAMICS S120 Equipment Manual Power Modules Booksize • SINAMICS S120 Equipment Manual for Booksize Compact Power Units • SINAMICS S120 Equipment Manual Power Modules Chassis • SINAMICS S120 Equipment Manual for AC Drives

Usage phase	Document / tool
Commissioning	• STARTER parameterization and commissioning tool • SINAMICS S110 Function Manual • SINAMICS S110 List Manual • SINAMICS S120 Getting Started • SINAMICS S120 Commissioning Manual • SINAMICS S120 CANopen Commissioning Manual • SINAMICS S120 Function Manual • SINAMICS S120/S150 List Manual
Usage/operation	• SINAMICS S110 Function Manual • SINAMICS S110 List Manual • SINAMICS S120 Commissioning Manual • SINAMICS S120/S150 List Manual
Maintenance/servicing	• SINAMICS S110 Function Manual • SINAMICS S110 List Manual • SINAMICS S120 Commissioning Manual • SINAMICS S120/S150 List Manual
References	• SINAMICS S110 List Manual • SINAMICS S120/S150 List Manual

Target group

This documentation is aimed at machine and plant builders, commissioning engineers, and service personnel who use SINAMICS.

Benefits

Note

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

This Commissioning Manual extends the description of "Initial commissioning using servo as an example" to include a description of the initial commissioning procedure for the CANopen communication interface:

- **SINAMICS S120 with the CBC10 communication module**
- **SINAMICS S110 with integrated CANopen interface**

Detailed instructions on commissioning the entire SINAMICS drive line-up are available in:

- SINAMICS S120 Commissioning Manual /IH1/.
- SINAMICS S110 Function Manual /FH3/.

Standard scope

This documentation only describes the functionality of the standard version. Extensions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

Other functions not described in this documentation might be able to be executed in the drive system. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

For the sake of simplicity, this documentation does not contain all detailed information about all types of the product and cannot cover every conceivable case of installation, operation, or maintenance.

Technical Support

If you have any questions, please get in touch with our Hotline:

	Europe / Africa
Phone	+49 180 5050 - 222
Fax	+49 180 5050 - 223
Internet	http://www.siemens.de/automation/support-request

	Americas
Phone	+1 423 262 2522
Fax	+1 423 262 2200
E-mail	mailto:techsupport.sea@siemens.com

	Asia/Pacific
Phone	+86 1064 719 990
Fax	+86 1064 747 474
E-mail	mailto:adsupport.asia@siemens.com

Note

Country telephone numbers for technical support are provided under the following Internet address:

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

Calls are subject to charge (e.g. € 0.14/min from fixed lines within Germany).

Tariffs of other telephone service providers may vary.

Questions about the Manual

If you have any questions (suggestions, corrections) regarding this documentation, please fax
or e-mail us at:

Fax	+49 9131 98 63315
E-mail	E-mail to: docu.motioncontrol@siemens.com

A fax form is available in the appendix of this document.

Internet address for SINAMICS

http://www.siemens.com/sinamics

ESD notices

⚠ CAUTION

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

Regulations for handling ESD components:

When handling components, make sure that personnel, workplaces, and packaging are well earthed.

Personnel in ESD areas with conductive flooring may only handle electronic components if:

- They are grounded with an ESD wrist band
- They are wearing ESD shoes or ESD shoe grounding straps

Electronic boards should only be touched if absolutely necessary. They must only be handled on the front panel or, in the case of printed circuit boards, at the edge.

Electronic boards must not come into contact with plastics or items of clothing containing synthetic fibers.

Boards must only be placed on conductive surfaces (work surfaces with ESD surface, conductive ESD foam, ESD packing bag, ESD transport container).

Do not place boards near display units, monitors, or television sets (minimum distance from screen: 10 cm).

Measurements must only be taken on boards when the measuring instrument is grounded (via protective conductors, for example) or the measuring probe is briefly discharged before measurements are taken with an isolated measuring device (for example, touching a bare metal housing).

Safety information/instructions

⚠ DANGER

Commissioning must not start until you have ensured that the machine in which the components described here are to be installed complies with Directive 98/37/EC.

SINAMICS S equipment 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.

⚠ WARNING

Correct and safe operation of SINAMICS S equipment assumes correct transportation, storage, setup, and installation, as well as careful operation and maintenance.

The details in the catalogs and proposals also apply to the design of special equipment versions.

In addition to the danger and warning information provided in the technical customer documentation, the applicable national, regional, and system-specific regulations and requirements must be taken into account.

Only protective extra-low voltages (PELVs) that comply with EN60204-1 must be connected to all connections and terminals between 0 and 48 V.

CAUTION

Operating the equipment in the immediate vicinity (< 1.5 m) of mobile telephones with a transmitter power of > 1 W may lead to incorrect operation.

Table of contents

Add infeed

1

1.1 Previous knowledge

Introduction

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

This Chapter includes, among other things,

- An overview of the most important terms and abbreviations
- A breakdown of the communication objects in the CANopen object directory in the CANopen slave software

You must be familiar with the following standards:

Note

SINAMICS with CANopen complies with the following standards:

- CiA DS-301 V4.02 (Application Layer and Communication Profile)

- CiA DS-402 V2.0 (Device Profile for Drives and Motion Control)

- CiA DR-303-3 V1.2 (Indicator Specification)

- CiA DS-306 V1.3: (Electronic data sheet specification for CANopen)

1.2 Commissioning options

Introduction

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

- Via predefined message frames ("predefined connection set")
- Via free PDO mapping (user-defined message frames)

1.3 Terminology

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: PDOs, SDOs), 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 messages.

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

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 COBs, 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 COBs that can be accessed via CAN, is available in the object directory for the relevant drive unit.

DBT (distributor)

A part of the CAL that controls the distribution of COB IDs. Like most CANopen devices, the digital servo amplifiers use more straightforward means of assigning COB IDs to a device: they are selected via default values on the basis of the node ID and can, if necessary, be changed via SDO.

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.

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.

Channel

With the SINAMICS drive line-up, up to 24 receive PDOs can be received.

One channel in the CAN controller is assigned to each activated receive PDO. Transmit PDOs are transmitted via two predefined channels.

Transmit PDOs always use two predefined channels.

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 PDOs and SDOs

- The data that is assigned to the PDOs
- The time at which the PDOs are 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. Mappings of certain PDOs are preconfigured for certain variables or groups of variables.

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. PDOs 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 SDOs or PDOs.

1.4 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 PROFIBUS.

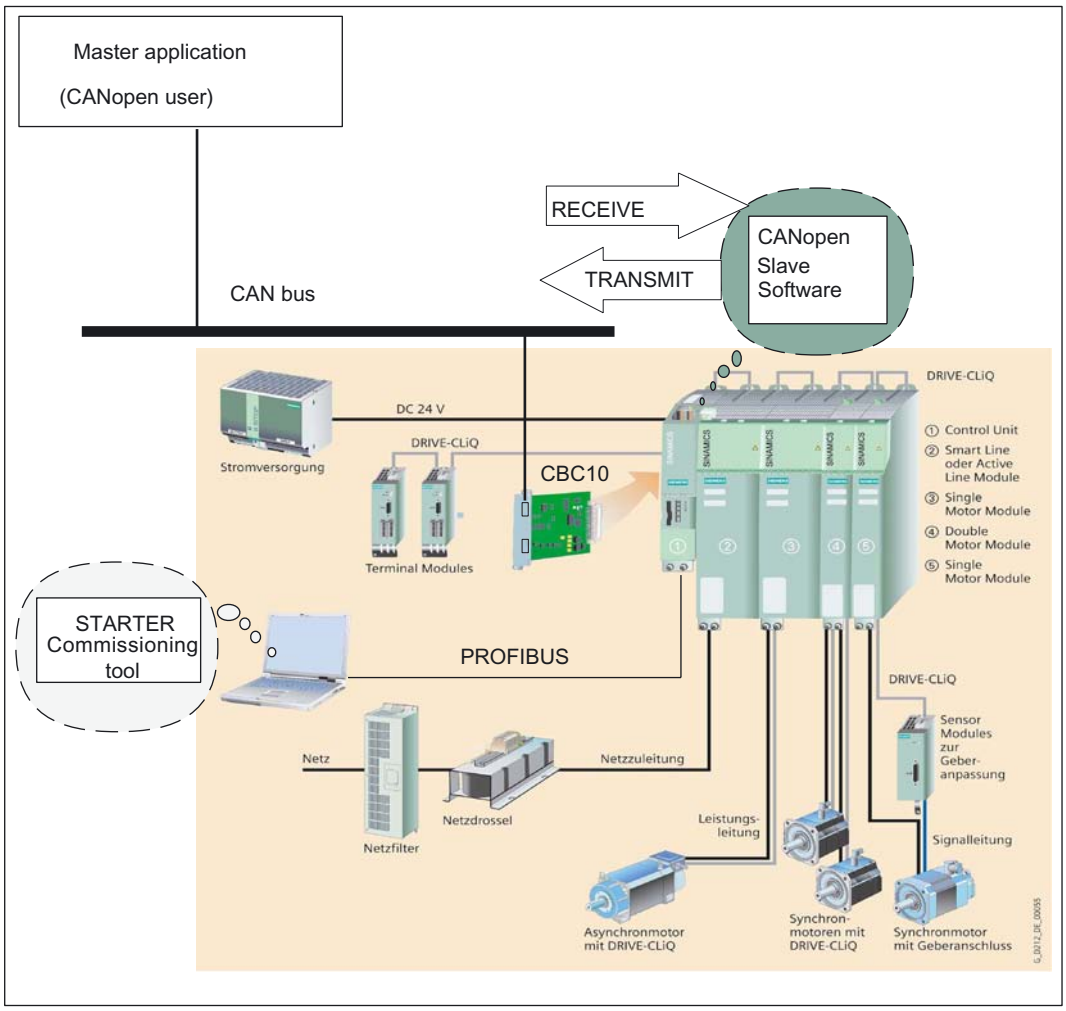


Figure 1-1 SINAMICS S120 drive line-up with CAN bus

Note

For SINAMICS S110:

- When ordering, you must have already decided whether to use a CAN bus or Profibus interface, because the bus interface is integrated in the Control Unit.
- It is not possible to exchange the bus interface afterwards, e.g. with a Profibus interface.
- The simultaneous use of CAN bus und Profibus is not possible.
- SINAMICS S110 does not support all the components displayed in the S120 topology.

1.5 CANopen slave software

Introduction

In the CANopen slave software, a separate expert list is created – among other things – for each module in the SINAMICS drive line-up. The expert list reflects – among other things – the CANopen object directory.

Example

The data value for the control word of the object directory index from the CANopen object directory is entered in parameter r8795 in the expert list.

Depending on how the Motor Modules are arranged in the drive line-up, they are counted as a drive object from the point of view of CANopen.

See the following diagram:

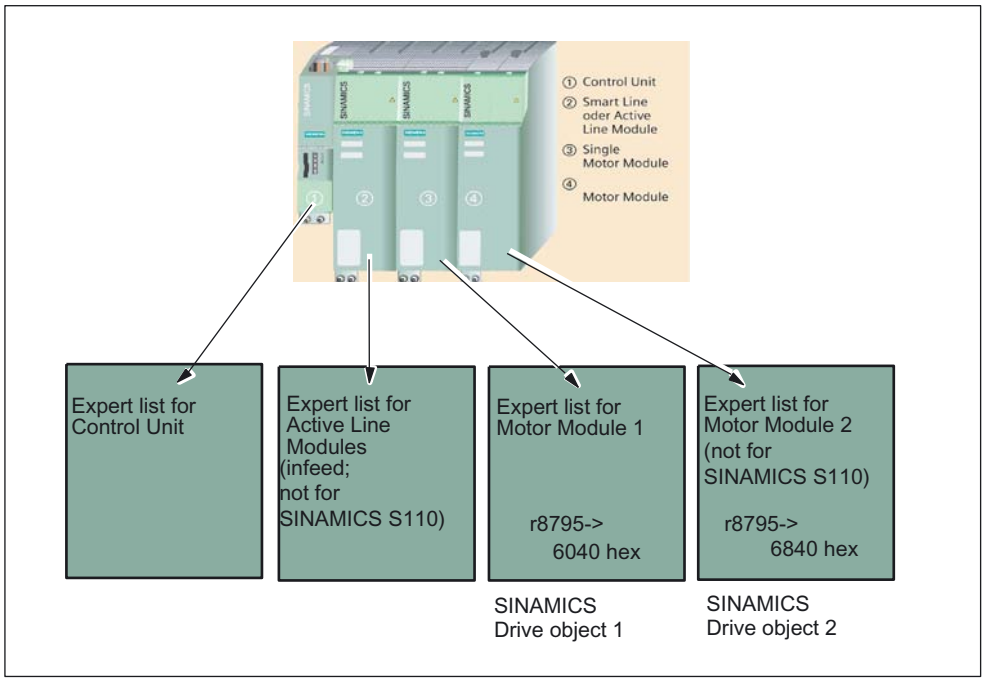


Figure 1-2 Breakdown of drive objects from the point of view of CANopen

Motor Module 1 is the first drive object for CANopen. In the description in the drive profile, drive object 2 begins with an offset of 800 hex.

If, for example, you want to read a value from parameter **r8795** (control word for Motor Module 1, that is, drive object 1), the value from object **6040** hex is displayed.

For Motor Module 2 (drive object 2), this value is **6840** hex.

Note

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

Note

For SINAMICS S110:

- The CAN bus interface is integrated into the Control Unit.
- Not all the components displayed are supported.
- For the S110 there is only one Power Module instead of a separate infeed and Motor Module (as for S120).

1.6 CANopen object directory

Introduction

Note

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). These are as follows:

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

Eight of the drive objects, supported by CANopen, which access the object directory.

Note

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

See also

Communication objects (Page 129)

CANopen object directory

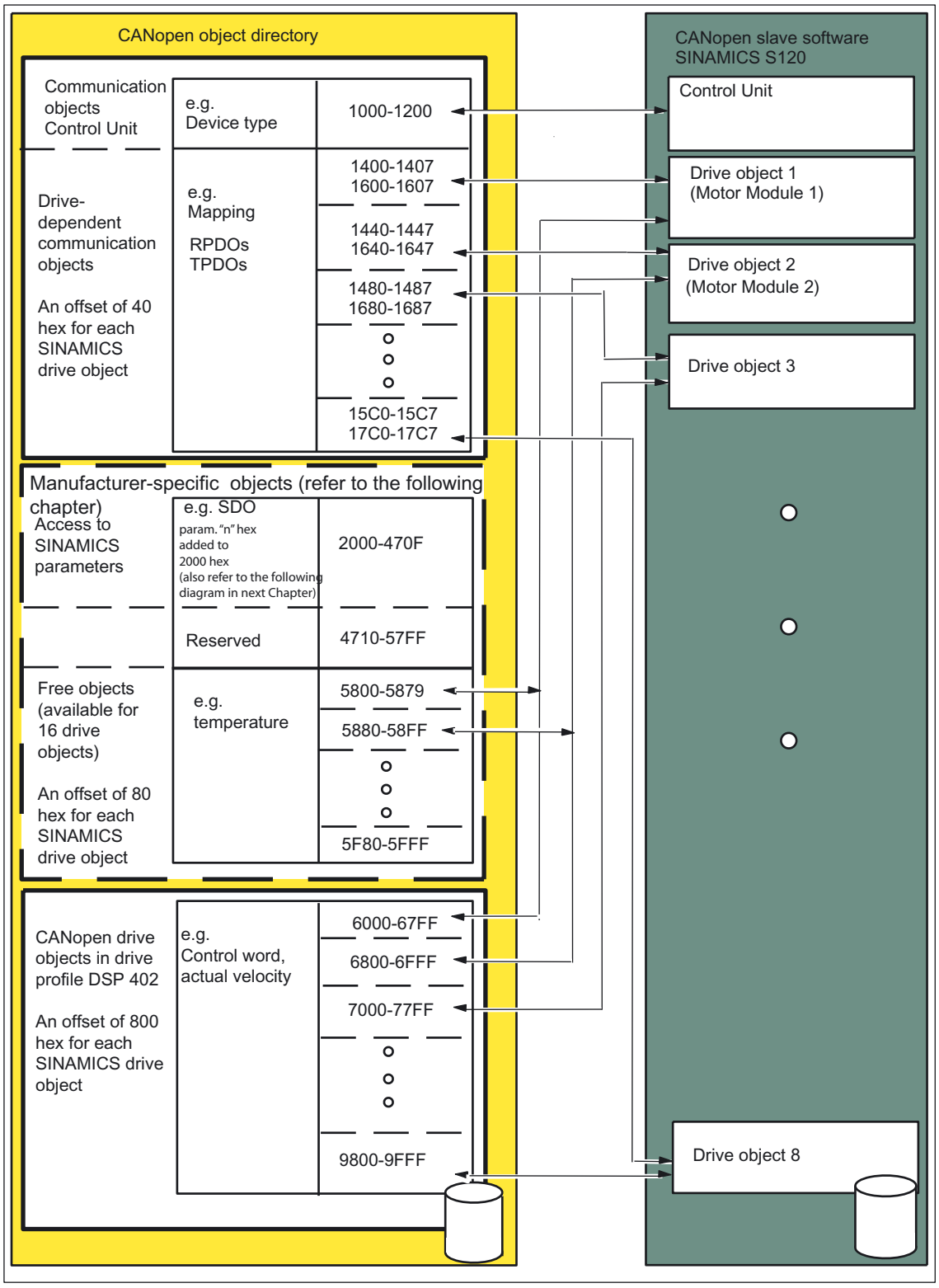


Figure 1-3 CANopen object directory

Note

For SINAMICS S110:

- SINAMICS S110 supports only one Power Module with integrated infeed, DC link and motor power unit.

1.7 Manufacturer-specific objects

Introduction

As shown in the previous chapter, 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.

1.7.1 Objects to access SINAMICS parameters

Introduction

You can access the data values of the SINAMICS parameters 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.

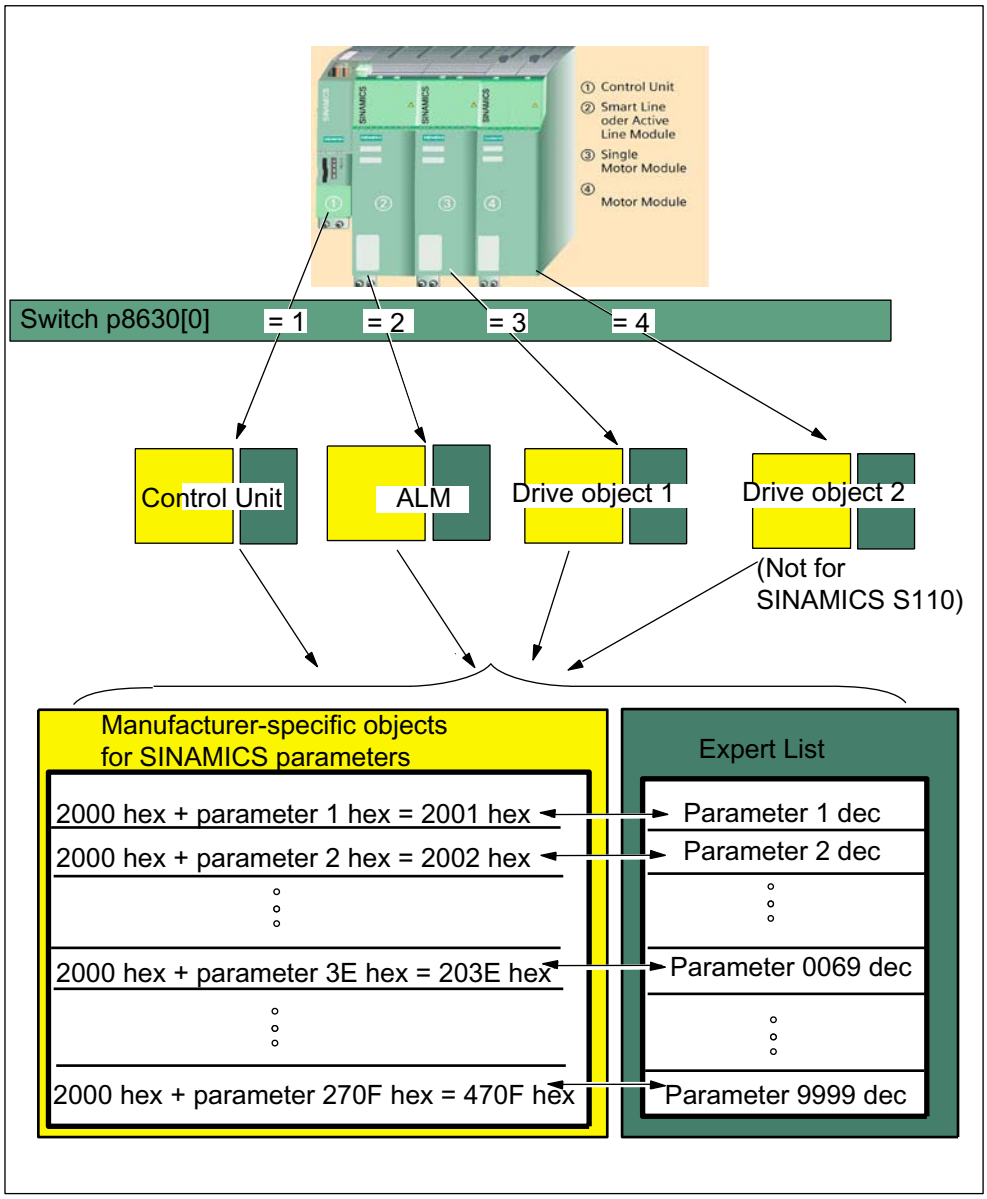


Figure 1-4 Manufacturer-specific objects

Note

For SINAMICS S110:

- SINAMICS S110 only supports one drive object.

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 the SDO->parameter channel.

This functions as follows:

- The SDO->parameter channel operates entirely within the manufacturer-specific area of the CANopen profile.

  This means that all SINAMICS parameters can be addressed via the objects 2000 hex to 470F hex.
- The SDO->parameter channel converts manufacturer-specific objects to parameters internally. 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 already 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".

1.7.2 Free objects

Introduction

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

Free objects

For each drive object (max. 16), 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	Integer16
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.

e.g.: If drive object 2 starts from 5880 hex.

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

Transfer PDO/SDO

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

- PDO
- SDO

Both transfer types require that the free object to be transferred is "mapped".

See also

CANopen object directory (Page 22)

PDO mapping (Page 32)

SDO services (Page 36)

1.8 Number of PDOs that can be set up (channels)

Introduction

PDOs (process data objects) are process data that allow fast, real-time access to selected data. Mappings have been preconfigured on certain PDOs for certain variables or groups of variables.

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

General procedure

Note

CANopen can monitor up to eight SINAMICS drive objects in the object directory. Max. eight PDOs can be specified in one SINAMICS drive object.

Note

The SINAMICS drive line-up can manage a maximum of 25 receive and transmit PDOs (channels) on the CAN bus.

As a maximum of eight PDOs can be specified for each SINAMICS drive object, this means that at maximum utilization of the created PDOs, a utilization of three SINAMICS drive objects would be possible. An additional PDO could also be created.

General procedure

Eight PDOs (plus one PDO in one drive object) multiplied by three SINAMICS drive objects -> max. 25 PDOs (channels).

1.9 Send and receive message frames for process data

Introduction

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

1. Via predefined message frames ("predefined connection set") and COB IDs.
2. Via free PDO mapping (user-defined message frames)

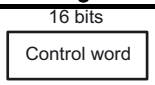
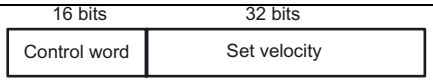
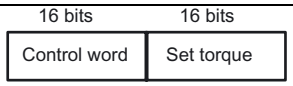
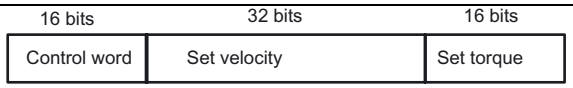
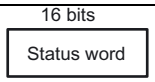
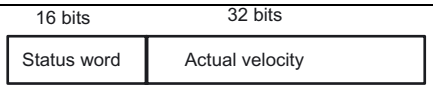
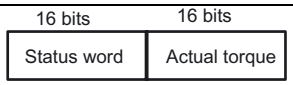
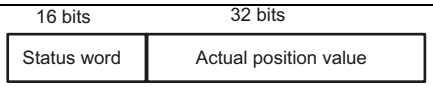
In the first step, you are advised to activate the "predefined connection set" and then update the preconfigured parameters in accordance with your application in the free PDO mapping process.

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+6062 hex
	TPDO3	 6041 hex+6074 hex
	TPDO4	 6041 hex+6083 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: as of parameter **p8710**

For transmit message frames: as of parameter **p8730**

1.10 PDO mapping

Introduction

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 itself transfers the data values for these objects.

Once detected, each drive object is assigned up to 8 receive and 8 transmit PDOs.

A CAN message frame can transfer up to 8 bytes of user data. The user can decide 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)):

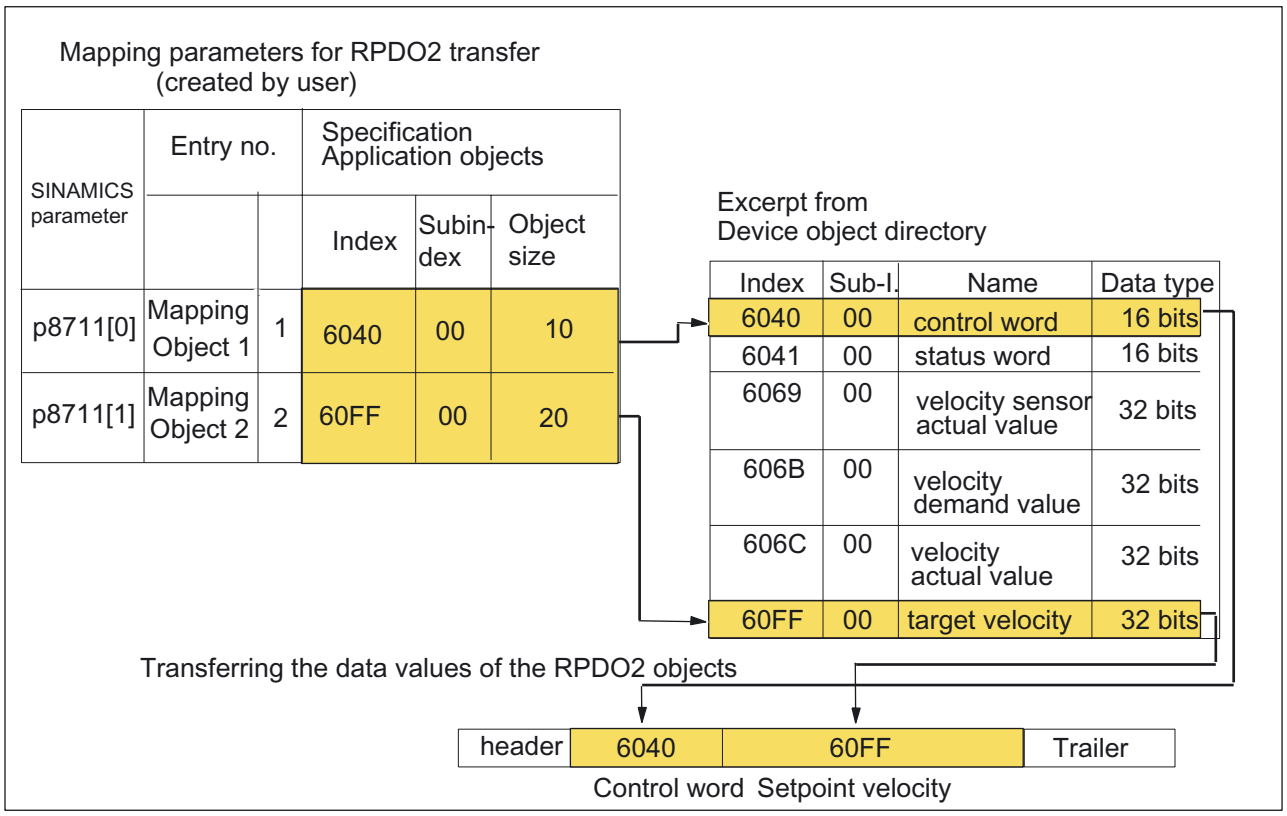


Figure 1-5 PDO mapping

See also

CANopen object directory (Page 22)

Free objects (Page 28)

1.11 COB-ID

Introduction

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 COBs, which are identified by means of 11-bit identifiers.

A list of COB identifiers, which contains all the COBs 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–SDO2)	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

Footnotes for above table

- [1)] The OD index for the Tx and Rx PDOs 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 TPDOs, the OD index for the first drive object begins with 1800 hex and, for RPDOs, 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.

COB IDs for SINAMICS

The following COB IDs for receive and transmit message frames are predefined for SINAMICS drive objects in the "Predefined Connection Set":

Table 1- 4 Identifier assignment

Communication objects	Function code		Resulting COB ID		OD index (hex)
	dec	bin	hex	Explanation	
TPDO1	3	0011	181–1FF	180 hex + node ID	1800
RPDO1	4	0100	201–27F	200 hex + node ID	1400
TPDO2	5	0101	281–2FF	280 hex + node ID	1801
RPDO2	6	0110	301–37F	300 hex + node ID	1401
TPDO3	7	0111	381–3FF	380 hex + node ID	1802
RPDO3	8	1000	401–47F	400 hex + node ID	1402
TPDO4	9	1001	481–4FF	480 hex + node ID	1803
RPDO4	10	1010	501–57F	500 hex + node ID	1403

Note

For the "Predefined Connection Set", the COB IDs are incremented by +1 for each additional SINAMICS drive object.

Example:

For the **first** SINAMICS drive object,
180 hex + node ID applies for TPDO1.

For the **second** SINAMICS drive object,
180 hex + node ID **plus 1** applies for TPDO1.

Note

You assign COB IDs for each drive object in the STARTER expert list as of **p8700** for receive message frames and **p8720** for transmit message frames.

1.12 SDO services

Introduction

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 channel of a drive unit are defined according to CANopen.

For communication between:

- Receive: server <= client, COB ID 600 hex + node ID applies
- Transmit: server => client, COB ID 580 hex + node ID applies

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

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

SDO Protocol Write
This protocol is used to write data in the drive unit.

Confirmation with the signal "Write Response".

Table 1- 5 SDO Protocol Write

Write Request			
CANopen Master Request ----> SINAMICS message			
Byte 0	**Bytes 1..2**	**Byte 3**	**Bytes 4..7**
cs = 2Fhex	index	subindex	data byte 4
cs = 2Bhex	index	subindex	data byte 4-5
cs = 27hex	index	subindex	data byte 4-6
cs = 23hex	index	subindex	data byte 4-7
Write Response			
SINAMICS confirmation ----> CANopen Master Response			
Byte 0	**Bytes 1..2**	**Byte 3**	**Bytes 4..7**
Cs = 60	index	subindex	Reserved

Explanation of the above table:

- **cs:** command specifier, this object always contains 4 bytes. In the cs it is determined how much of the sent byte is 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 Read

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

Request to read via the signal "Read Request".

Table 1- 6 SDO Protocol Read

Read Request			
CANopen Master Request ----> SINAMICS message			
Byte 0	**Bytes 1..2**	**Byte 3**	**Bytes 4..7**
cs = 40	Index	subindex	Reserved
SINAMICS confirmation <---- CANopen Master Response			
Read Response			
Byte 0	**Bytes 1..2**	**Byte 3**	**Bytes 4..7**
cs = 4Fhex	index	subindex	data byte 4
cs = 4Bhex	index	subindex	data byte 4-5
cs = 47hex	index	subindex	data byte 4-6
cs = 43hex	index	subindex	data byte 4-7

Explanation of the above table:

- **cs:** command specifier, this object always contains 4 bytes. In the cs it is determined how much of the sent byte is 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 carry out the SDO service "Cancel Transfer Protocol".

Table 1- 7 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, the size of 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- 8 SDO Abort codes

Abort code	Description
0503 0000h	Toggle bit not alternated
0504 0000h	SDO protocol timed out
0504 0001h	Client/server command specifier not valid or unknown
0504 0002h	Invalid block size (block mode only)
0504 0003h	Invalid sequence number (block mode only)
0504 0004h.	CRC error (block mode only)
0504 0005h	Out of memory
0601 0000h	Unsupported access to an object
0601 0001h	Attempt to read a write-only object
0601 0002h	Attempt to write a read-only object
0602 0000h	Object does not exist in the object dictionary
0604 0041h	Object cannot be mapped to the PDO
0604 0042h	The number and length of the objects to be mapped would exceed PDO length
0604 0043h	General parameter incompatibility reason
0604 0047h	General internal incompatibility in the device
0602 0000h	Object does not exist in the object dictionary
0604 0041h	Object cannot be mapped to the PDO
0604 0042h	The number and length of the objects to be mapped would exceed PDO length
0604 0043h	General parameter incompatibility reason
0604 0047h	General internal incompatibility in the device
0606 0000h	Access failed due to a hardware error
0607 0010h	Data type does not match, length of service parameter does not match.
0607 0012h	Data type does not match, length of service parameter too high.
0607 0013h	Data type does not match, length of service parameter too low.
0609 0011h	Subindex does not exist
0609 0030h	Value range of parameter exceeded (only for write access)
0609 0031h	Value of parameter written too high
0609 0032h	Value of parameter written too low
0609 0036h	Maximum value is less than minimum value
0800 0000h	General error
0800 0020h	Data cannot be transferred to or stored in the application
0800 0021h	Data cannot be transferred to or stored in the application because of local control
0800 0022h	Data cannot be transferred to or stored in the application because of the current device state.
0800 0023h	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).

1.13 PDO services

Introduction

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

The PDOs 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 PDOs 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 data transmission and the receive PDO (RPDO) is responsible for receiving data. CANopen devices that support TPDOs are called PDO producers and CANopen devices that support RPDOs 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.

PDO communications parameter

Table 1- 9 PDO communications parameter 1400h ff, 1800h ff

Subindex	Name	Data type
00h	Highest subindex that is supported	UNSIGNED8
01h	COB ID	UNSIGNED32
02h	Transfer mode	UNSIGNED8
03h*	Inhibit time	UNSIGNED16
04h*	Reserved	UNSIGNED8
05h*	Event timer	UNSIGNED16

* Only valid for 1800h ff

PDO mapping parameter

Table 1- 10 PDO mapping parameter 1600h ff, 1A00h ff

Subindex	Name	Data type
00h	Number of mapped objects in the PDO	UNSIGNED8
01h	First object to be mapped	UNSIGNED32
02h	Second object to be mapped	UNSIGNED32
03h	Third object to be mapped	UNSIGNED32
04h	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 logic device inside a CANopen device. If the PDO definition should be valid for the second logic device, the PDO number of the CANopen device used needs to be increased by the value 64 (40h) corresponding to the PDO number in the device profile (see following table).

Table 1- 11 Example of PDO number calculation

Logic device in CANopen device	PDO number in CANopen device	PDO number in device profile
1. logic device	PDO number + 0 (PDO1 to PDO64)	PDO number (PDO1 to PDO64)
2. logic device	PDO number + 64 (PDO65 to PDO128)	PDO number (PD O1 to PDO64)
3. logic device	PDO number + 128 (PDO129 to PDO192)	PDO number (PDO1 to PDO64)
4. logic device	PDO number + 192 (PDO193 to PDO256)	PDO number (PDO1 to PDO64)
5. logic device	PDO number + 256 (PDO257 to PDO320)	PDO number (PDO1 to PDO64)
6. logic device	PDO number + 320 (PDO321 to PDO384)	PDO number (PDO1 to PDO64)
7. logic device	PDO number + 384 (PDO385 to PDO448)	PDO number (PDO1 to PDO64)
8. logic 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 communications parameter index = 1400h + RPDO-number -1
- TPDO communications parameter index = 1800h + TPDO-number -1
- RPDO mapping parameter index = 1600h + RPDO-number -1
- TPDO mapping parameter index = 1A00h + TPDO- number -1

Data transfer types

The following PDO transmission types are available:

- Synchronous transmission
- Asynchronous transmission

In order for the communicating devices to 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:

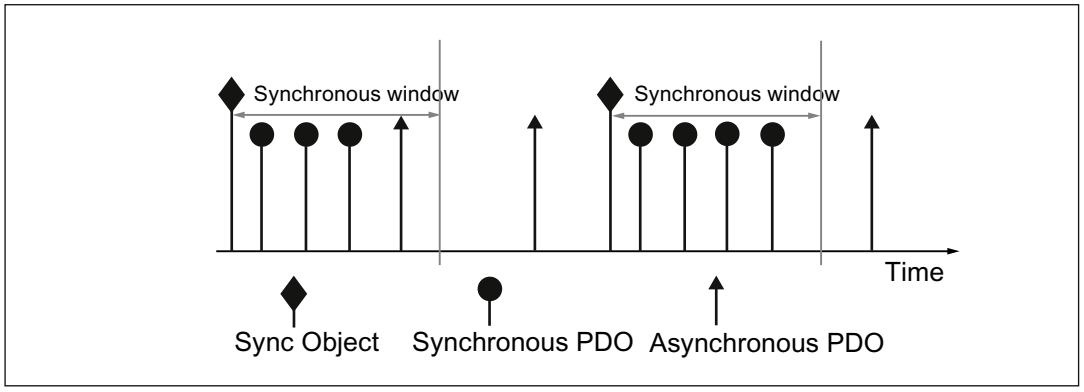


Figure 1-6 Principle of synchronous and asynchronous transmission

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 are transmitted to the application only after the next SYNC signal.

Note

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

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

PDO services

Transmission via PDO services takes place according to the producer/consumer principle:

- PDO number: PDO number [1..512] for each user type of a device
- User type: One of the values {consumer, producer}
- Data type: According to the PDO mapping

Write PDO

The "Write PDO" service follows the push model. There are either none or several consumers for a PDO.

The PDO has exactly one producer.

Using this service, the PDO producer transmits the data of the mapped application object to the individual consumers. The operational sequence for this service is as follows:

Table 1- 12 Write PDO

Producer	Write PDO	Consumer
Inquiry ---->	Process data	----> Display ----> Display ----> Display

SYNC service

The SYNC object is periodically sent 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 1005h and the time between two SYNC signals is defined in the object index 1006h.

1.14 BICO interconnection in conjunction with CANopen

Introduction

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 conditions.

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

- SINAMICS S120: See also the Installation and Start-Up Manual /IH1/ or the List Manual /LH1/.
- SINAMICS S110: See also the section "Commissioning" in the Function Manual or in the List Manual /LH7/.

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 "Interconnecting the receive and transmit buffer".

See also

BICO interconnection procedure in STARTER (Page 67)

Interconnecting the receive buffer (Page 112)

Interconnecting the transmit buffer (Page 115)

1.15 Behavior of SINAMICS when booting with the CANopen communications board

Introduction

SINAMICS only boots as a CANopen slave when the CANopen Communication Board is inserted.

Boot-up

During boot-up, the CAN software first queries the hardware address switch. If it is set to 0 or 127, the parameter of the CAN bus address can be written.

The address can be set via parameter **p8620**. If the address switch is set to 1-126, this address is copied to the CAN bus address parameters and displayed. The parameter is then read only.

Profile velocity mode

Servo and vector drives support the velocity mode profile.

This mode allows velocity setpoints to be defined.

See also

CAN interface (Page 87)

1.16 Bootup protocol

Bootup protocol

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

Table 1- 13 Bootup protocol COB-ID = 700h + Node-ID

NMT Master	Bootup protocol	NMT Slave
Display <----	Data	<---- Inquiry
	0	

1.17 Network management (NMT service)

Introduction

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

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

SINAMICS 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.

A detailed description of the NMT service is available in CANopen standard "CiA DS-301 V4.01 (Application Layer and Communication Profile)".

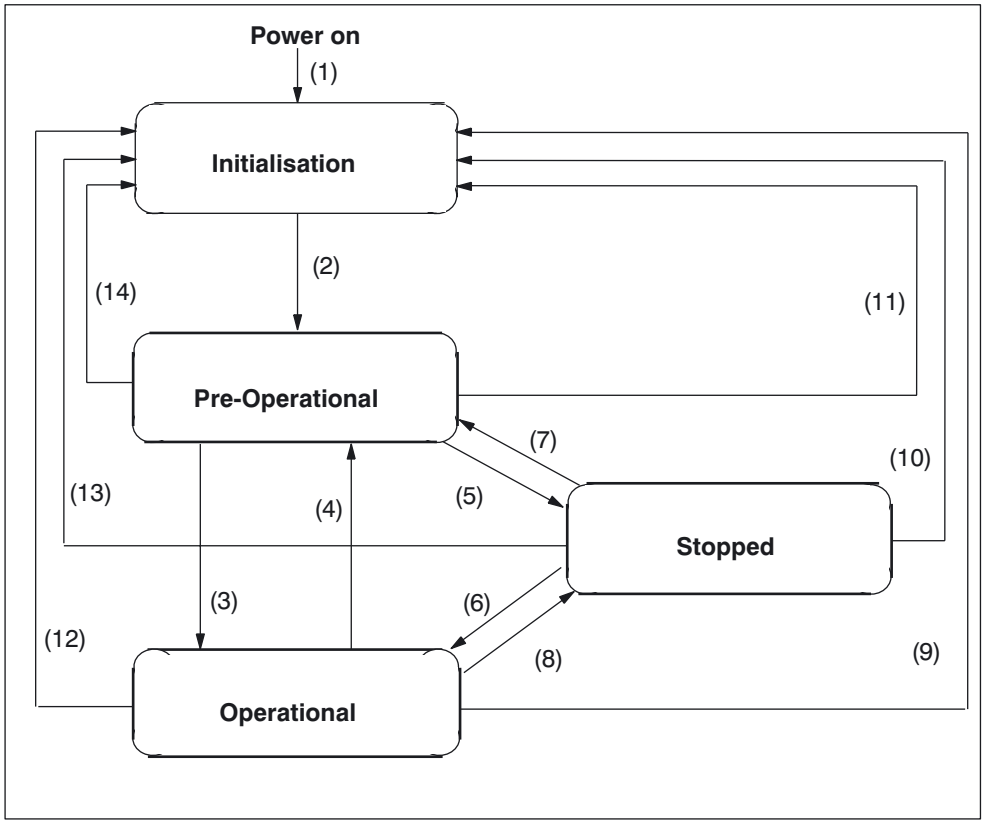


Figure 1-7 Status diagram of a CANopen node

Note

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

In the "Operational" status, 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- 14 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 status to "Operational". The drive can only transmit and receive process data in the "Operational" status.
- 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" status.
- Enter Pre-Operational:
  command for switching from "Operational" or "Stopped" to "Pre-Operational". In the "Pre-Operational" status, the node cannot process any PDOs. 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 status 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 status.

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- 15 COB-ID = 0

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

The parameter **p8684 CBC NMT state after power up / NMT power up state**
is used to set the CANopen NMT state, which is functional after power up or via the NMT services "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 CBC NMT states / NMT states**.

1.18 CANopen device state machine

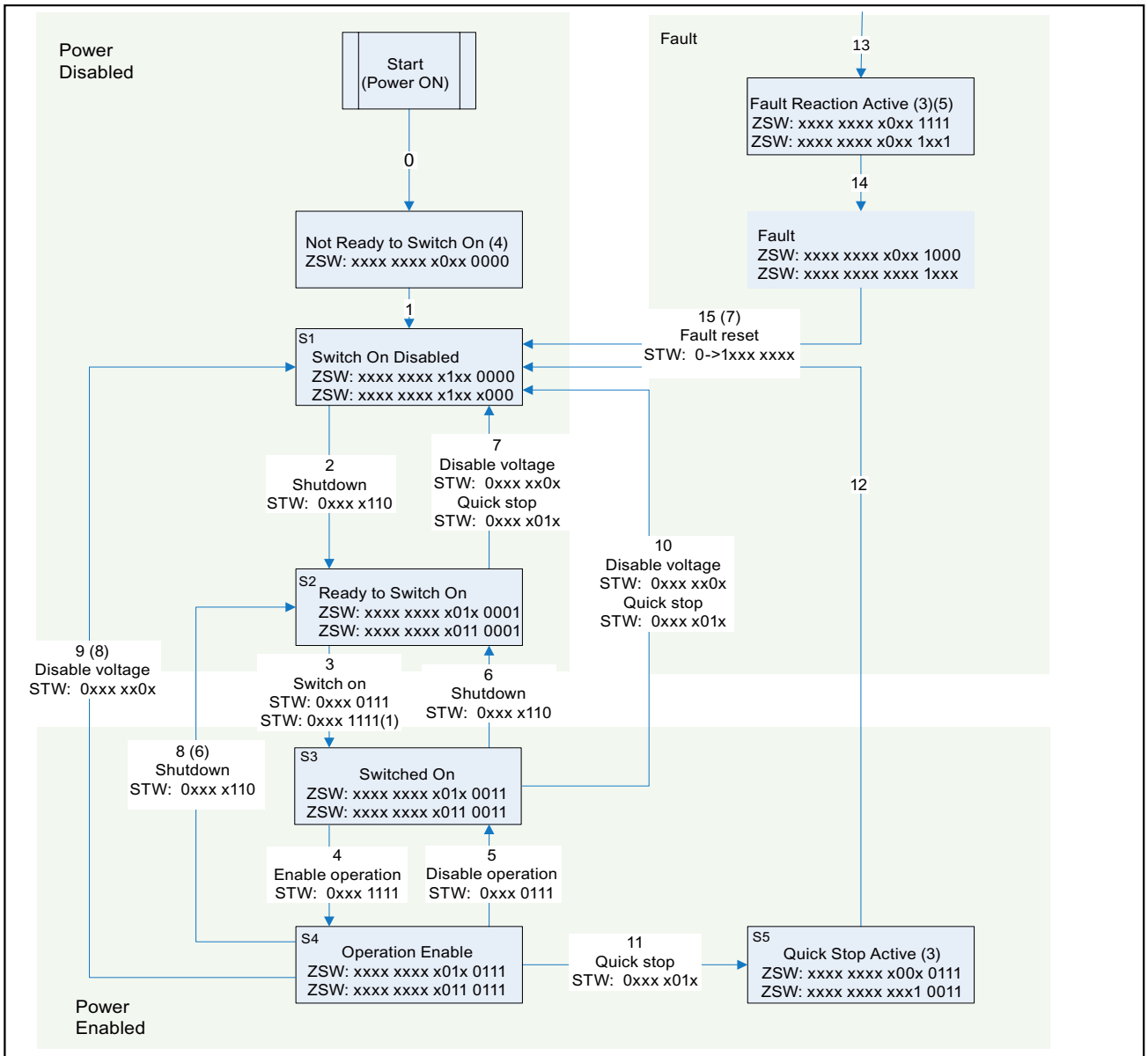
The CANopen device state machine describes the device status and the possible drive device status transitions. Each individual status describes a particular internal or external behavior. Depending on the drive device status, only certain transition commands are accepted.

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- 16 Statuses of the CANopen device machine

Status	Condition
Not Ready to Switch On	Low level power has been applied to the drive. The drive is being initialized. The drive function is disabled.
Switch On Disabled	Drive initialization is complete. Drive parameters may be changed. The drive function is disabled. High voltage may not be applied to the drive.
Ready to Switch On	The drive parameters may be changed. The drive function is disabled. High voltage may be applied to the drive.
Switched On	The drive parameters may be changed. The drive function is disabled. High voltage has been applied to the drive. The power amplifier is ready.
Operation Enable	No faults have been detected. The drive parameters may be changed. The drive function is enabled and power is applied to the motor. This corresponds to normal operation of the drive.
Quick Stop Active	The Quick Stop function is being executed. The drive parameters may be changed. The drive function is enabled and power is applied to the motor.
Fault Reaction Active	A non-fatal fault has occurred in the drive. The Quick Stop function is being executed. The drive parameters may be changed. The drive function is enabled and power is applied to the motor.
Fault	The drive parameters may be changed. A fault has occurred in 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 1-8 CANopen device state machine

1.19 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.

Node guarding
The NMT master sends 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, the NMT master informs its master application of this.

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) sends out heartbeat messages periodically. One or several CANopen devices in the network recognize this heartbeat message. If the "Heartbeat producer" misses a heartbeat cycle, the local application is informed using a "Heartbeat event".

Note

For effective fault monitoring, one of the two methods must be activated.
"Node guarding" and "heartbeat" are deactivated in the basic setup.

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 "Life guarding event" is created if

- 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.

Resolved faults in the network ("Remote node error") are displayed not only in a "Node guarding event" but also in a "Life guarding event".

In the case of an error in the CAN communication, e.g. too many telegram failures, the fault F08700 is reported (for details see List Manual S120). 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- 17 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" **

** for monitoring faults

s : Status of the NMT slave
4: Stopped
5: Operational
127: Pre-Operational

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 should be 0. The toggle bit in the "Guarding protocol" should be reset to 0 if the NMT state "Reset communication" is run (no other state 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- 18 Node guarding 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 |	t	s	| Heartbeat consumer time |
Inquiry ---->	Byte 0, bit 7	Byte 0, Bit 6..0	----> Signaling
	t	s	| Heartbeat consumer time |
			After expiry of the heartbeat consumer time: Heartbeat event

r: reserved (always 0)

s: Status of the "heartbeat producer"
0: Boot-up
4: Stopped
5: Operational
127: Pre-Operational

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.

1.20 Parameters, save, restore factory settings

Introduction

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 1000hex - 1FFFhex) 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 6000hex - 9FFFhex) 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.

When reading subindices 1.3, a value of 1 is obtained, which 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 1000hex - 1FFFhex) 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 6000hex - 9FFFhex) of the drive are reset to their factory setting.

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

See also

Control Unit communication objects independent of the drive (Page 129)

1.21 CAN bus sampling time

Description

The sampling time of the CANopen communication board 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 the period of 4 ms.
For cyclically received message frames, the cycling time must be larger than at least double the sampling time, according to the Shannon sampling theorem.

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

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.

1.22 Number of controllable drives via CAN bus

Description

Depending on the request to the drive, up to 4 axes can be controlled simultaneously using the CAN bus, where for CANopen a maximum of 4 transmit message frames and 4 receive message frames can be used for communication. If all 8 of the maximum possible communication message frames are used for drives, only 3 drives can be controlled -> 3 * 8 = 24.

The maximum possible number of drives is limited by

1. the demands on the drives (e.g. slave drives often need less than 8 message frames)
2. the maximum number of communication message frames, i.e 25.

The same rules apply for the number of controllable drives as for PROFIBUS – up to the number of maximum controllable drives/axes.

Servo control

- maximum of 4 axes controllable.

Vector control

- maximum of 2 axes controllable.

Prerequisites for commissioning 2

Section content

This section describes the commissioning requirements:

- SINAMICS S120: CBC10 Communication Board
- SINAMICS S110: CU305 CAN
- STARTER commissioning tool

Note

All parameters, faults, alarms, and function diagrams for CANopen in the SINAMICS drive line-up are described in the SINAMICS S List Manuals.

2.1 Prerequisites for commissioning

Overview

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

- SINAMICS S120:
  - Communication board CBC10 (CAN communication board)
  - Memory card with firmware
- SINAMICS S110:
  - CU305 CAN with firmware in non-volatile memory
- Connecting the Control Unit PROFIBUS interface to a PC/PG with PROFIBUS interface
- The STARTER commissioning tool must be installed on your PC/PG

Note

For descriptions of the components in a SINAMICS drive line-up and instructions on wiring the components, connecting the PROFIBUS interface to a PC/PG, and installing the STARTER commissioning tool, see the following:

- SINAMICS S120: Equipment Manuals /GH1/ and /GH2/ and Commissioning Manual /IH1/
- SINAMICS S110: Equipment Manual /GH8/ and Function Manual /FH3/

2.2 CBC10 Communication Board for CAN Bus (SINAMICS S120)

Introduction

The CBC10 CANopen communication board (CAN communication board) is used to connect drives in the SINAMICS S120 drive system to higher-level automation systems with a CAN bus.

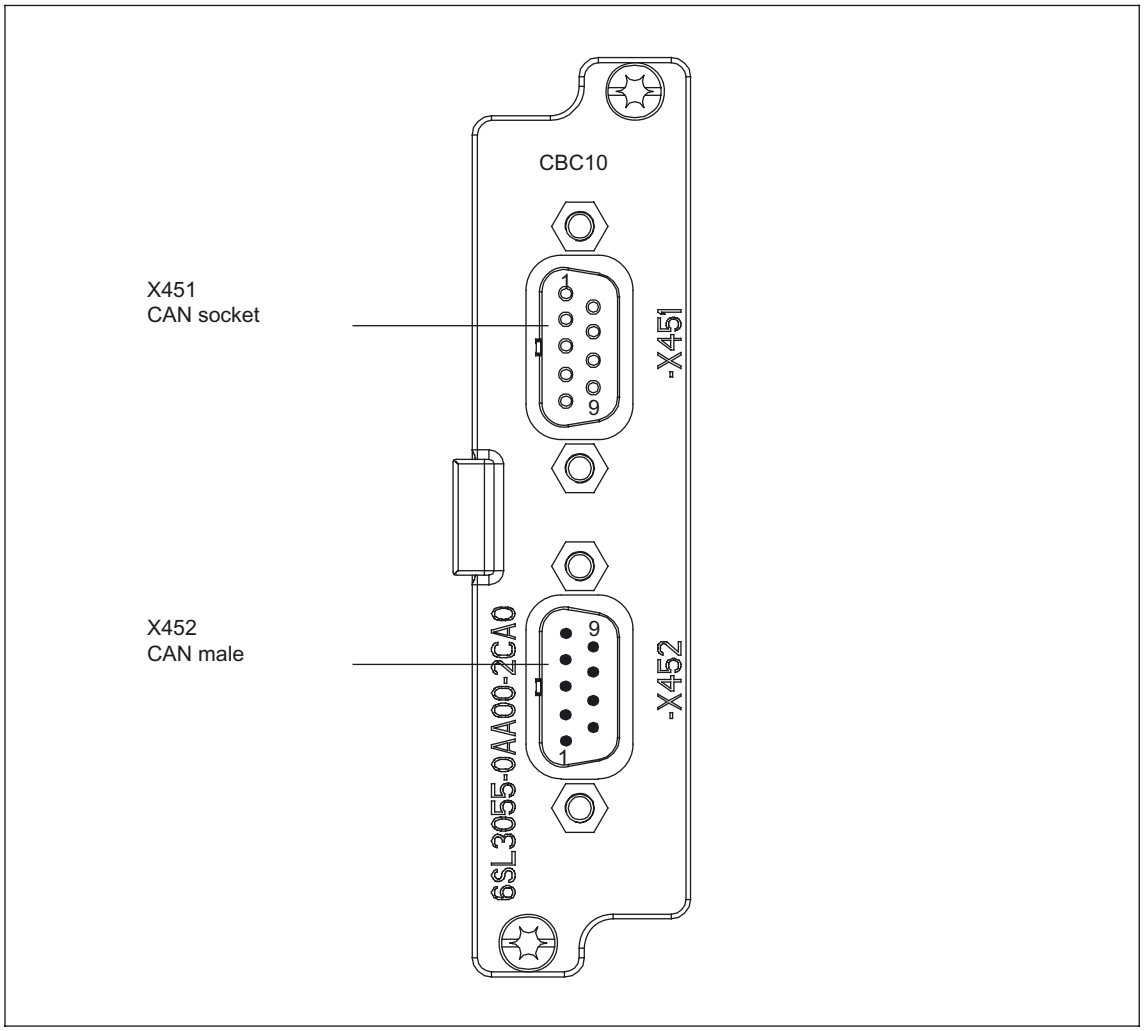


Figure 2-1 View of the CBC10 Option Board

The CANopen Option Board uses two 9-pin Sub D connectors for the connection to the CAN bus system.

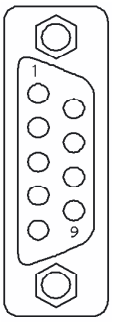
The connectors can be used as inputs or outputs. Unused pins are plated through.

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

CAN bus interface X451

The X451 CAN bus interface has the following socket assignments:

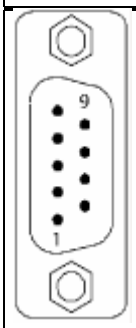
Table 2- 1 CAN bus interface X451

	Pin	Designation	Technical specifications
	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			

CAN bus interface X452

The X452 CAN bus interface has the following socket assignments:

Table 2- 2 CAN bus interface X452

	Pin	Designation	Technical specifications
	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 Communication Board is installed on the option slot on Control Unit CU320 as follows (see diagram below):

1. Unscrew and remove the protective cover.
2. Insert the Communication Board.
3. Secure the Communication Board.

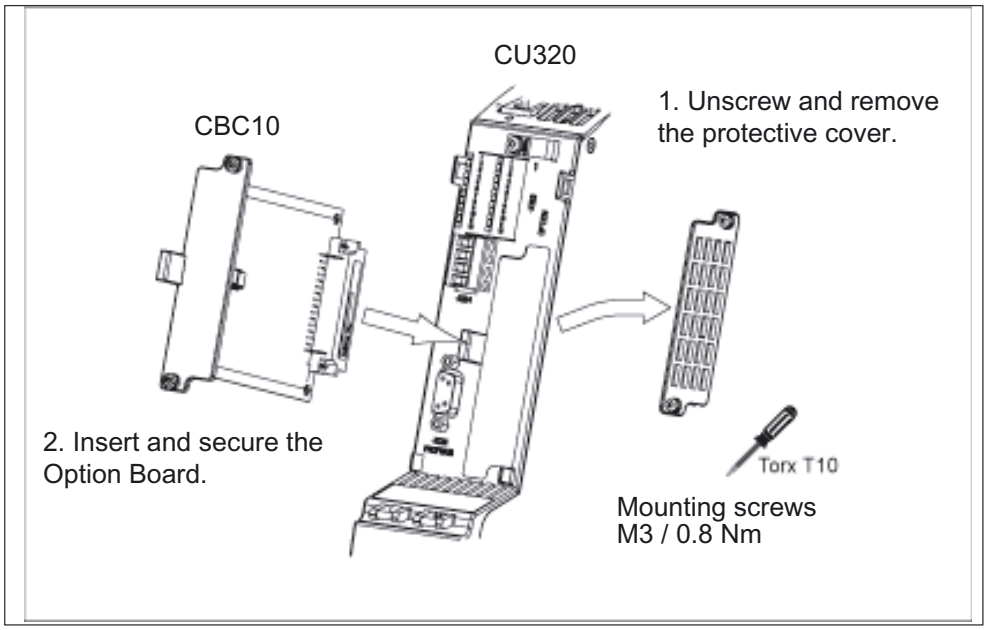


Figure 2-2 Installing the CBC10 Communication Board

2.2.2 CANopen functionality

Introduction

The CBC10 supports the CANopen transfer types with SDOs (service data objects) and PDOs (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->parameter channel that can be used to read or write all the SINAMICS parameters.

The CBC10 firmware supports profile velocity mode.

Node 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.

2.2.3 Diagnostics LED "OPT"

Overview

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

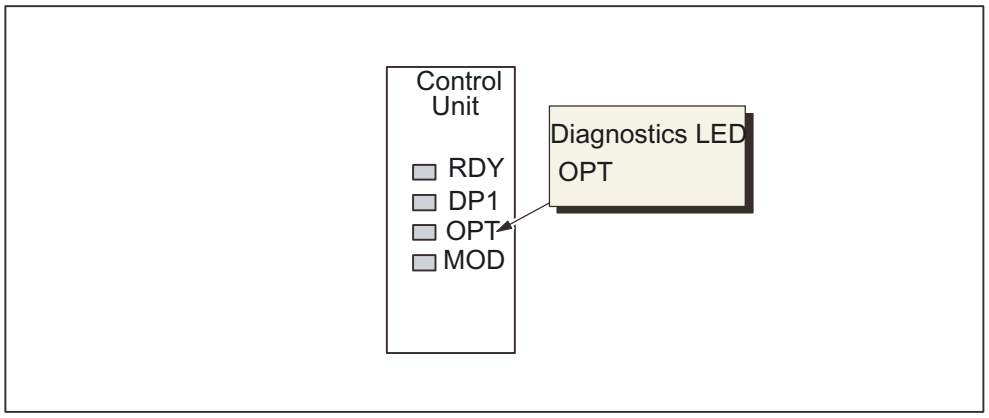


Figure 2-3 Overview of the LEDs on the Control Unit

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 (see the appropriate table below) indicates whether or not an error is present.
- Diagnostics LED OPT -> green (see the appropriate table below) indicates the status of the nodes 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 operate
Single flash	Warning limit reached	At least one of the error counters in the CAN controller has reached the warning threshold "Error passive". (too many message frames with errors).
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 status.
Flashing	PRE-OPERATIONAL	The node is in the PRE-OPERATIONAL status.
On	OPERATIONAL	The node is in the OPERATIONAL status.

2.3 CU305 CAN (SINAMICS S110)

2.3.1 CAN bus on the CU305

Introduction

The CU305 can either be ordered with an integrated PROFIBUS interface or with an integrated CANopen interface.

The version with the CANopen interface will be described here.

This interface is used to connect drives in the SINAMICS S110 drive system to higher-level CAN bus automation systems.

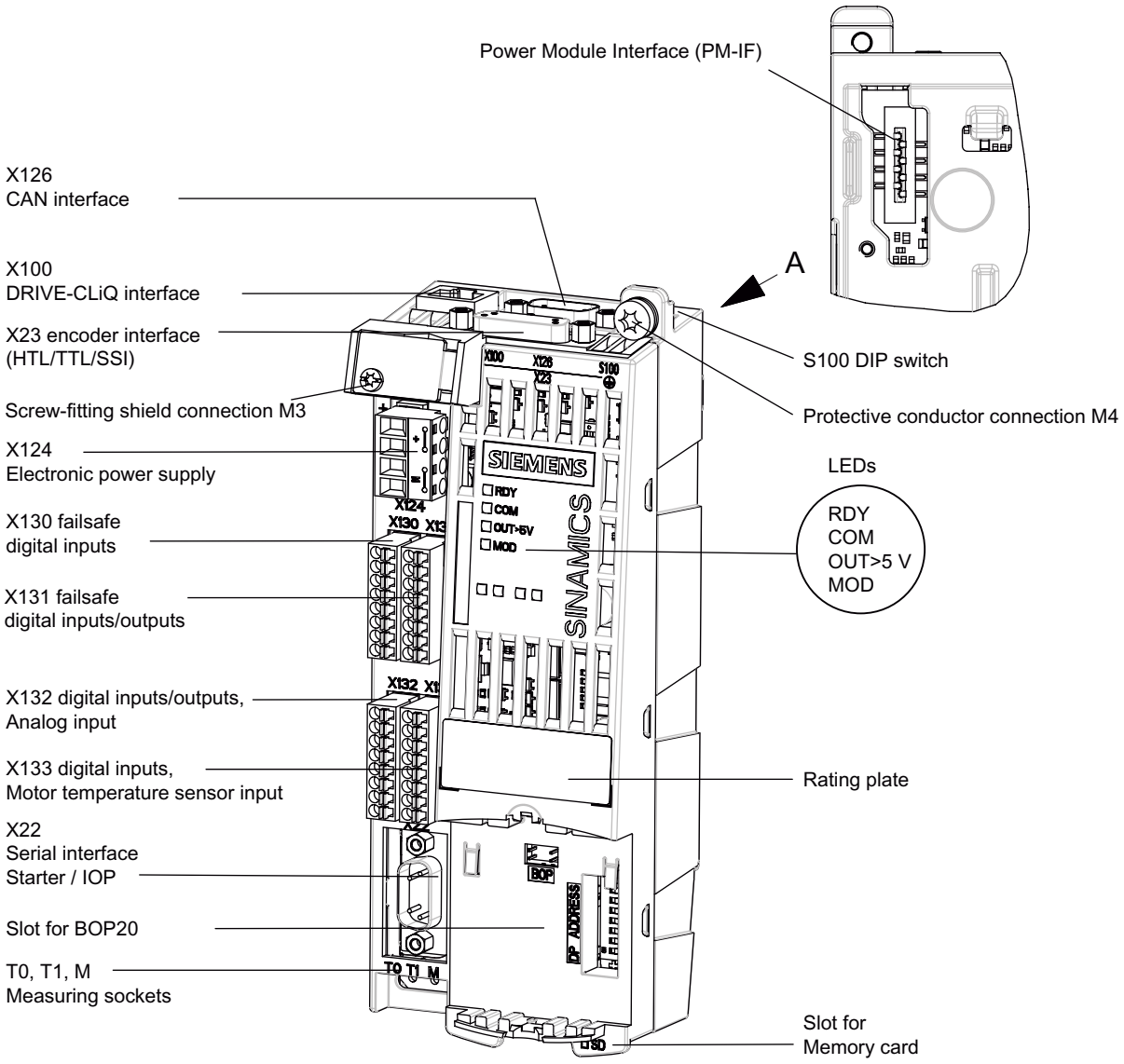
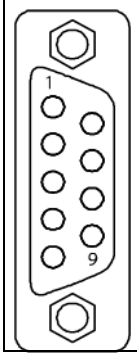


Figure 2-4 View CU305 CAN

The CU305 CAN is connected to the CANopen bus system via the X126, a 9-pin Sub D connector. The connector can be used as an input or an output. Unused pins are plated through.

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

Table 2- 5 X126 CAN interface

	Terminal	Designation	Technical specifications
	1	Reserved, do not use	
	2	CAN_L	CAN signal
	3	CAN_GND	CAN ground
	4	Reserved, do not use	
	5	CAN_SHL	Optional shield
	6	CAN_GND	CAN ground
	7	CAN_H	CAN signal
	8	Reserved, do not use	
	9	Reserved, do not use	

⚠ CAUTION

If the CAN interface is connected to the PROFIBUS connector, then this can destroy the CAN interface.

2.3.2 CANopen functionality CU305 CAN

Introduction

The CU305 CAN supports the CANopen transmission types with SDOs (service data objects) and PDOs (process data objects).

The CU305 CAN also supports free PDO mapping.

The CU305 CAN 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 CU305 CAN supports node guarding and the heartbeat protocol (heartbeat producer).

The CU305 CAN features an SDO->parameter channel that can be used to read or write all the SINAMICS parameters.

The CU305 CAN firmware supports profile velocity mode.

Node 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.

2.3.3 Diagnostics LED "COM"

Diagnostics LED COM

Table 2- 6 Diagnostics LED -> red (CANopen error LED)

ERROR LED flashing frequency	Status	Description
Off	No error	Ready to operate
Single flash	Warning limit reached	At least one of the error counters in the CAN controller has reached the alarm threshold "Error passive". (too many telegrams with errors).
Double flash	Error control event	A guard event has occurred.
On	Bus off	The CAN controller is "Bus off".

Table 2- 7 Diagnostics LED -> green (CANopen RUN LED)

RUN LED flashing frequency	Status	Description
Single flash	Stopped	The node is in the STOPPED status.
Flashing	PRE-OPERATIONAL	The node is in the PRE-OPERATIONAL status.
On	OPERATIONAL	The node is in the OPERATIONAL status.

2.4 STARTER commissioning tool

Start

To launch STARTER, click the STARTER icon or choose Start > SIMATIC >STARTER in the Windows Start menu.

2.4.1 The STARTER user interface

You can use STARTER to create the sample project. The different areas of the user interface are used for different configuration tasks (refer to diagram below):

- Project navigator: this area displays the elements and objects that can be added to your project.
- Working area: you create the project in this area:
  - When you are configuring the drive, this area contains the Wizards that help you configure the drive objects.
  - You can configure the parameters for the speed setpoint filter, for example.
  - When you call up the expert list, the system displays a list of all the parameters that you can view or change.
- Detail view: this area contains detailed information on faults and alarms, for example.

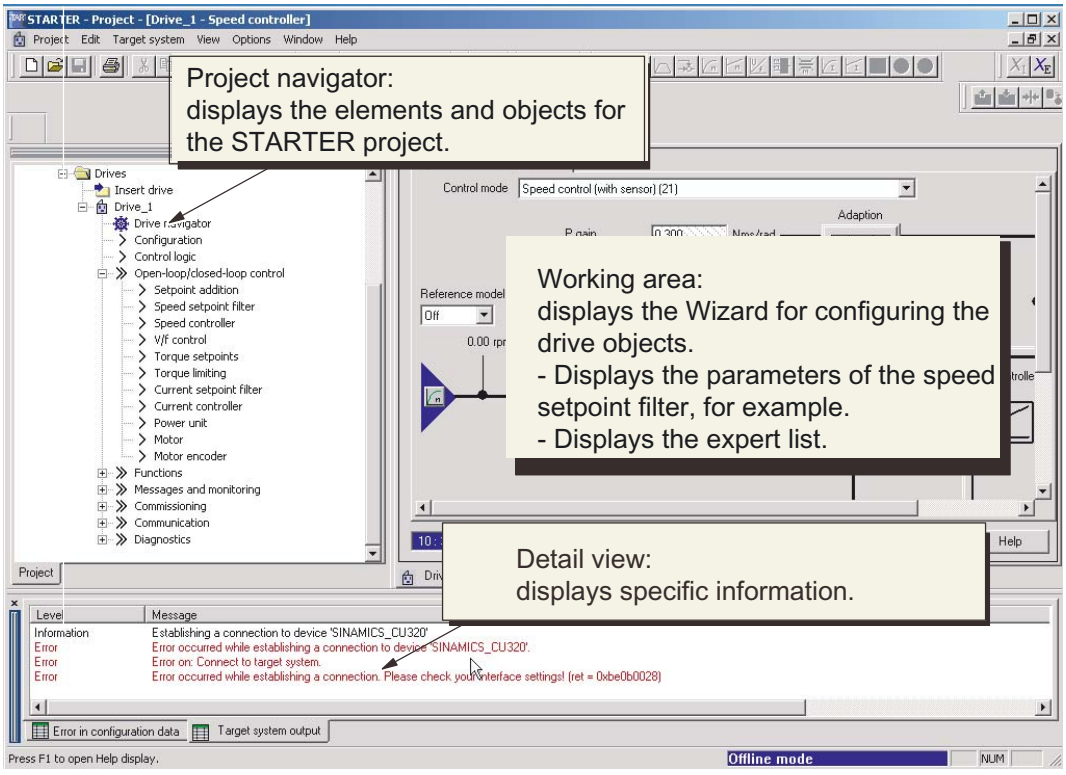


Figure 2-5 The different areas of the STARTER user interface

2.4.2 BICO interconnection procedure in STARTER

Introduction

You can parameterize the drive settings on the Motor Module in 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 BICO interconnection procedure in STARTER.

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 r parameter **r8890[0]**, for example.

1. In the project navigator, call up the expert list as follows: **Drive_1 > right-click > Expert > Expert list**.
2. Search for parameter **p0840**.

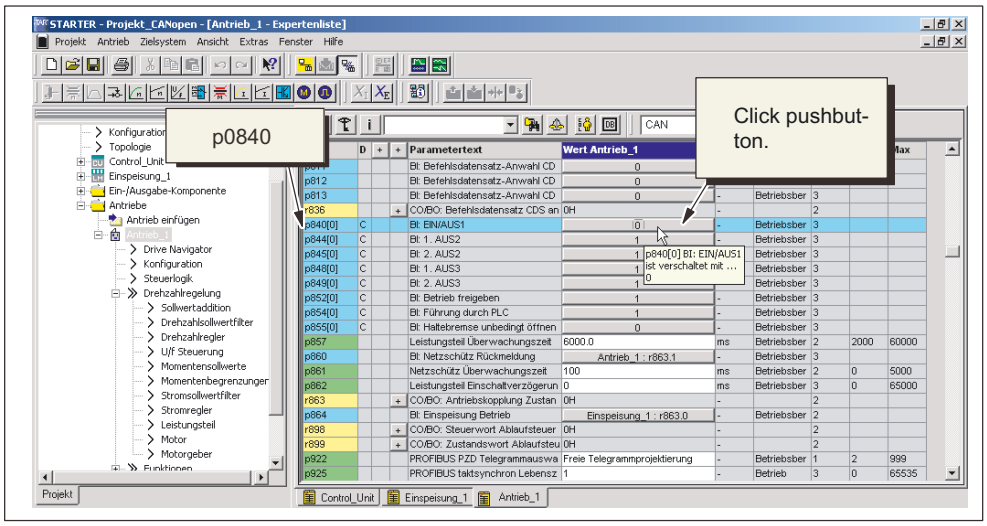


Figure 2-6 Interconnect 1

3. Click the pushbutton to interconnect with an r parameter.

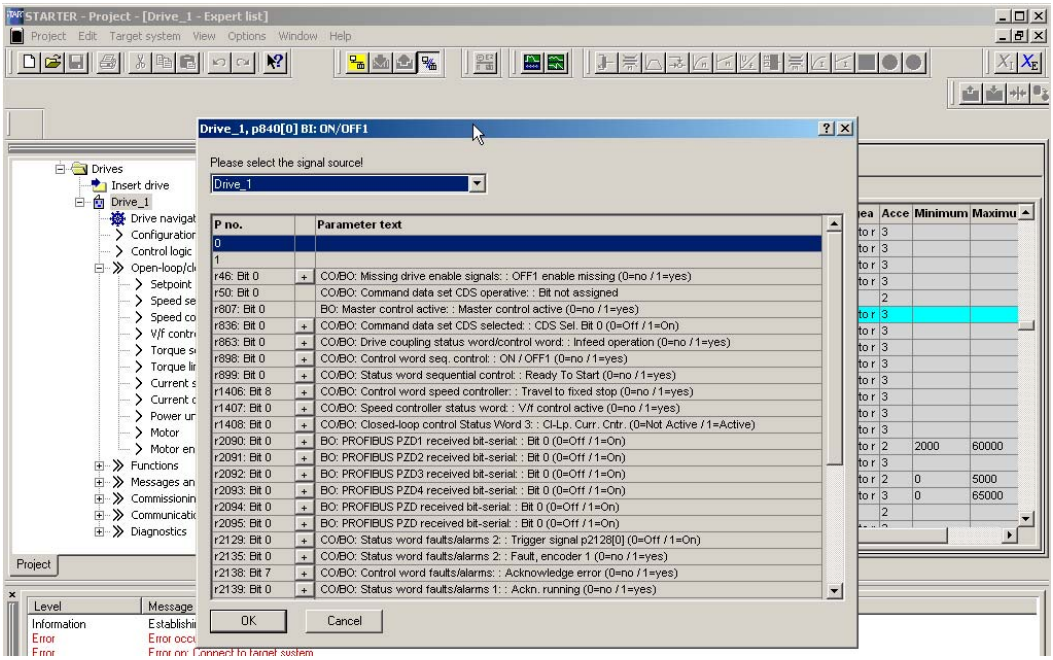


Figure 2-7 Interconnect 2

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

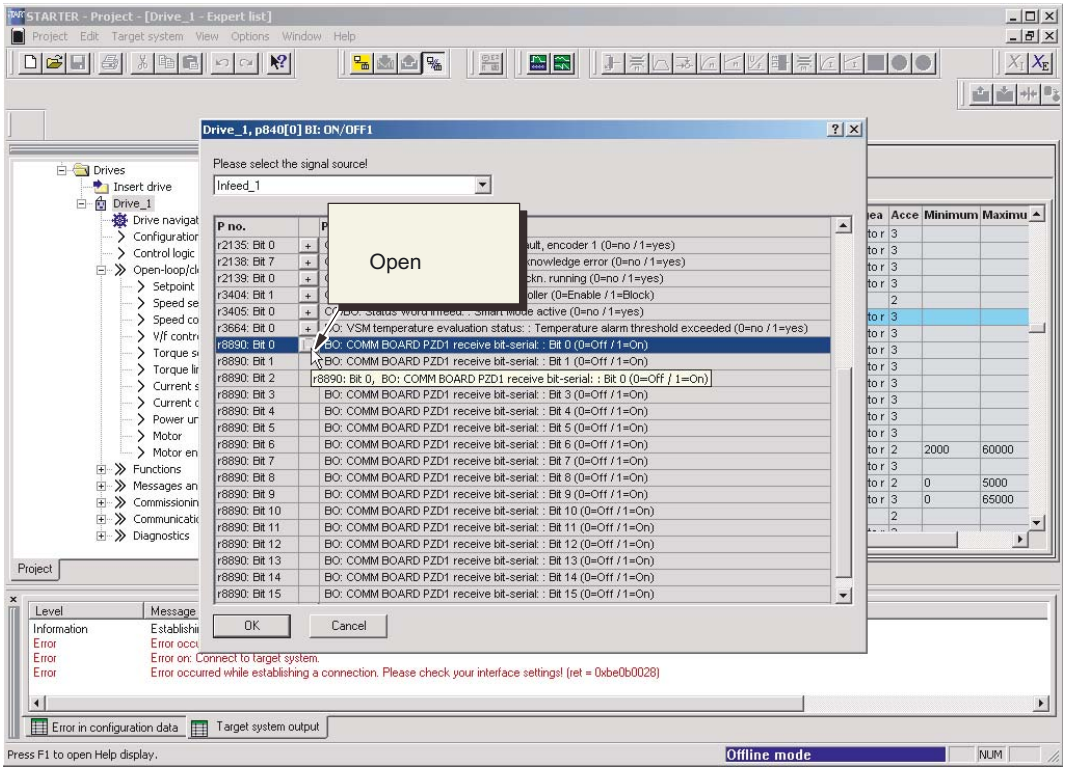


Figure 2-8 Interconnect 3

5. Open the 16 bit of r parameter **r8890**.

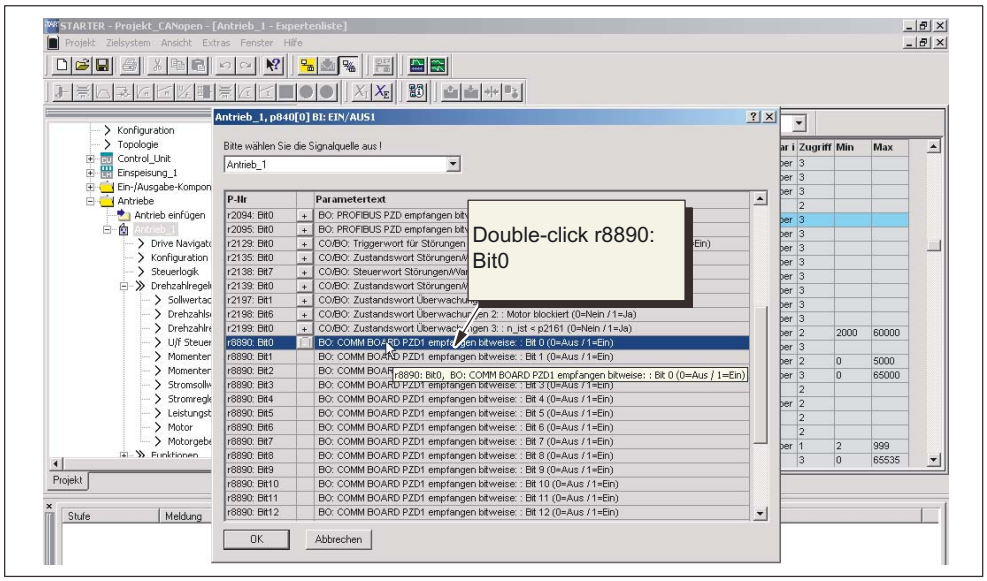


Figure 2-9 Interconnect 4

6. Double-click **r8890: Bit0.**

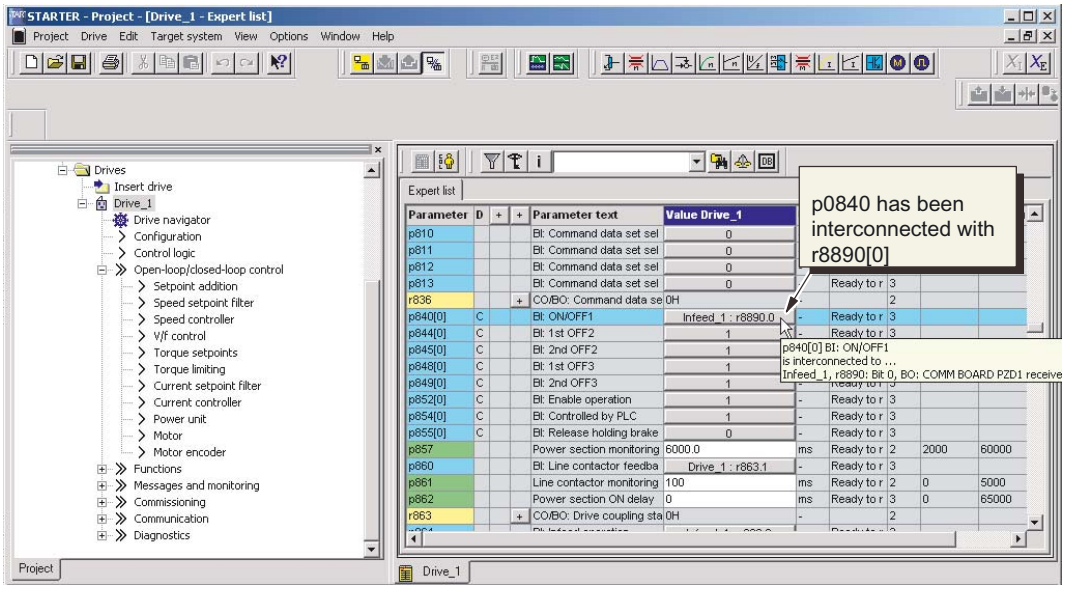


Figure 2-10 Interconnect 5

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

Graphical screen interface

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

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

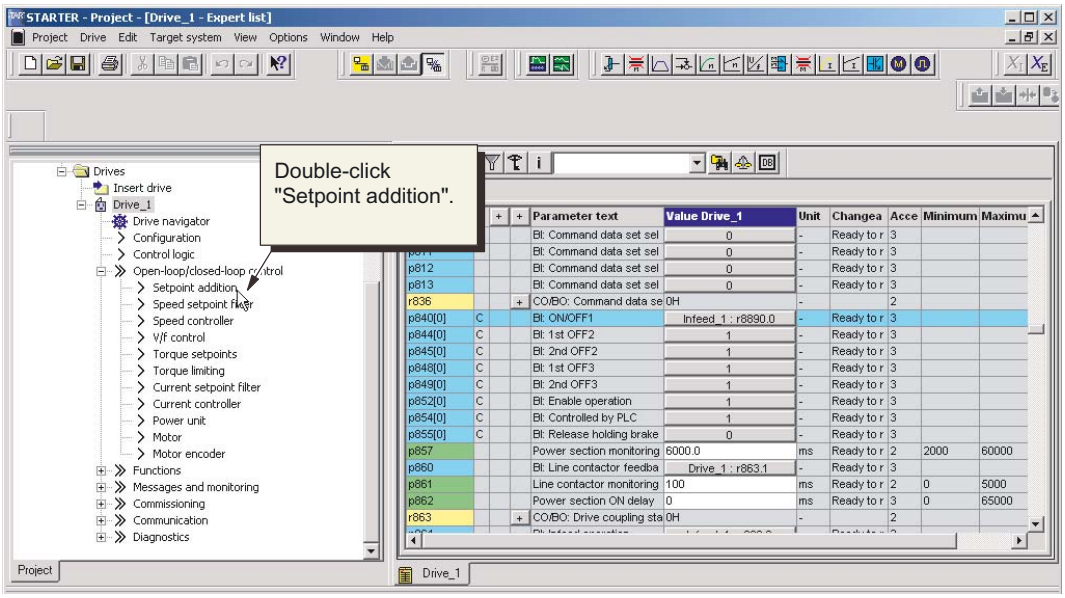


Figure 2-11 Interconnection via graphical screen interface 1

1. In the project navigator under **Drive_1 > Open-loop/closed-loop control**, double-click the selection **Setpoint addition**.

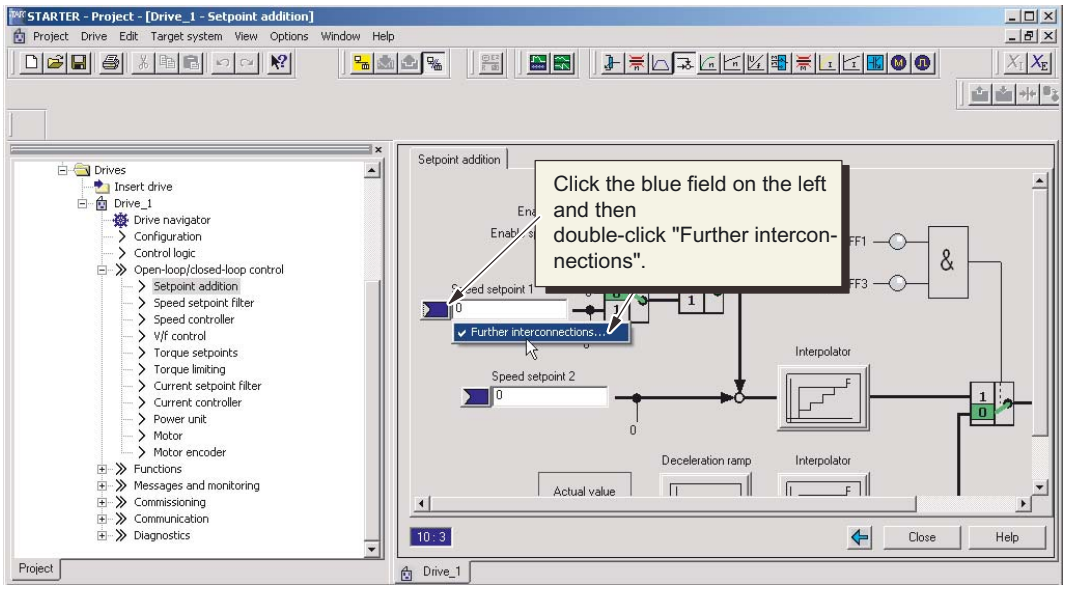


Figure 2-12 Interconnection via graphical screen interface 2

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

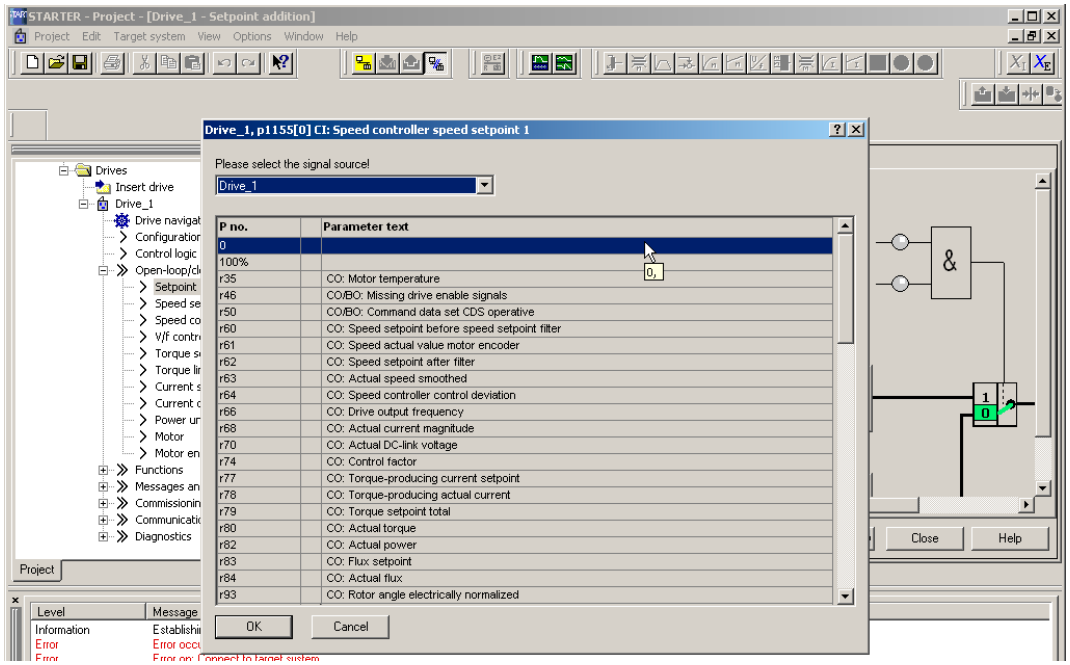


Figure 2-13 Interconnection via graphical screen interface 3

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

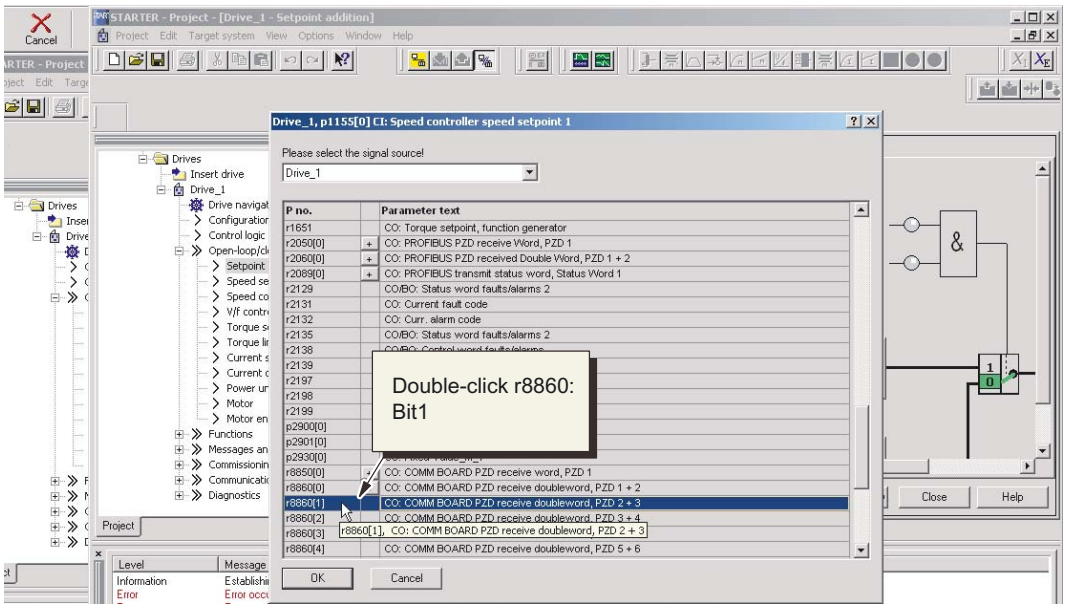


Figure 2-14 Interconnection via graphical screen interface 4

4. Double-click **r8860: Bit1**.

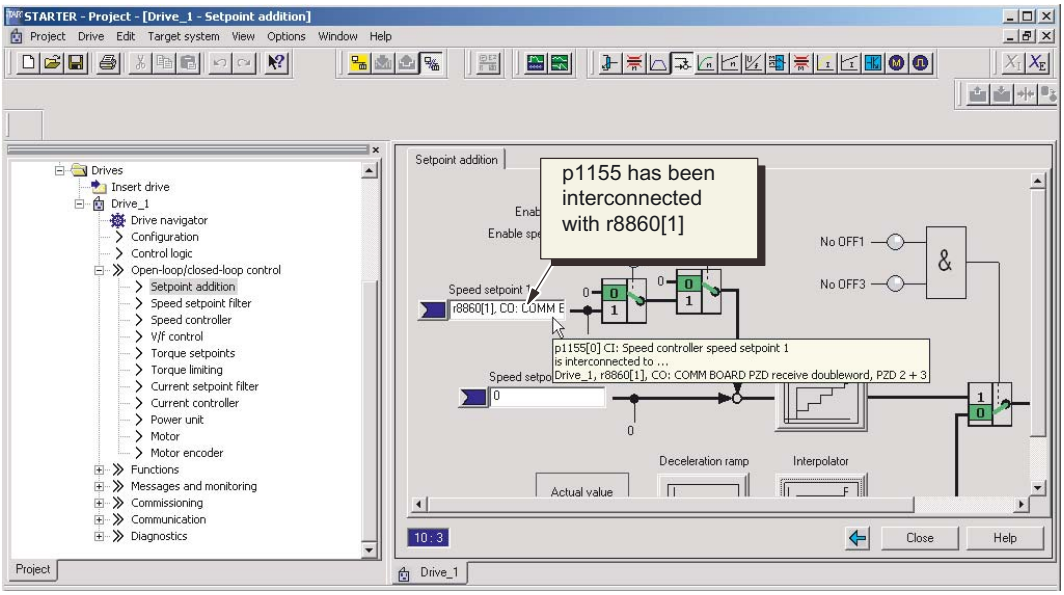


Figure 2-15 Interconnection via graphical screen interface 5

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

See also

BICO interconnection in conjunction with CANopen (Page 43)

Interconnecting process data in the receive and transmit buffers (Page 108)

2.4.3 Entering parameter values in the expert list

Introduction

Note

Parameters can be entered and displayed in the expert list as **decimal** or hexadecimal values.

Example

If you want to process decimal values as **hexadecimal** or **binary** values, use the calculator on your PC and carry out the following:

1. Copy the decimal value in the expert list to your calculator.
2. Display this decimal value on your calculator as either a hexadecimal or binary value.
3. Change the value.
4. Use your calculator to convert the value back to a decimal value.
5. Copy this decimal value to the expert list.

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.

A table is provided showing the individual commissioning steps in which the current step is highlighted in **bold**.

The initial commissioning procedure described here takes place with STARTER in 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".

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:

Table 3- 1 CANopen initial commissioning

Step	Procedure
1	Hardware settings on CBC10 (not for SINAMICS S110)
2	Configure the drive unit using the STARTER commissioning tool in 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 ONLINE mode, load the projects from the drive unit to the PC/PG and save.

See also

Hardware settings on CBC10 (not for SINAMICS S110) (Page 77)

Configuring the drive unit using the STARTER commissioning tool (Page 79)

Configure the COB IDs and process data objects for the receive and transmit message frames. (Page 97)

Interconnecting process data in the receive and transmit buffers (Page 108)

Load the projects from the drive unit to the PC/PG in ONLINE mode and save them (Page 119)

3.2 Example configurations

3.2.1 SINAMICS S120 example configuration

Overview

Note

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

- One infeed (Active Line Module)

- Two drives (Single Motor Module)

For a more detailed description of this configuration, refer to the Chapter "Initial commissioning using servo as an example" in the Commissioning Manual SINAMICS S120 /IH1/.

This Commissioning Manual extends the description of "Initial commissioning using servo as an example" to include a description of the initial commissioning procedure for the CANopen communication interface with the CBC10 Communication Board.

The following overview diagram illustrates the drive objects. From the point of view of CANopen and with this arrangement:

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

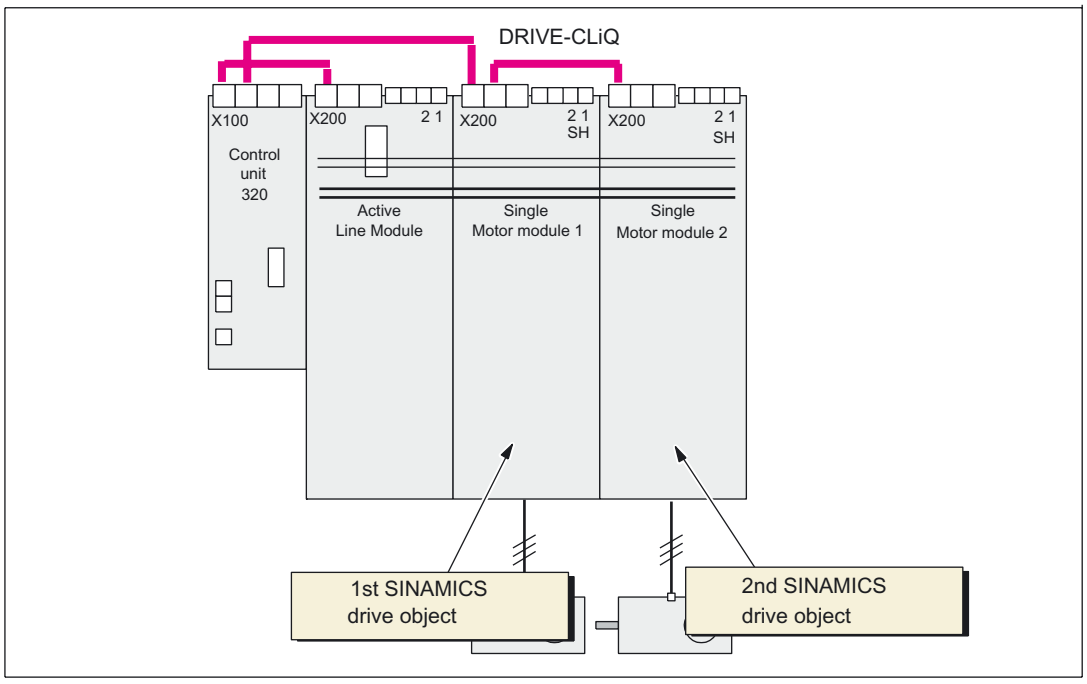


Figure 3-1 SINAMICS S120 components (example)

3.2.2 SINAMICS S110 example configuration

Overview

Note

The commissioning procedure described here is based on the sample configuration (as shown in the diagram below) in a SINAMICS drive line-up for an S110 drive: This example supplements the description of "Initial commissioning using servo as an example" to include a description of the initial commissioning procedure for the CANopen communication interface.

From CANopen's point of view, the Power Module 340 is the first SINAMICS drive object in the arrangement shown below.

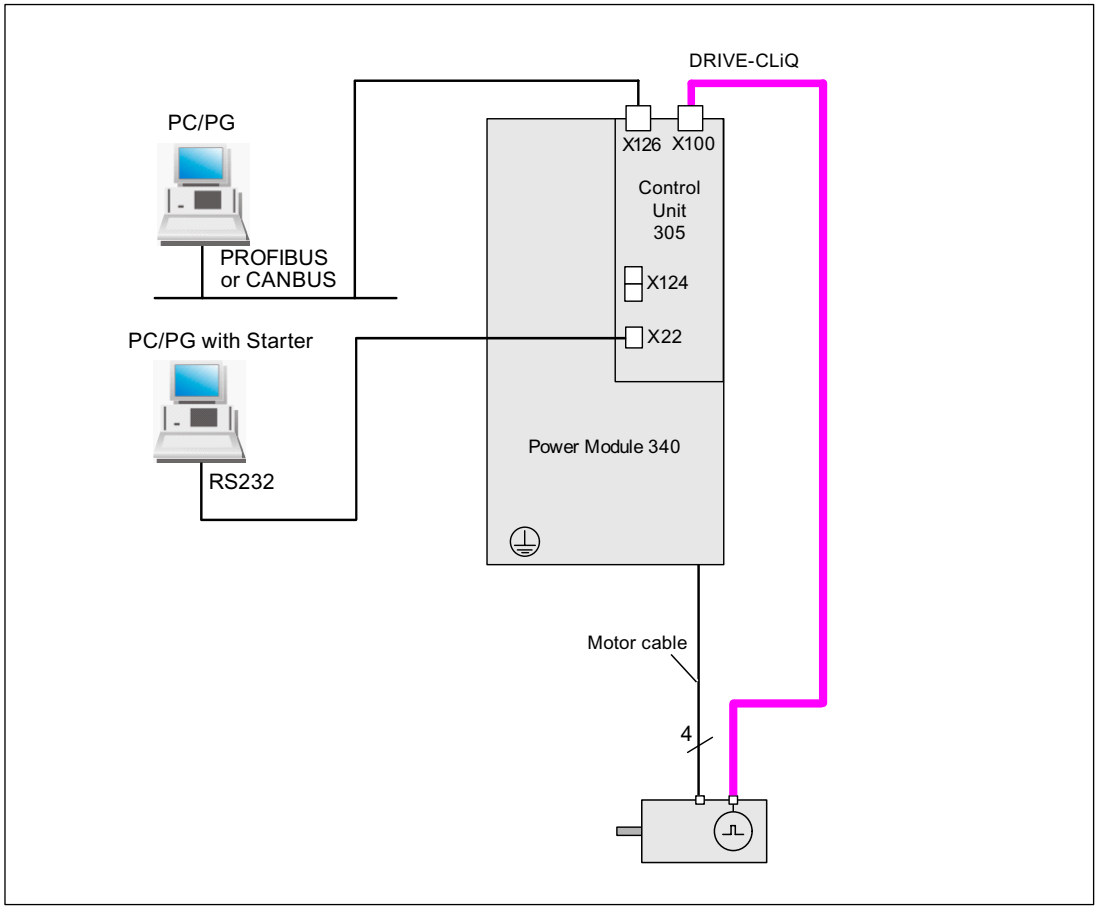


Figure 3-2 SINAMICS S110 components

3.3 Hardware settings on CBC10 (not for SINAMICS S110)

Initial commissioning: procedure

In the table below, the current commissioning step is highlighted in **bold**:

Table 3- 2 CANopen initial commissioning

Step	Procedure
1	**Hardware settings on CBC10 (not for SINAMICS S110)**
2	Configure the drive unit using the STARTER commissioning tool in 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 ONLINE mode, load the projects from the drive unit to the PC/PG and save.

Carrying out the commissioning step

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

- Bus terminating resistor
- Operation with/without ground

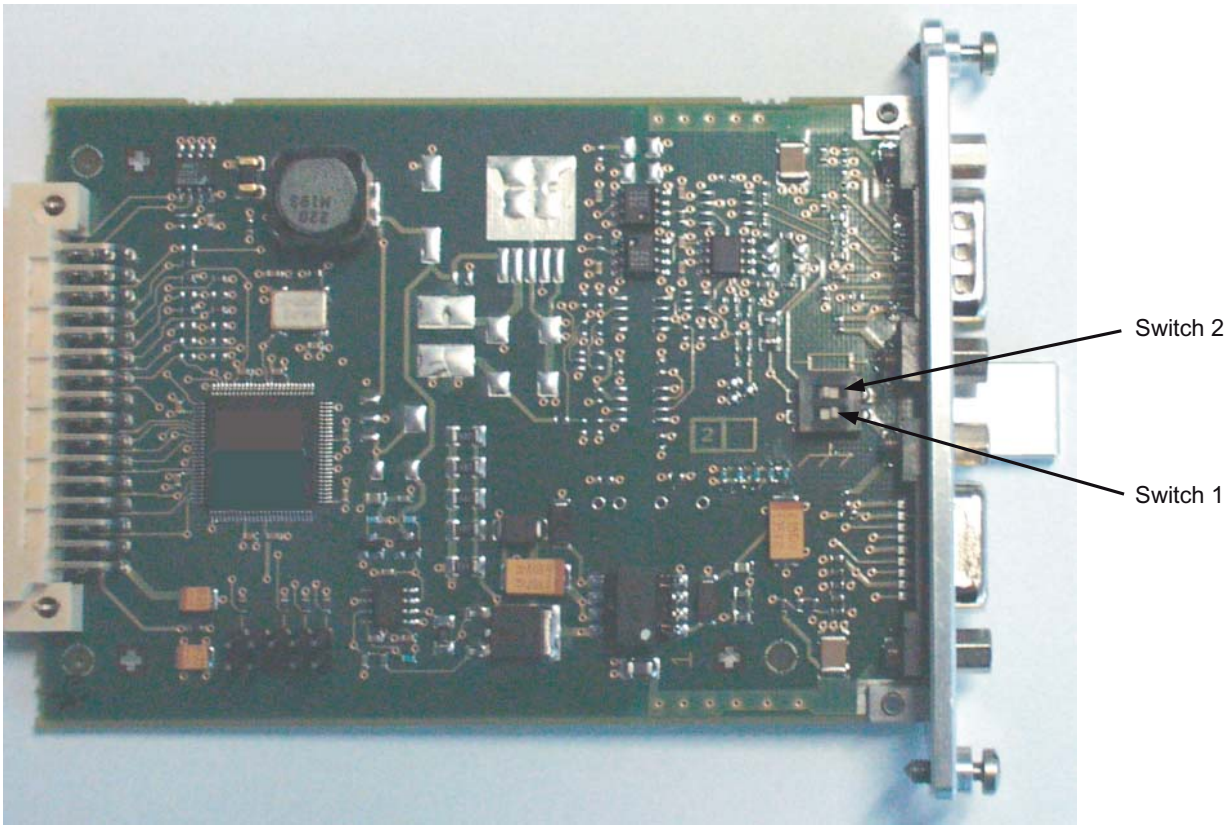
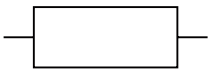
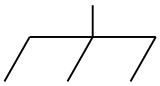


Figure 3-3 Switch 2/1

Table 3- 3 2-pin SMD DIL switch

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

Note

In a line-up comprising more than one drive unit (bus nodes), switch 2 must be set to **ON** for the last bus node and **OFF** for the preceding bus node.

3.4 Configuring the drive unit using the STARTER commissioning tool

Initial commissioning: procedure

In the table below, the current commissioning step is highlighted in **bold**:

Table 3- 4 CANopen initial commissioning

Step	Procedure
1	Hardware settings on CBC10 (not for SINAMICS S110)
2	**Configure the drive unit using the STARTER commissioning tool in 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 ONLINE mode, load the projects from the drive unit to the PC/PG and save.

Carrying out the commissioning step

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

- Search for the drive unit ONLINE.
- The component topology and configuration of the drive unit are determined automatically.
- Configure the motor.
- Configure the interface for the CBC10 Option Board on the Control Unit.
  - CAN interface
  - PDO message frame
  - Monitoring
- Load the project to the drive unit.

Note

When carrying out these steps, refer to the Installation and Start-Up Manual /IH1/ for SINAMICS S120.

3.4.1 Searching for the drive unit ONLINE

Introduction

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

Steps

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

1. Click the STARTER icon or choose **Start > Programs > STARTER** in the Windows Start menu to launch the STARTER commissioning tool.
   The STARTER Project Wizard is displayed.
2. Choose **Find drive units online....**

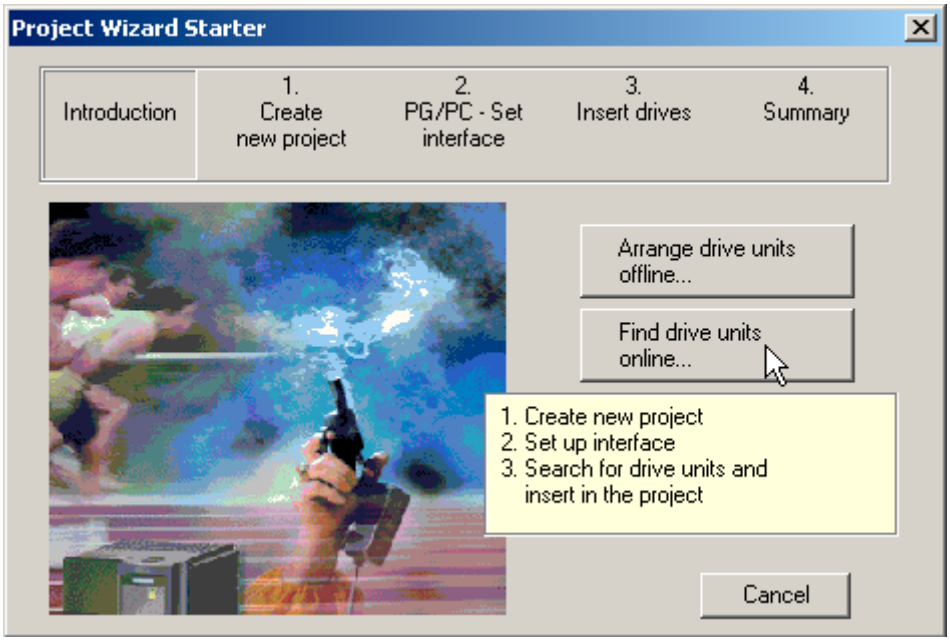


Figure 3-4 Find drive units online...

3. The Wizard guides you through the procedure for creating a new project. In the next dialog box, enter a name for the project (e.g. **Project_CANopen_0** and click **Continue >**.
4. The Project Wizard searches for the drive unit ONLINE and inserts it in the project. Click **Continue >**. The Wizard displays a summary of the project.
5. Choose **Fertig stellen (Complete)**. The new project and drive unit are displayed in STARTER.

Note

The system searches for drive units or, more precisely, Control Units; in other words if more than one Control Unit exists in the system, more than one drive unit is found. The peripheral components of a drive unit (Control Unit, Active Line Module, and so on) are not yet displayed. you carry out **Automatic configuration**.

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

Introduction

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.

Steps

1. Select the drive unit **Drive_Unit_Adr126** in the project navigator.
2. Choose **Connect to target system**. An ONLINE connection is established and the dialog box **Online/OFFLINE comparison** is displayed.
3. Click **Close** and, if necessary, restore the factory settings.
4. Select the drive unit **Drive_Unit_Adr126** in the project navigator.
5. Choose **Restore factory settings** (see screenshot below).

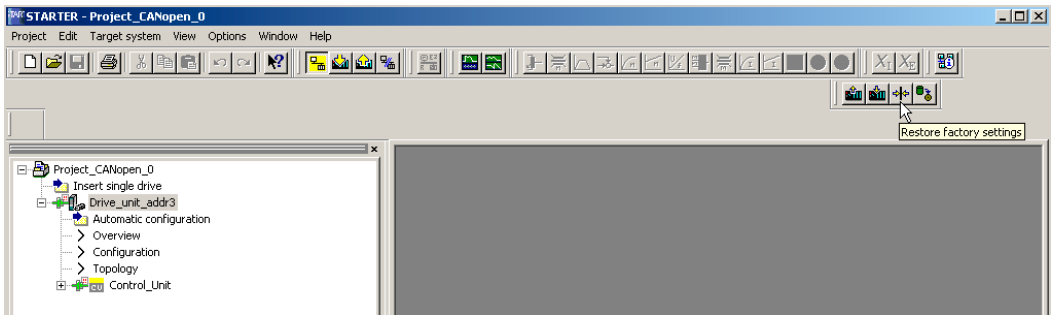


Figure 3-5 Restore factory settings

6. Confirm the following queries and messages by choosing **OK**:
   - "Restore factory settings?" dialog box
   - "The factory settings have been restored" dialog box
   - "The data has been successfully copied from RAM to ROM" dialog box

7. In the project navigator, double-click **Automatic configuration** under the drive unit.

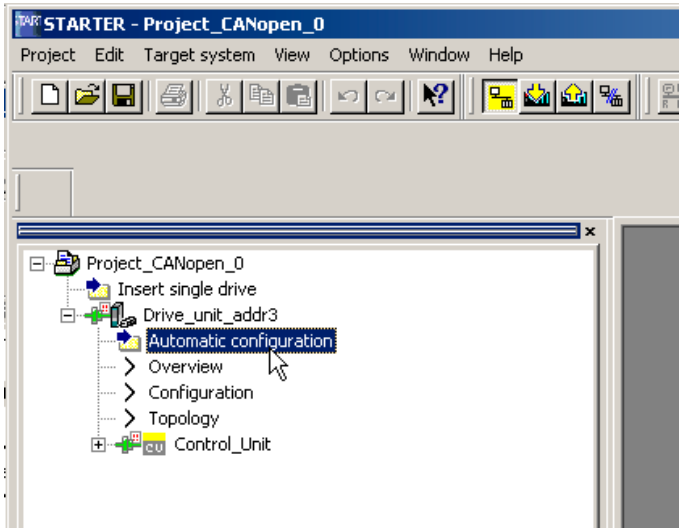


Figure 3-6 Automatic configuration

8. In the "Automatic configuration" dialog box, click **Start automatic configuration**. STARTER 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**.

Note

When selecting **Vector**, (not for SINAMICS S110) CANopen is supported, just like for servo, using the velocity mode profile and free process data.

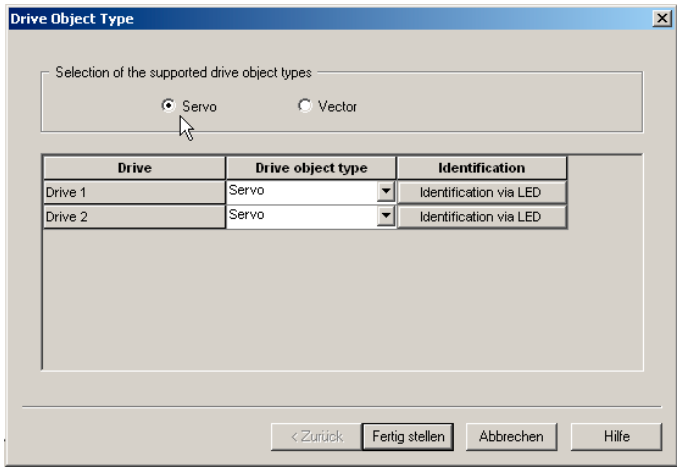


Figure 3-7 Drive object type

10. Click **Fertig stellen (Complete)**. The system loads the data from RAM to ROM and to the PG.

11. A message is output to inform you that the motors also have to be configured. The motor configuring is described in the following Chapter "Configuring motors". Confirm this message with **OK**.
12. Initialization is complete: Click **Close** in the "Automatic configuration" dialog box.

Automatic configuration is complete

Automatic configuration is complete. In the project navigator, all the drive unit components that have been found (e.g. **Control Unit, CBC10, infeed**, and **drive**) are displayed.

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

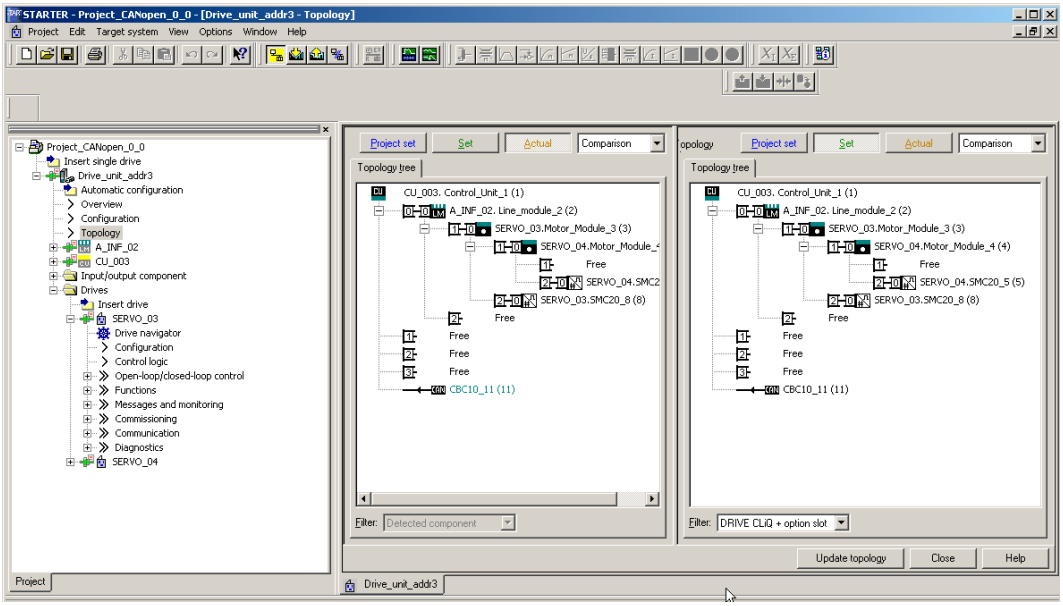


Figure 3-8 Target and actual topology drive configuration (example)

Note

For SINAMICS S110:

Only the components available are displayed for SINAMICS S110; e.g. there is no CBC10 Communication Board.

You will now be shown how to configure the motor OFFLINE in STARTER.

3.4.3 Configuring the motor

Introduction

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

Steps

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

Note

You only have to change the motor and encoder configuration; leave the infeed and so on as it is.

1. Choose **Disconnect from target system...**. The modified data is loaded from RAM to ROM and to the PG.
   The motors are configured in OFFLINE mode and are then loaded to the target system in ONLINE mode.
2. In the project navigator, choose the Drives folder and double-click **Configuration** under the relevant drive (see example screenshot below). **Note:** SINAMICS S110 only supports one drive.

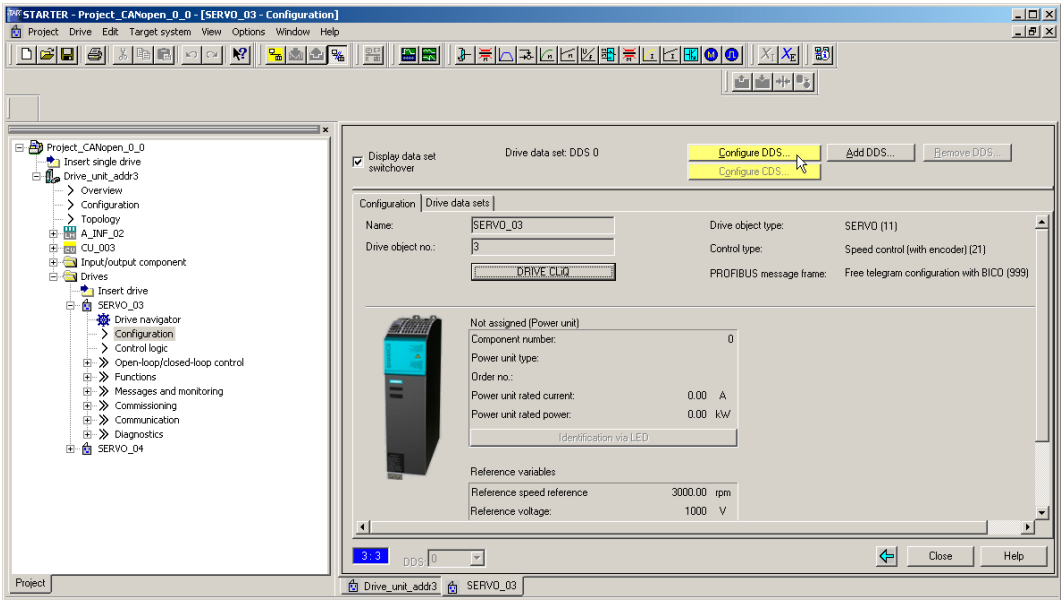


Figure 3-9 Configure the drive (parameterize the motor)

3. In the dialog box that is displayed, choose **Configure DDS...**
4. In the displayed dialog box "Configuration - Drive unit_addr_126 - Control structure", you can define whether or not the drive object (function module) is operated with or without an 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 clicked out.

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 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 and 6083 hex of the drive provide are included.

5. You only configure the motor and the encoder! Work through the Wizard by choosing **Continue >** until you reach the point at which you configure the motor (refer to the following diagram).

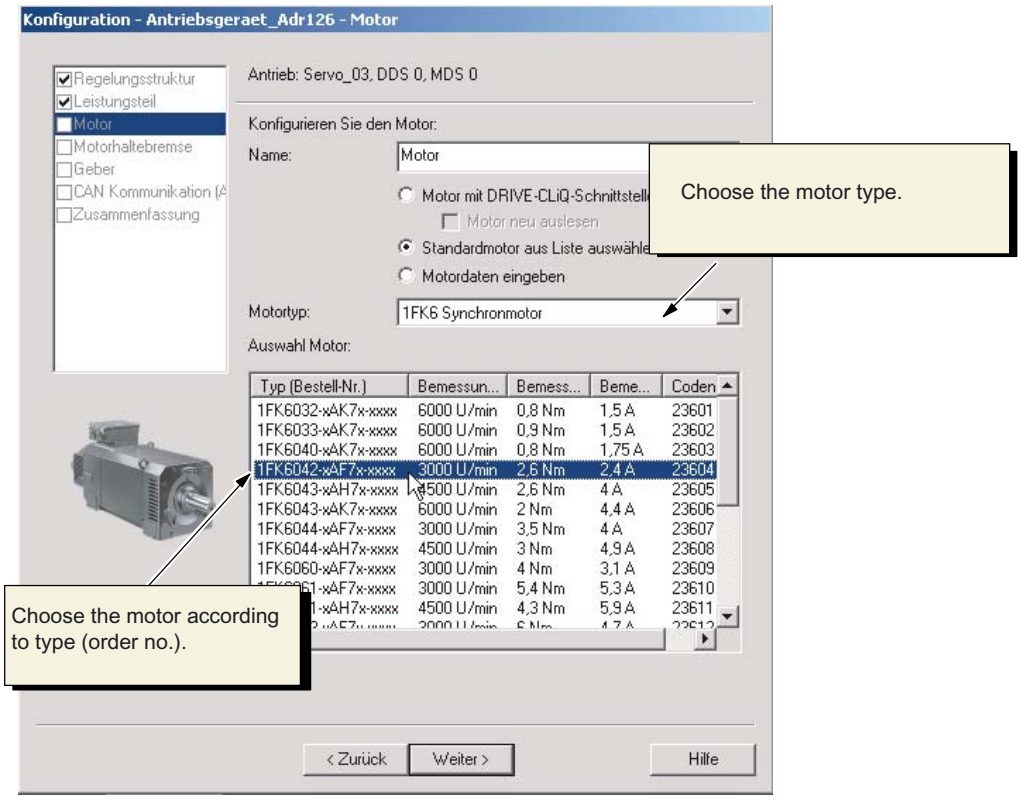


Figure 3-10 Configure the motor

6. Choose the motor type and the motor according to the type (order no.) (see the type plate).
7. Click **Continue >** until you reach the point at which you configure the encoder.
8. Choose the motor encoder and work through the Wizard by choosing **Continue >** until the dialog box containing a summary is displayed.
9. Click **Fertig stellen (Complete)**.
10. Once you have configured the motor and the encoder for the first drive, repeat these steps for the second drive. **Note:** SINAMICS S110 only supports one drive.

When you have completed configuring the drive unit OFFLINE, carry out the following steps to parameterize the CANopen interface on the Control Unit.

3.4.4 Configuring the interface for the CBC10 Option Board on the Control Unit

Prerequisite

You have configured the drive unit with the CBC10 Option Board in STARTER and connected the drive OFFLINE with STARTER.

In the project navigator, double-click **Drive unit_Adr_126 > Control_Unit > CAN Option Board > Configuration**.

You can now configure the CBC10 Option Board in the dialog box that is displayed.

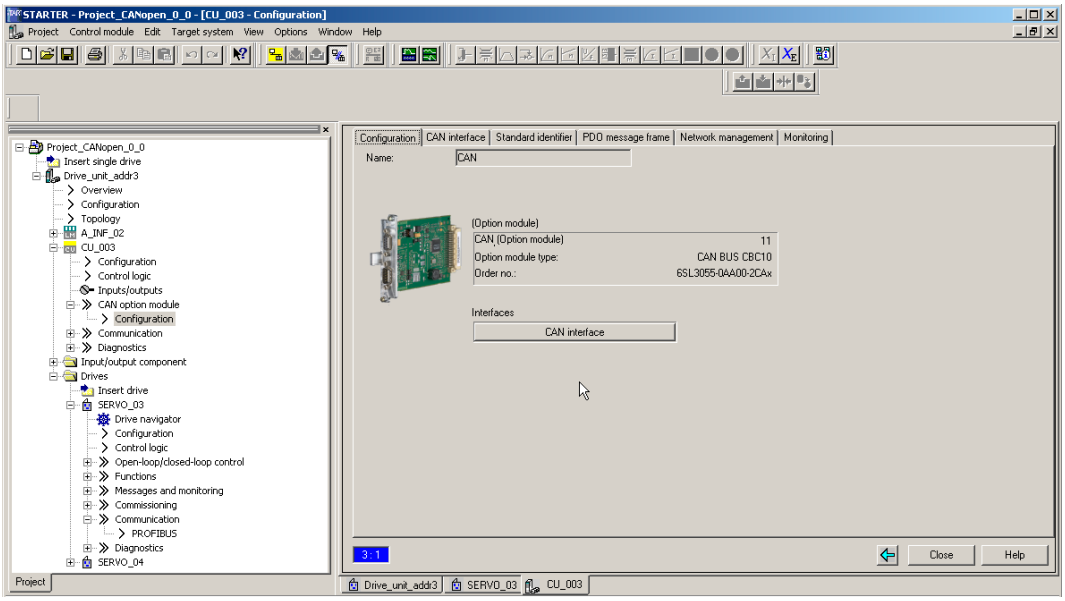


Figure 3-11 Configuring the CBC10 Option Board

Carrying out the commissioning step

On the Control Unit CU320, configure the following transmission properties for the CBC10 Option Board:

- Transmission rate
- CAN bus address (node ID)
- Number of PDO message frames
- Node monitoring (heartbeat, node guarding)

3.4.4.1 CAN interface

Steps

On the **CAN interface** tab, enter the transmission rate and the CAN bus address/node ID.

1. Select the **CAN interface** tab (see screenshot below).

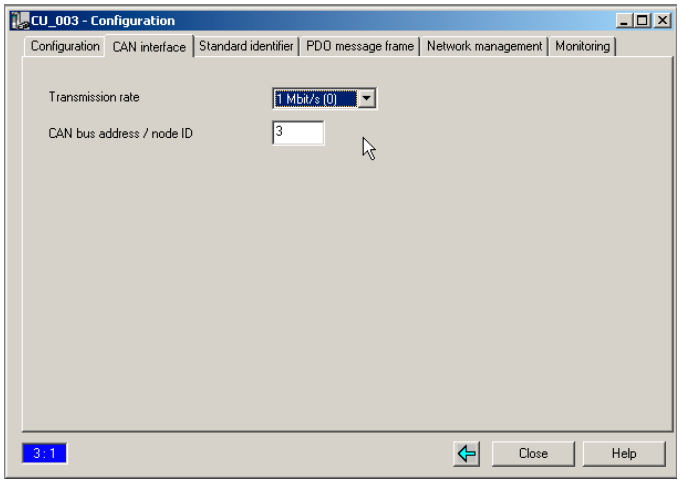


Figure 3-12 CAN interface

2. Enter a transmission rate of **1 MBit/s** for commissioning, for example.
   The factory setting is 20 kBit/s.

Note

If, during commissioning, you power down/power-up the control or carry out a RESET, then the factory settings are restored.

---

3. Two options are available for the bus address/node ID:
   - In this dialog box, you can set a value of between 1 and 126 if the address switch on the Control Unit ("DP address") is set to 0 or 127.
   - Directly using the address switch on the Control Unit.
     The following diagram shows an example for address **5**.

Example

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

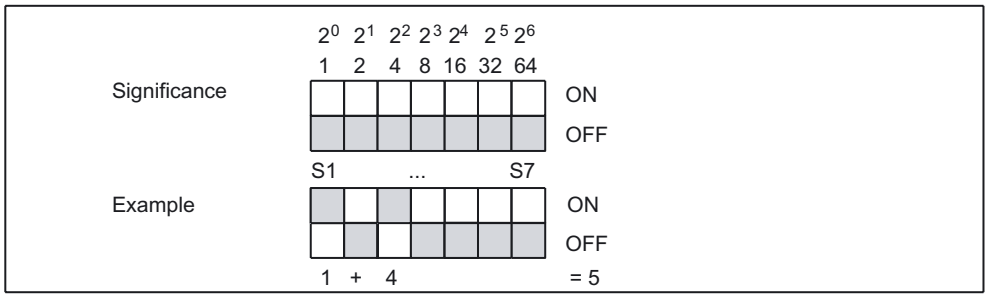


Figure 3-13 Example: Bus address via the address switch on the Control Unit

The following information must be taken into account!

Note

Permissible CAN bus address: 1...126. 1...126.

The address setting on the switch is displayed in **p8620.0**.

If the switch setting is changed, the new setting does not become effective until POWER ON.

The factory setting is "ON" or "OFF" for all switches.

While the SINAMICS is being booted, the address switch is queried first in order to set the bus address. If the switch setting is 0 or 127, the address can be set via parameter **p8620.0**.

If the address is set to a valid node address (1...126), this is copied to parameter **p8620.0** and displayed.

3.4.4.2 PDO message frame

Introduction

This description of the initial commissioning procedure uses a predefined setting ("predefined connection set") with a fixed assignment of SDOs and PDOs – whereby the device features the following for each SINAMICS drive object (in this case, the Single Line Module):

- 4 receive PDOs (channels) and
- 4 transmit PDOs (channels).
- An SDO pair for reading and writing the objects (acyclic configuration) is available with a "predefined connection set" and for free PDO mapping.

4 receive and 4 transmit channels are set as default in the CU320 for each Motor Module. A maximum of 8 transmit and receive channels can be set.

Note

CU305:

Precisely 8 transmit and 8 receive PDOs are assigned to the CU305.

Steps

In the **PDO telegram**, enter the number of transmit and receive PDO telegrams (channels) for each drive object.

1. Select the **PDO telegram** tab (see screenshot below).

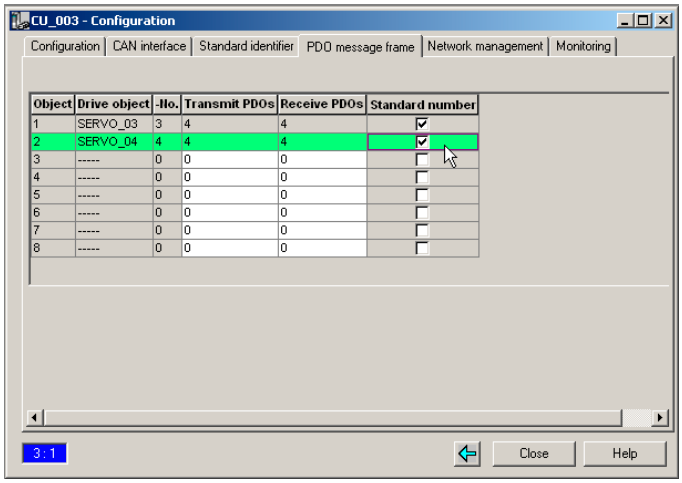


Figure 3-14 Standard number

2. Unless already preset, enter **4** transmit PDOs and **4** receive PDOs for both **Drive_1** and **Drive_2**.

Note

The standard number (predefined connection set) is **4**. You can also enter this value for each drive via the **Standard number** field.

Commissioning in ONLINE mode

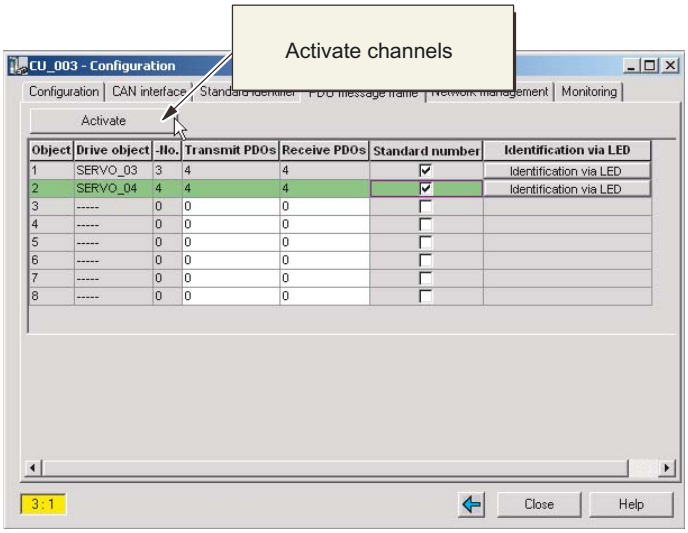


Figure 3-15 Activating the channels

Note

Once you have entered the channel assignments in ONLINE mode, confirm them in this dialog box by choosing **Activate** (refer to diagram above).

You can also confirm them in the expert list in parameter **p8741** by choosing **1** (see screenshot below).

Operator steps, activate the channel distribution in the OFFLINE mode

1. In the project navigator, open the expert list for the Control Unit as follows: **Control Unit > right-click > Expert > Expert list**.
2. Search for parameter **p8741** for confirming the channel assignment and enter **1**.
3. In parameter **p8742**, you can see how many channels are still available (max. 25).

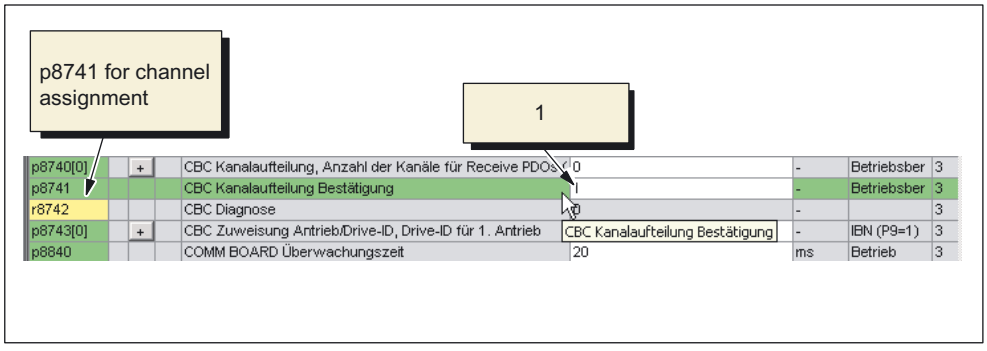


Figure 3-16 Confirm channel assignment

3.4.4.3 Monitoring

Introduction

SINAMICS supports the following two 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.
- Node 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.

Note

Only one node monitoring service can be activated at any one time (either heartbeat or node guarding).

If both monitoring services are activated, node guarding is effective.

Steps

On the **Monitoring** tab, enter the required monitoring service (heartbeat or node guarding).

1. Select the **Monitoring** tab.
2. The default commissioning value for the **Heartbeat** monitoring mechanism can be specified as **100 ms** . Enter this value (unless it has already been entered).
3. The default commissioning values for the **node guarding** monitoring mechanism could be
   – Time interval (guard time) **100 ms,**
   – Number of failures (lifetime factor) **3.**
   Enter these values (unless they have already been entered).

The CANopen interface is now parameterized. To load the project to the target system in ONLINE mode, carry out the following steps.

In the case of an error in the CAN communication, e.g. too many message frame failures, the fault **F08700** is reported (for details see List Manual S120). 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** govern the behavior of the drive/CAN node in the event of a CAN communication error or device malfunction.

The factory settings for the parameters are

p8609 = 1, => no change

p8641 = 0, => 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 response (Factory setting)
  - 1: OFF1
  - 2: OFF2
  - 3: OFF3

3.4.5 Activating the "predefined connection set"

Introduction

Note

The STARTER commissioning tool is in OFFLINE mode.

Activating the "predefined connection set"

To activate the "predefined connection set", carry out the following steps:

1. In the project navigator, open the expert list for the drive as follows: **Drive_1 > right-click > Expert > Expert list**.

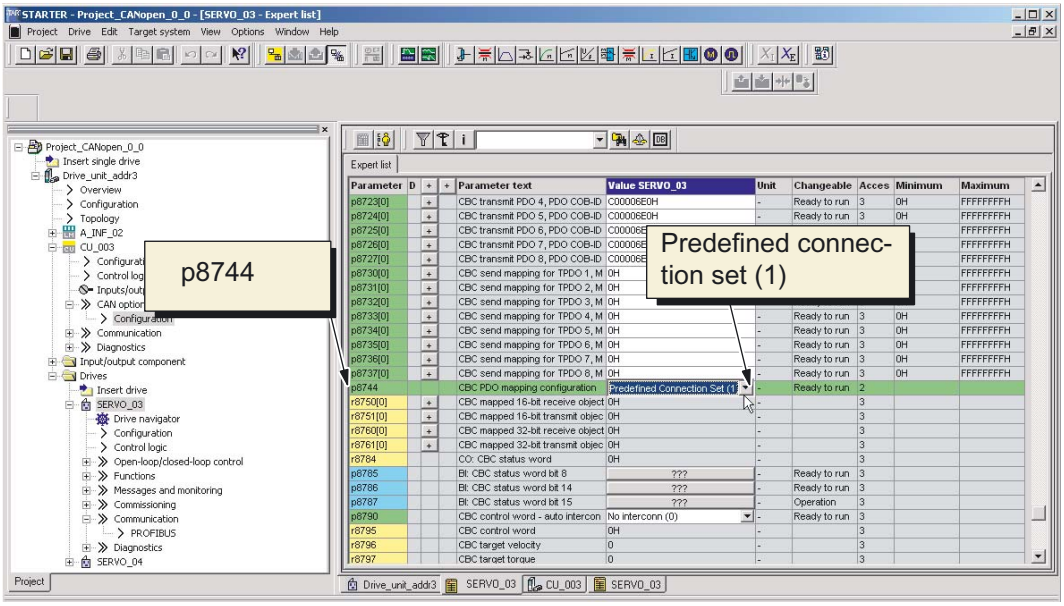


Figure 3-17 Predefined connection set

2. Find parameter **p8744** (see screenshot above).
3. Choose "Predefined connection set (1)" to set the "predefined connection set".
4. Repeat steps 1 to 3 for the next SINAMICS drive object, **Drive_2**. **Note:** SINAMICS S110 only supports one drive.

3.4.6 Loading the project to the drive unit

Introduction

To load the project to the drive unit, proceed as follows:

Steps

1. Click **Connect to target system**. An ONLINE connection is established and an ONLINE/OFFLINE comparison takes place. If any discrepancies are identified, they are displayed (see screenshot below).

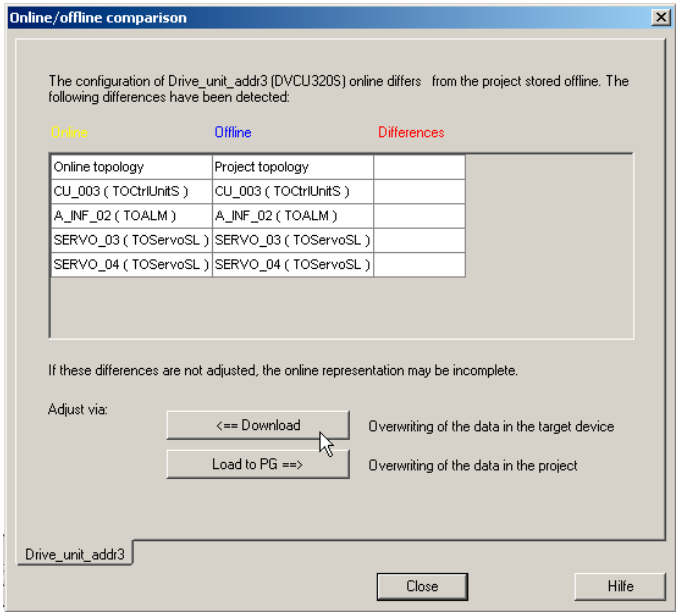


Figure 3-18 ONLINE/OFFLINE comparison (example)

2. You changed the data OFFLINE and now have to load it to the target system. Carry out the following:
   - **<-- Download** in the "ONLINE/OFFLINE comparison" dialog box
   - When the system asks "Are you sure?", click **Yes**. The system now starts loading the data.
   - When the system informs you that the data was successfully loaded to the target system, click **OK**.
   - Click **OK** for "Load from RAM to ROM".

3. Discrepancies were identified again during the ONLINE/OFFLINE comparison. Now click **Load to PG -->** (see screenshot below).

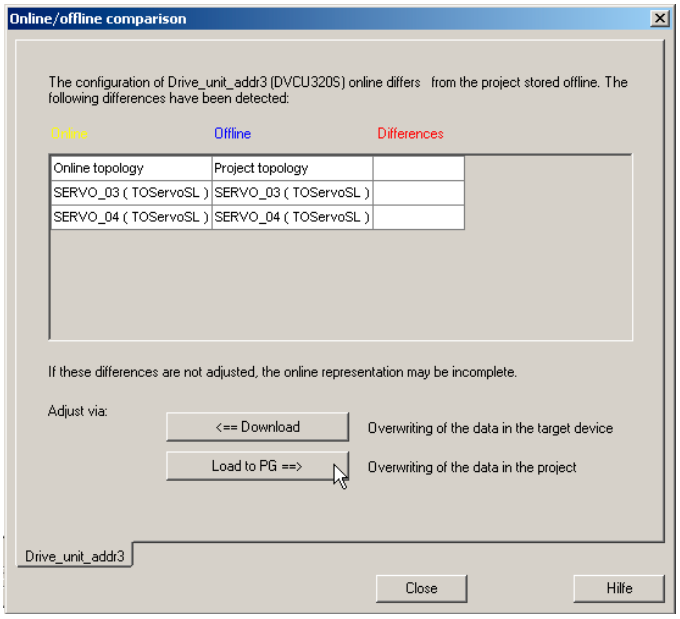


Figure 3-19 Load to PG

4. Load the new data from the drive unit to the PG. Carry out the following:
   - When the system asks "Are you sure?", click **Yes**. The system now starts loading the data.
   - When the system informs you that the data was successfully loaded to the PG, click **OK**.
5. No further discrepancies are displayed in the ONLINE/OFFLINE comparison dialog box. Click **Close** (see screenshot below).

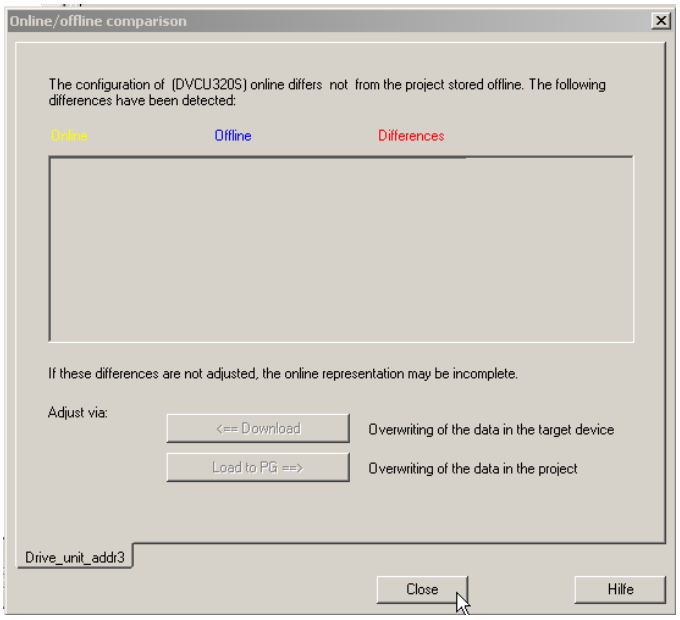


Figure 3-20 ONLINE/OFFLINE comparison OK

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.

3.4.7 Interconnecting the infeed

Carrying out the commissioning step

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

The DC link can still be activated 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 BICO interconnection procedure in STARTER is described in the section "BICO interconnection procedure in STARTER".

Steps

To interconnect parameter **p0840** (ON/OFF1) with the signal source (e.g. from the control word of Motor Module 1 (Drive_1)) from the infeed, carry out the following:

1. In the project navigator, choose parameter **p0840** ("search"): **Infeed > right-click > Expert > Expert list**.
2. Interconnect the p parameter with r parameter **r8890 bit 15** = PZD 1 from Drive_1 (SERVO_3).

Note

Please also refer to the commissioning instructions in the Commissioning Manual SINAMICS S120 /IH1/

e.g. the DC link must be activated before the Motor Modules are switched on.

3.5 Configure the COB IDs and process data objects for the receive and transmit message frames.

Initial commissioning: procedure

In the table below, the current commissioning step is highlighted in **bold**:

Table 3- 5 CANopen initial commissioning

Step	Procedure
1	Hardware settings on CBC10 (not for SINAMICS S110)
2	Configure the drive unit using the STARTER commissioning tool in 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 ONLINE mode, load the projects from the drive unit to the PC/PG and save.

Carrying out the commissioning step

You need to commission the communication and mapping parameters for the individual drive objects. STARTER 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:

- COB IDs
- Mapping the process data objects

Note

As already described in the introduction, the communication parameters, mapping parameters, and COB IDs are 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 the receive and transmit buffers".

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

Introduction

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

1. In the project navigator, call up the expert list for drive object 1: **Drives > Drive_1 > right-click > Expert > Expert list**.
2. In the project navigator, call up the expert list for drive object 2: **Drives > Drive_2 > right-click > Expert > Expert list**.

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

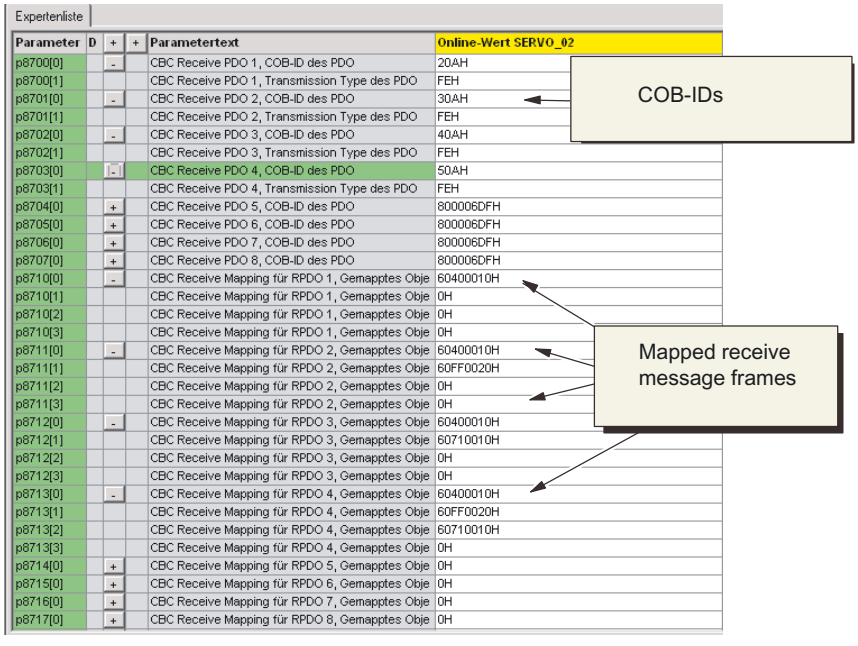
Note

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

For each drive object in the expert list, the process data objects for mapping the message frames begin as of parameter **p8710** for receive message frames and as of 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 STARTER.



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 3-21 COB IDs and mapped process data objects for receive message frame drive object 1

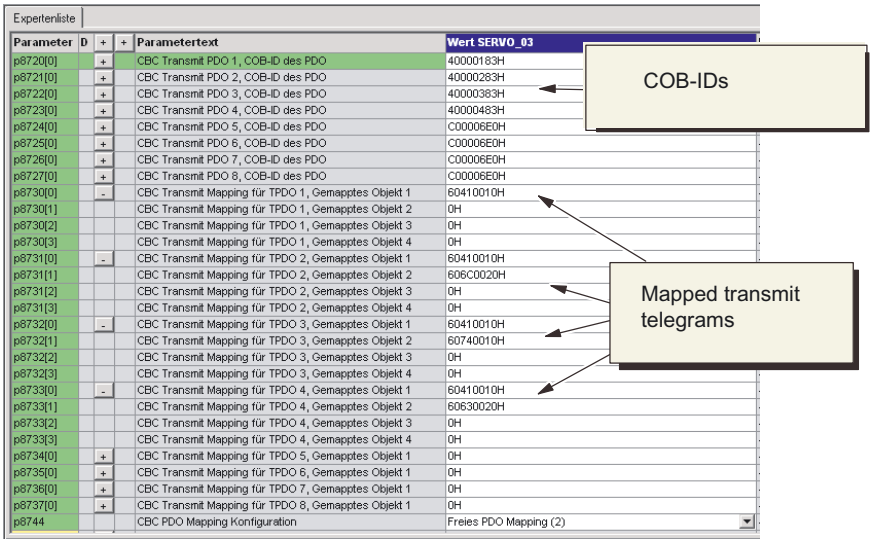


Figure 3-22 COB IDs and mapped process data objects for transmit message frame drive object 1

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 STARTER. The process data objects are parameterized with an offset of 800 hex for the second drive object.

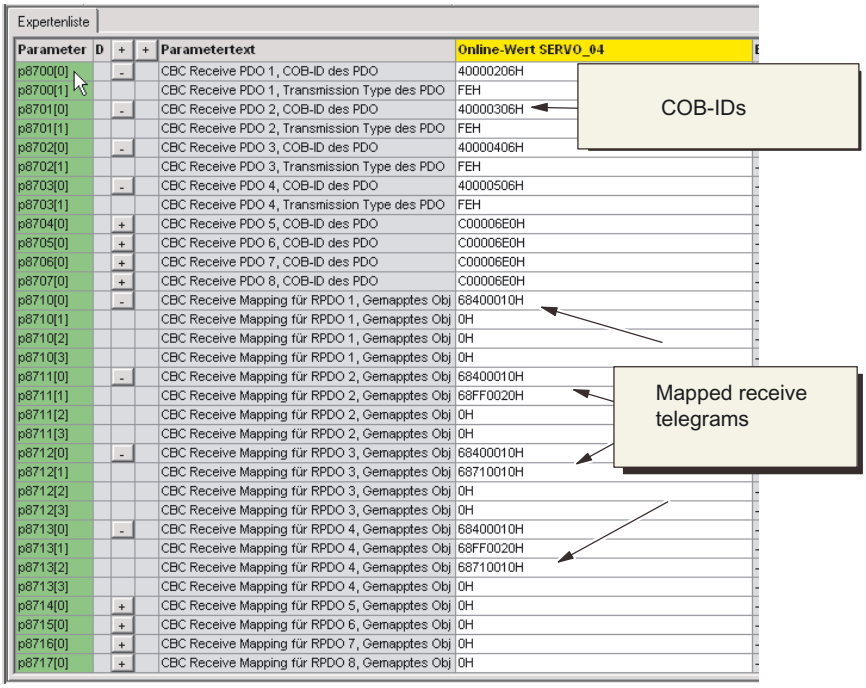


Figure 3-23 COB IDs and mapped process data objects for receive message frames drive object 2

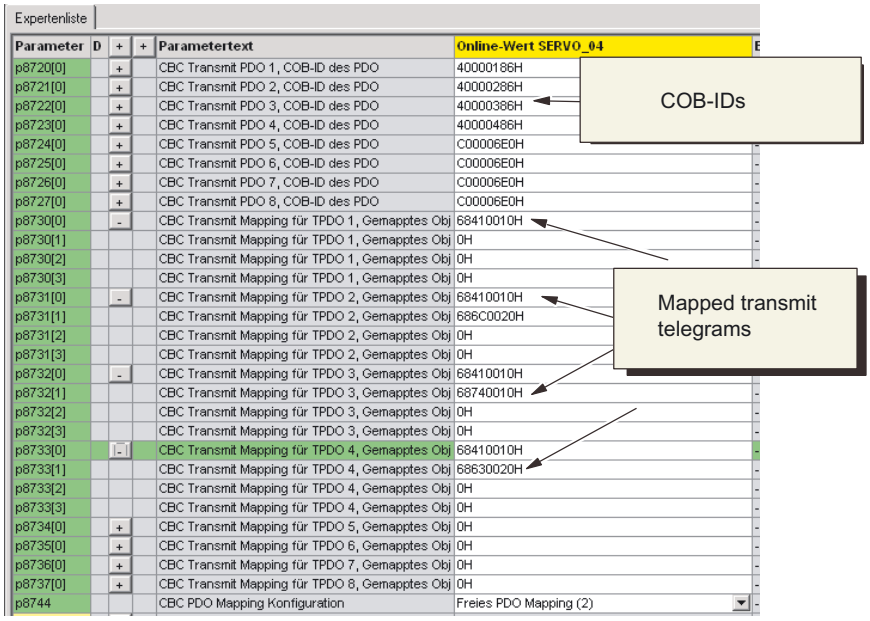


Figure 3-24 COB IDs and mapped process data objects for transmit message frames drive object 2

3.5.2 Activating "free PDO mapping"

Steps

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

1. In the project navigator, open the expert list for the drive as follows: **Drive_2 > right-click > Expert > Expert list**.
2. Search for parameter **p8744** (see screenshot below) to activate free PDO mapping.

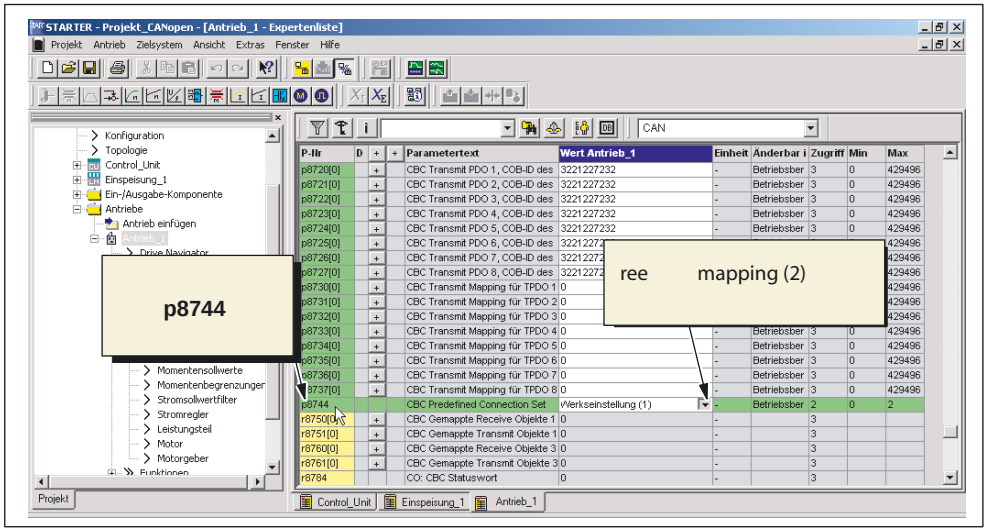


Figure 3-25 Free PDO mapping

3. Choose **Free PDO mapping (2)**.

3.5.3 Assigning COB IDs and mapping parameters for free PDO mapping

Introduction

If you are in 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) as of parameter **p8710**
  - Transmit message frames (TPDO) as of parameter **p8730**

Note

If mapping parameters are changed in STARTER 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 ONLINE mode, proceed as follows:

1. Set the COB ID of the RPDO or TPDO in question to "invalid" (e.g.: **p8700[0]** CBC: receive PDO1, COB ID of PDO).
2. Enter the process data objects as mapping parameters into the relevant RPDO or TPDO (for e.g.: from **p8710[0]** CBC ReceiveMapping for RPDO1, mapped object).
3. Set the COB ID of the RPDO or TPDO in question to "valid".

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. In the project navigator, call up the expert list for Single Motor Module 2 (drive_2): **Drives > Drive_2 > right-click > Expert > Expert list**.
2. Search for COB ID parameter **p8700 [0]** for communication parameter RPDO1.

3. Copy the hexadecimal value from STARTER to a calculator and enter **Or** 8000 0000 hex. Copy the result to the STARTER parameter. As a result, bit 31 has now been set to "invalid".
4. You can now adjust the mapping parameters.

Mapping parameters

1. Choose the process data object to be mapped
   (e.g. RPDO1, control word = 6040 hex) from the "Objects in drive profile DSP402" table (see "Communication objects") .
2. Add a suitable offset for the SINAMICS drive object (e.g. plus 800 hex as of 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 STARTER in the corresponding mapping parameter.
   See the column "Mapping parameter in STARTER" (e.g. **p8710 [0]**) in the parameterization table in the following section.

Setting the COB ID to "valid"

1. In the Project Navigator, select e.g. **Drives > Drive_2 > righthand mouse key > Expert > Expert list** the expert list for the Single Motor Module 2.
2. Search for COB ID parameter **p8700 [0]** for communication parameter RPDO1.
3. Copy the hexadecimal value from STARTER to a calculator and enter **And** 7FFF FFFF hex. Copy the result to the STARTER parameter. As a result, bit 31 has now been set to "valid".
4. Copy the value to STARTER.

Note

The RPDOs and TPDOs described below serve to illustrate the free PDO mapping process. You can decide for yourself how you want to map your own process data objects for transmission message frames.

See also

Objects in drive profile DSP402 (Page 146)

Mapping RPDO1

When you choose the process data object for the control word from the "Objects in 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- 6 Values for RPDO1

Object name	Values from "Objects in drive profile DSP402" table			Mapping parameter in STARTER	
	OD index (hex)	Sub-index (hex)	Object size	Resulting hexa-decimal value	Parameter RPDO1
Control word	6840	00	10 hex (16 bit)	6840 0010	p8710 [0]
No object				0	p8710 [1] to [3]

Mapping RPDO2

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

Table 3- 7 Values for RPDO2

Object name	Values from "Objects in drive profile DSP402" table			Mapping parameter in STARTER	
	OD index (hex)	Sub-index (hex)	Object size	Resulting hexa-decimal value	Parameter RPDO2
Control word	6840	00	10 hex (16 bit)	6840 0010	p8711 [0]
Set velocity	68FF	00	20 hex (32 bit)	68FF 0020	p8711 [1]
No object				0	p8711 [2] to [3]

Mapping RPDO3

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

Table 3- 8 Values for RPDO3

Object name	Values from "Objects in drive profile DSP402" table			Mapping parameter in STARTER	
	OD index (hex)	Sub-index (hex)	Object size	Resulting hexa-decimal value	Parameter RPDO3
Control word	6840	00	10 hex (16 bit)	6840 0010	p8712 [0]
Set torque	6871	00	10 hex (16 bit)	6871 0010	p8712 [1]
No object				0	p8712 [2] to [3]

Mapping RPDO4

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

Table 3- 9 Values for RPDO4

Object name	Values from "Objects in drive profile DSP402" table			Mapping parameter in STARTER	
	OD index (hex)	Sub-index (hex)	Object size	Resulting hexa-decimal value	Parameter RPDO4
Control word	6840	00	10 hex (16 bit)	6840 0010	p8713 [0]
Set velocity	68FF	00	20 hex (32 bit)	6840 0020	p8713 [1]
Set 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 in drive profile DSP402" table, this yields the following values, which you enter in parameters **p8730 [0]** to **p8730 [3]** for TPDO1 mapping (see the "Resulting hexadecimal value" column in the following table):

Table 3- 10 Values for TPDO1

Object name	Values from "Objects in drive profile DSP402" table			Mapping parameter in STARTER	
	OD index (hex)	Sub-index (hex)	Object size	Resulting hexa-decimal value	Parameter TPDO1
CBC status word	6841	00	10 hex (16 bit)	6841 0010	p8730 [0]
No object				0	P8730 [1] to [3]

Mapping TPDO2

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

Table 3- 11 Values for TPDO2

Object name	Values from "Objects in drive profile DSP402" table			Mapping parameter in STARTER	
	OD index (hex)	Sub-index (hex)	Object size	Resulting hexa-decimal 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] to [3]

Mapping TPDO3

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

Table 3- 12 Values for TPDO3

Object name	Values from "Objects in drive profile DSP402" table			Mapping parameter in STARTER	
	OD index (hex)	Sub-index (hex)	Object size	Resulting hexa-decimal 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] to [3]

Mapping TPDO4

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

Table 3- 13 Values for TPDO4

Object name	Values from "Objects in drive profile DSP402" table			Mapping parameter in STARTER	
	OD index (hex)	Sub-index (hex)	Object size	Resulting hexa-decimal 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] to [3]

3.5.4 Interconnecting the receive and transmit buffers

Steps

Once you have mapped receive and transmit PDOs in ONLINE mode in free PDO mapping, you have to interconnect the image of the mapped process data objects with the receive and transmit buffers.

The receive and transmit buffer is updated ONLINE with objects from validated PDOs. Objects from non-validated PDOs are not taken into account in the process data buffers.

Note

To interconnect the receive and transmit buffers in ONLINE mode, carry out the steps described in the following section ("Interconnecting receive and transmit buffers").

3.6 Interconnecting process data in the receive and transmit buffers

Initial commissioning: procedure

In the table below, the current commissioning step is highlighted in **bold**:

Table 3- 14 CANopen initial commissioning

Step	Procedure
1	Hardware settings on CBC10 (not for SINAMICS S110)
2	Configure the drive unit using the STARTER commissioning tool in 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 ONLINE mode, load the projects from the drive unit to the PC/PG and save.

Carrying out the commissioning step

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:

- Read the image of the process data objects to the PZD receive and PZD transmit words for the receive and transmit buffers.
- Interconnect the source parameter of the PZD receive word on the receive buffer with the SINAMICS target parameter of the process data object.
- Interconnect the SINAMICS source parameter of the process data object with the target parameter of the PZD transmit word on the transmit buffer.

Note

The STARTER commissioning tool is in ONLINE mode.

See also

BICO interconnection procedure in STARTER (Page 67)

Objects in drive profile DSP402 (Page 146)

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

Introduction

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

To read 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:

- 16-bit process data objects for
  - Receive, as of parameter **r8750[0]**
  - Transmit, as of parameter **r8751[0]**
- 32-bit process data objects for
  - Receive, as of parameter **r8760[0]**
  - Transmit, as of parameter **r8761[0]**

Steps (using drive object 2 as an example)

1. In the project navigator, call up the expert list for drive object 2: **Drives > Drive_2 > right-click > Expert > Expert list**.
2. Search for r 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
   - Set torque (6871 hex) in PZD 4

3. Search for r parameter **r8760[0]** with the image of the 32-bit process data objects in the receive buffer. In this example, the following is mapped (see also screenshot below):
   – Set velocity (68FF hex) in PZD 2+3

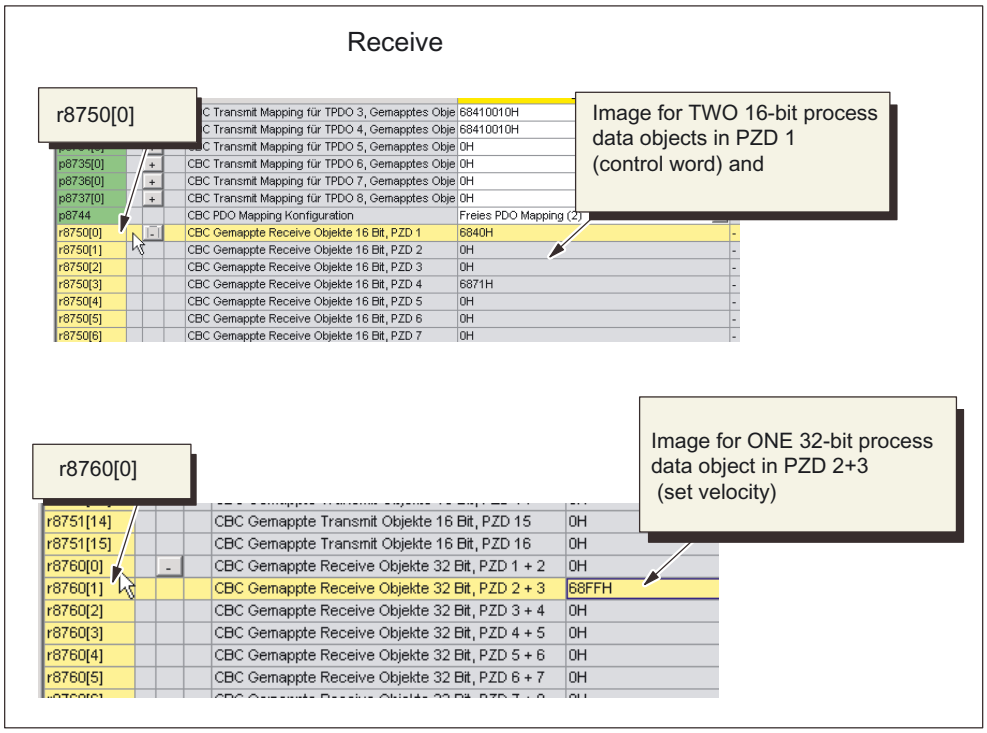


Figure 3-26 Image of process data objects in receive buffer

4. Search for r 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
5. Search for r 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

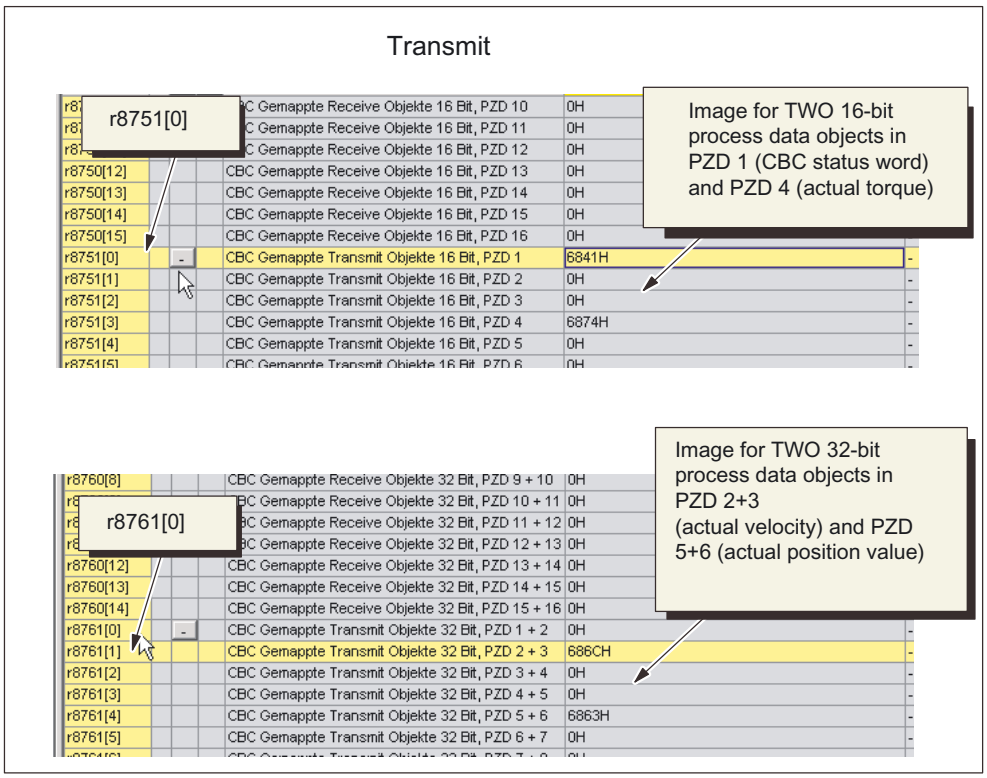


Figure 3-27 Image of process data objects in transmit buffer

6. In the project navigator, call up the expert list for drive object 1: **Drives > Drive_1 > right-click > Expert > Expert list**.
   For receiving purposes, the 16-bit process data objects as of parameter **r8750[0]** and the 32-bit process data objects as of parameter **r8760[0]** are also available here.
   Likewise, the 16-bit process data objects as of parameter **r8751[0]** and the 32-bit process data objects as of parameter **r8761[0]** can be read for transmission purposes.
7. 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

Prerequisite

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

The following are interconnected:

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

Note

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**).

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 in drive profile DSP 402" table.

See also

Objects in drive profile DSP402 (Page 146)

3.6.2.1 Interconnecting the receive buffer

Introduction

In the receive buffer, the following are interconnected for the transmission message frames:

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

Steps

If you for example: are interconnecting the set 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 set velocity (**p1155[0] =>** 32 bit, see "Objects in drive profile DSP402" table)

Table 3- 15 Excerpt of "Objects in drive profile DSP402"

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Pre-set values:	Read/ write
...	...	...	...	...	...	...	...
60FF		Target velocity Set 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 set velocity is **highlighted** accordingly).

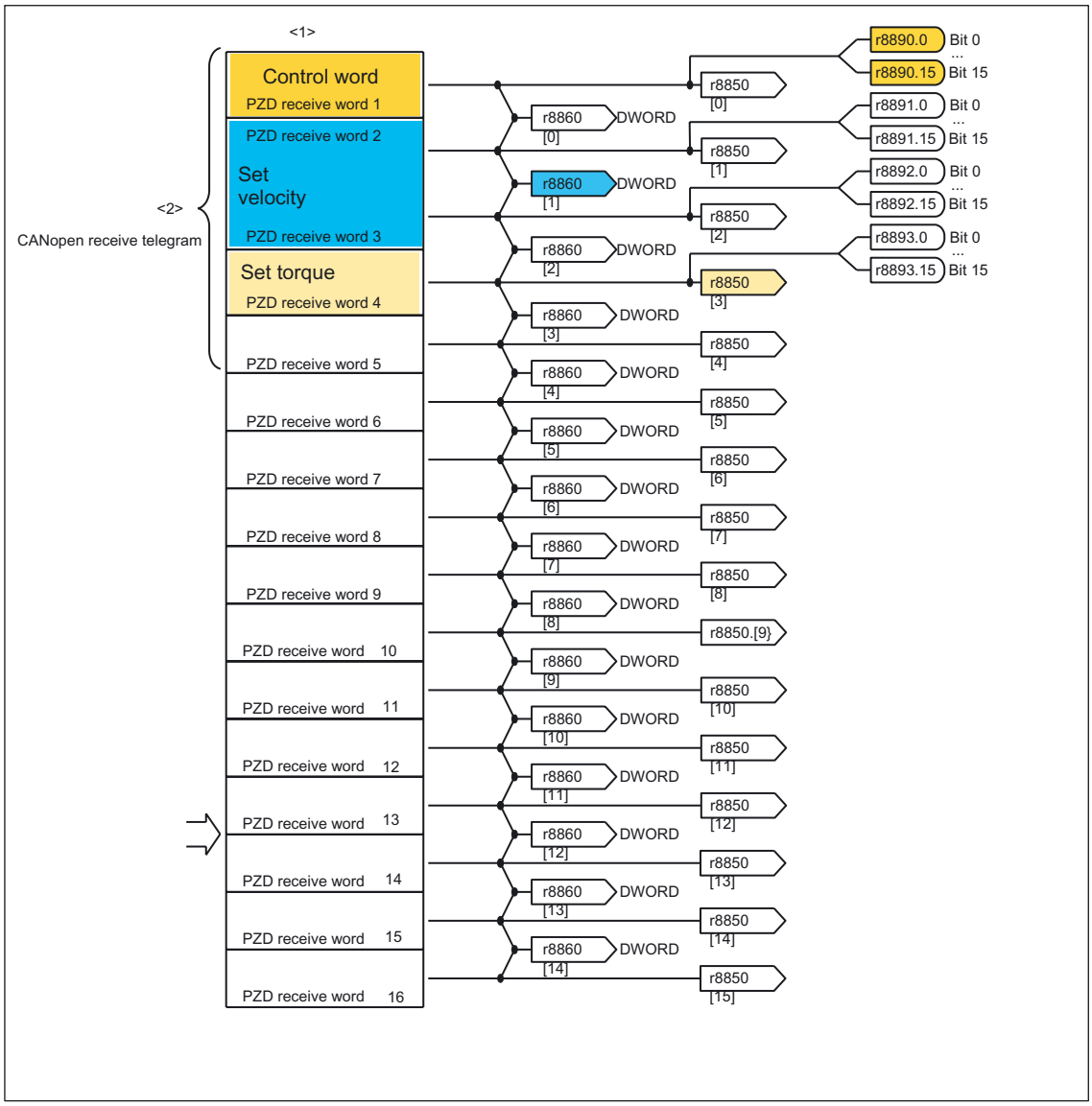


Figure 3-28 Excerpt of function diagram: receive buffer

Note

You can now interconnect the process data objects listed below.

Control word (PZD 1 16 bit)

When commissioning the system for the first time, you automatically interconnect the control word as SINAMICS target parameter to the source parameter **r8890** by "searching" for parameter **p8790** for the particular drive e.g. **Drive_1 > right click > Expert > Expert list** and

then activating the interconnection (1). After you exit this field, then the interconnection is made.

Set velocity (PZD 2+3 32 bit)

The following table shows the p parameter for the set velocity, which has to be interconnected with the corresponding source.

Table 3- 16 Interconnect set velocity

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

Interconnect p parameter **p1155 [0]** with r parameter **r8860 [1]** for the target velocity (32-bit data type) by carrying out the following:

1. In the project navigator, choose **Drive_1 > Open-loop/closed-loop control** and double-click **Setpoint addition**.
2. Choose **Speed setpoint1 > Further interconnections**.
3. Interconnect parameter **p1155 [0]** with **r8860 [1]** = PZD 2+3.

Set torque (PZD 4 16 bit)

The following table shows the p parameter for the set torque, which has to be interconnected with the corresponding source.

Table 3- 17 Interconnect the set torque

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

Interconnect p parameter **p1513 [0]** with r parameter **r8850 [3]** for the set torque by carrying out the following:

1. In the Project Navigator under **Drive_1 > Open-loop/closed-loop control** double-click on **torque setpoints**.
2. Choose **Supplementary torque 2 > Further interconnections**.
3. Interconnect parameter **p1513 [0]** with **r8850 [3]** = PZD 4.

Note

To ensure that the set torque becomes active, it still has to be interconnected with a free bit of the control word. This can be carried out in this dialog box by choosing **Speed/torque control > Further interconnections** (**p1501 [0]** with **bit 14** from **r8890**).

3.6.2.2 Interconnecting the transmit buffer

Introduction

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)

Steps

If you for example: are interconnecting the CBC status word process data object in the 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 "Objects in drive profile DSP402" table)

Table 3- 18 Excerpt of "Objects in drive profile DSP402"

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Default values	Read/ write
6041		Status word	r8784	PDO/SDO	Unsigned16	-	ro

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

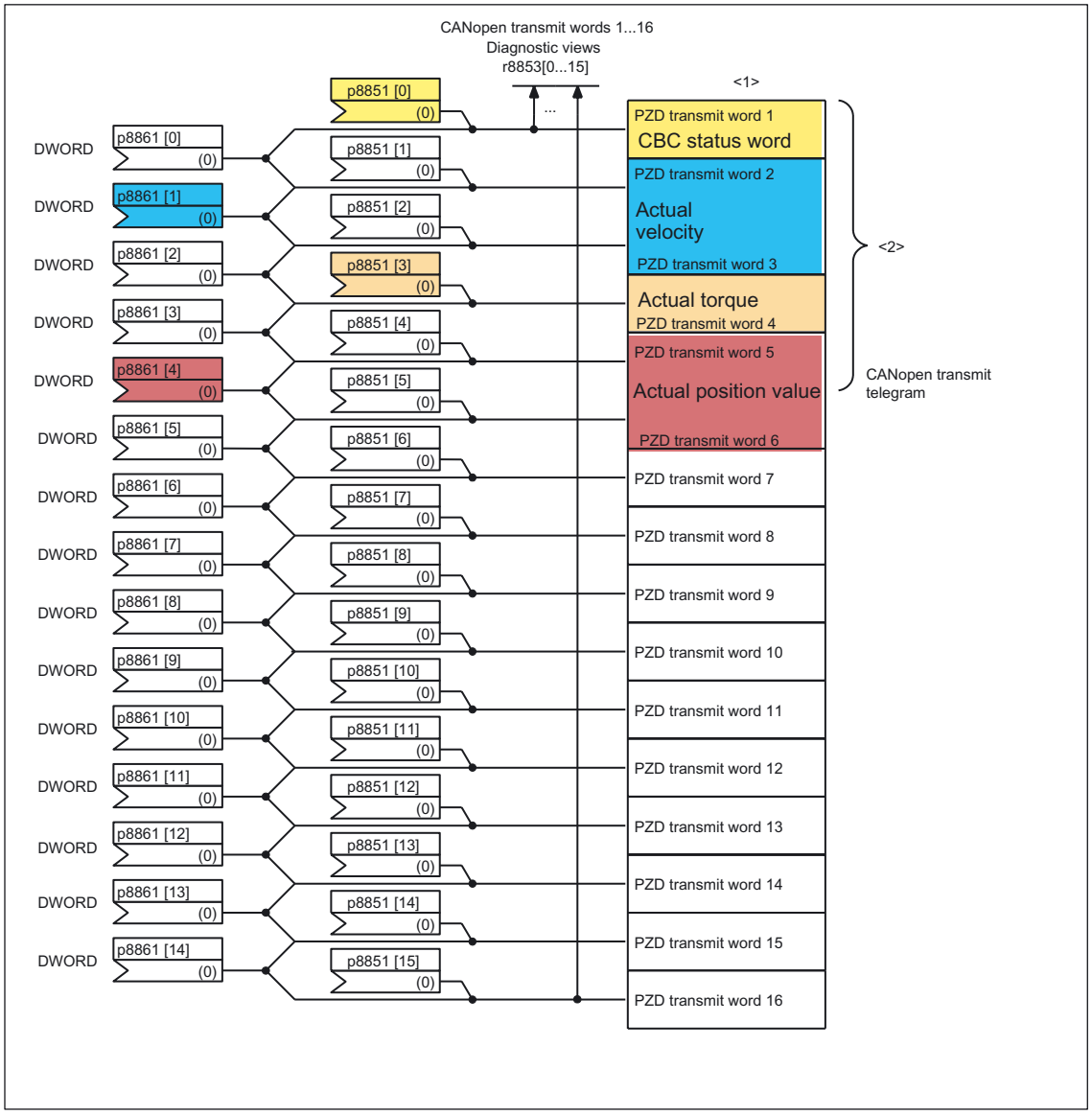


Figure 3-29 Excerpt of function diagram: transmit buffer

Note

You can now interconnect the process data objects listed below.

CBC status word (PZD1 16 bit)

The following table shows the p parameter for the CBC status word, which has to be interconnected with the corresponding source.

Table 3- 19 Interconnect the CBC status word

Target (sink) PZD1	Source	Meaning
p8851 [0]	r8784	CBC status word

Interconnect p parameter **p8851 [0]** with r parameter **r8784** for the CBC status word by carrying out the following:

1. In the project navigator, search for the p parameter **p8851 [0]** by choosing **Drive_1 > right-click > Expert > Expert list**.
2. Interconnect parameter **p8851 [0]** = PZD 1 with the associated r parameters (**r8784**).

Actual velocity (PZD 2+3 32 bit)

The following table shows the p parameter for the actual velocity, which has to be interconnected with the corresponding source.

Table 3- 20 Interconnect the actual velocity

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

Interconnect p parameter **p8861 [1]** with r parameter **r0063** for the actual velocity by carrying out the following:

1. In the project navigator, search for the p parameter **p8861 [1]** by choosing **Drive_1 > right-click > Expert > Expert list**.
2. Interconnect parameter **p8861 [1]** = PZD 2 + 3 with the associated r parameters (**r0063**).

Actual torque (PZD 4 16 bit)

The following table shows the p parameter for the actual torque, which has to be interconnected with the corresponding source.

Table 3- 21 Interconnect the actual torque

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

Interconnect p parameter **p8851 [3]** with r parameter **r0080** for the actual velocity by carrying out the following:

1. In the project navigator, search for the p parameter **p8851 [3]** by choosing **Drive_1 > right-click > Expert > Expert list**.
2. Interconnect parameter **p8851 [3]** = PZD 4 with the associated r parameters (**r0080**).

Actual position value (PZD 5+6 32 bit)

The following table shows the p parameter for the actual position value, which has to be interconnected with the corresponding source.

Table 3- 22 Interconnect the actual position value

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

Interconnect p parameter p8851 [3] with r parameter r0482 for the actual position value by carrying out the following:

1. In the project navigator, search for the p parameter **p8861 [4]** by choosing **Drive_1 > right-click > Expert > Expert list**.
2. Interconnect parameter **p8861 [4]** = PZD 5 + 6 with the associated r parameters (**r0482 [0…2]**) for the appropriate encoder.

3.6.2.3 Interconnecting an additional drive object

Steps

To interconnect the next drive object (Single Motor Module 2), repeat the steps described in "Interconnecting transmission message frames in the receive buffer" and "Interconnecting transmission message frames in the transmit buffer".

When carrying out the steps, choose **Drive_2** in the project navigator.

3.7 Load the projects from the drive unit to the PC/PG in ONLINE mode and save them

Initial commissioning: procedure

In the table below, the current commissioning step is highlighted in **bold**:

Table 3- 23 CANopen initial commissioning

Step	Procedure
1	Hardware settings on CBC10 (not for SINAMICS S110)
2	Configure the drive unit using the STARTER commissioning tool in 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 ONLINE mode, load the projects from the drive unit to the PC/PG and save.**

Prerequisite

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

Steps

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

1. In the project navigator, select **Drive unit_126**. Click the **Load to PG** function key (see screenshot below).

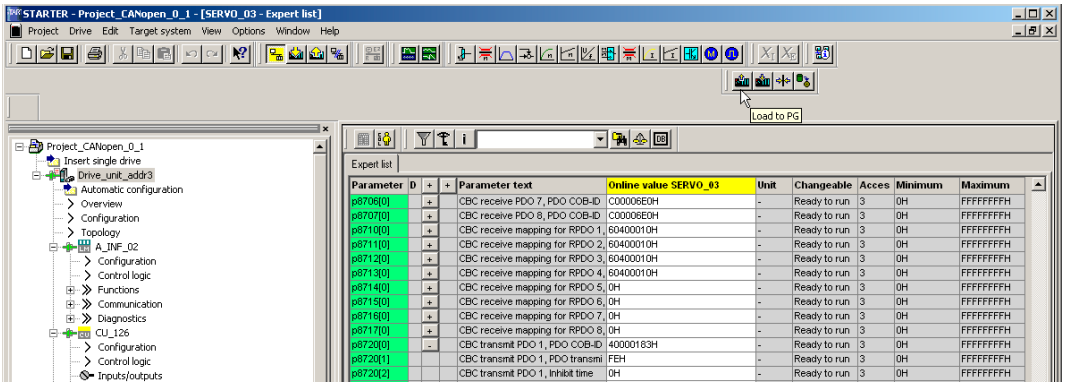


Figure 3-30 "Load to PG" function key

2. Carry out the following:
   - When the system asks "Are you sure?", click **Yes**. The system now starts loading the data.
   - When the system informs you that the data was successfully loaded, click **OK**.
3. Click the **Disconnect from target system** function key.

4. If prompts are displayed, then click on the following one after the other:
   - **Changes in the drive unit...**
   - **Save data,** for SERVO_3
   - When the system informs you that the data was successfully copied from RAM to ROM, click **OK**.
   - When the system prompts you to confirm that you want the data to be loaded to the PG, click **Yes**.
   - When the system informs you that the data was successfully loaded to the PG, click **OK**.
   - **Save data,** for SERVO_4
   - When the system informs you that the data was successfully copied from RAM to ROM, click **OK**.
   - When the system prompts you to confirm that you want the data to be loaded to the PG, click **Yes**.
   - When the system informs you that the data was successfully loaded to the PG, click **OK**.
5. STARTER is in OFFLINE mode.
6. Click **Project > Save as...**

Note

This completes initial commissioning for the CANopen interface.

4 Examples

4.1 Speed setpoint input and evaluation of the speed actual value

Introduction

The CANopen object Target Velocity is accessed using the PDO or SDO transfer.

As standard, the value is displayed in parameter**r8796** (SINAMICS view) in the unit increments/second.

Note

The parameter corresponds to the CANopen object 60FF 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).

  In the factory setting, for CANopen, the velocity units

  – with encoder -> increments/second

  – without encoder -> revolutions/minute

  The internal velocity is calculated as follows:

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

- With encoder

  – **p0408** = pulse number

  – **p0418** = fine resolution

Speed setpoint

- With encoder

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

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

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:

$$\text{C800000 hex} = \frac{3000\ [\text{RPM}]}{60\text{s}} * 2048 * 211 * \frac{1}{1}$$

- Without encoder

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

Actual speed value

The actual velocity is calculated as follows:

- With encoder

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

- Without encoder

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

Note

For "without encoder" it should be observed that this does mean modes such as encoderless closed-loop speed control (p1300=20) or encoderless closed-loop torque control (p1300=22) where an encoder is commissioned anyway!

4.2 Torque setpoint input and evaluation of the torque actual value

Introduction

The CANopen object Target Torque is accessed using the PDO or SDO transfer.

As standard, the value is displayed in parameter**r8796** in per mille (1/1000).

Note

The parameter corresponds to the CANopen object 6071 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:

- **p0333** = rated motor torque

Torque setpoint

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

$$\text{m\_set\_bus [per mille]} = \frac{\text{m\_set [N/m]}}{\text{p0333 [N/m]}} * 1000$$

Actual torque value

The torque actual value is calculated as follows:

$$\text{m\_act [N/m]} = \frac{\text{m\_act\_bus [per mille]}}{1000} * \text{p0333 [N/m]}$$

Diagnosis

5

Section content

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

- One emergency object per drive unit
- A device-internal error list ("predefined error field")
- An error register

Note

See also the following standards:

- CiA DS-301 V4.01 (Application Layer and Communication Profile)
- CiA DS-402 V2.0 (Device Profile for Drives and Motion Control)

5.1 Emergency object

Introduction

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

The relevant parameters are located:

- 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 asynchronously. The emergency message 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 Chapter).

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. Faults trigger emergency messages 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")

Introduction

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

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

This list contains the individual, unacknowledged faults and pending alarms in the CANopen alarm number range A8700-A8799 diagnosed in a drive unit. The faults (errors) are listed in the order in which they occur, along with an error code and additional, device-specific information. As soon as a fault is acknowledged or an alarm is resolved, they are deleted from the drive device internal error list.

Note

Faults whose cause has been resolved and which only need to be acknowledged are no longer included in the predefined error field. These faults are still visible in parameter r0945.

All drive objects in the selected CU 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
8110 hex	CAN overflow, message lost	CBC: Message frame loss (A8751) [alarm]
8120 hex	CAN Error Passive	CBC: Error number for Error Passive exceeded (A8752) [alarm]
8130 hex	CAN Life Guard Error	CBC: Communications error, alarm value 2 F8700(A) [fault/alarm]
1000 hex	CAN Error 1	All other SINAMICS faults
1001 hex	CAN Error 2	All other CANopen alarms in the alarm number range A8700 to A8799

Note

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

5.3 Error register

Introduction

The 1-byte error register can be read via:

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

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	Set for CAN communication alarms, i.e. for alarms in the number range A08700-A08799.
Bit 7	Manufacturer error	Set for all SINAMICS errors outside the CAN communication alarm number range.

Communication objects

6

Section content

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

- 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.

See also

CANopen object directory (Page 22)

6.1 Control Unit communication objects independent of the drive

Overview

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 range in which they are located for SINAMICS.

Table 6- 1 Control Unit communication objects independent of the drive

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Default values	Read/ write
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	rw
	0	Number of errors	p8611.0	SDO	Unsigned32	0	rw
	1	Number of module	p8611.1	SDO	Unsigned32	0	rw
	2	Number of errors: module 1	p8611.2	SDO	Unsigned32	0	rw
	3-A	Standard error field: module 1	p8611.3-p8611.10	SDO	Unsigned32	0	rw
	B	Number of errors: module 2	p8611.11	SDO	Unsigned32	0	rw

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Default values	Read/ write
	C-13	Standard error field: module 2	p8611.12-p8611.19	SDO	Unsigned32	0	rw
	14	Number of errors: module 3	p8611.20	SDO	Unsigned32	0	rw
	15-1C	Standard error field: module 3	p8611.21-p8611.28	SDO	Unsigned32	0	rw
	1D	Number of errors: module 4	p8611.29	SDO	Unsigned32	0	rw
	1E-25	Standard error field: module 4	p8611.30-p8611.37	SDO	Unsigned32	0	rw
	26	Number of errors: module 5	p8611.38	SDO	Unsigned32	0	rw
	27-2E	Standard error field: module 5	p8611.39-p8611.46	SDO	Unsigned32	0	rw
	2F	Number of errors: module 6	p8611.47	SDO	Unsigned32	0	rw
	30-37	Standard error field: module 6	p8611.48-p8611.55	SDO	Unsigned32	0	rw
	38	Number of errors: module 7	p8611.56	SDO	Unsigned32	0	rw
	39-40	Standard error field: module 7	p8611.57-p8611.64	SDO	Unsigned32	0	rw
	41	Number of errors: module 8	p8611.65	SDO	Unsigned32	0	rw
	42-49	Standard error field: module 8	p8611.66-p8611.73	SDO	Unsigned32	0	rw
	4A-52	NOP	p8611.74-p8611.82	SDO	Unsigned32	0	rw
1005		SYNCH COB ID	p8602.0	SDO	Unsigned32	128	rw
1006		Communication cycle period	p8602.1	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 sub-index supported		SDO			
	1	Save all parameters	p0977	SDO	Unsigned16	0	rw
	2	Save communication parameters (0x1000-0x1fff)	p0977	SDO	Unsigned16	0	rw

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Default values	Read/ write
	3	Save application-related parameters (0x6000-0x9fff)	p0977	SDO	Unsigned16	0	rw
1011		Restore default parameters	p0976	SDO	Unsigned16	0	rw
	0	Largest sub-index 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	p8603.0	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					
	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
41AE		Baud rate selection	p8622	SDO	Integer16	6 (20kBit/s)	rw

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Default values	Read/ write
41B6		Virtual objects	p8630 [0...2]	SDO	Unsigned16		rw
	0	Axis number	p8630.0		Unsigned16	1	rw
	1	Sub-index range	p8630.1		Unsigned16	0	rw
	2	Parameter range	p8630.2		Unsigned16	0	rw

6.2 Drive-dependent communication objects

Introduction

Eight transmit/receive PDOs can be parameterized for each drive, whereby a total of 25 PDOs (channels) 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".

6.2.1 Table: Communication objects for receive PDOs

Overview

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

Table 6- 2 Drive-dependent communication objects for receive PDOs

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
1400		**Receive PDO 1 communication parameter**					
	0	**Largest sub-index 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 sub-index 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

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
1402		Receive PDO 3 communication parameter					
	0	Largest sub-index 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 sub-index 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 sub-index 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 sub-index 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 sub-index 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 sub-index supported		SDO	Unsigned8	2	ro
	1	COB ID used by PDO	p8707.0	SDO	Unsigned32	C000 06E0 hex	rw

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
	2	Transmission type	p8707.1	SDO	Unsigned8	FE hex	rw
1600		Receive PDO 1 mapping parameter					
	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)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
	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
	4	PDO mapping for the fourth application object to be mapped	p8714.3	SDO	Unsigned32	0	rw

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
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
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

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
	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.

6.2.2 Table: Communication objects for transmit PDOs

Overview

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

Table 6- 3 Drive-dependent communication objects for transmit PDOs

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
1800		Transmit PDO 1 communication parameter					
	0	Largest sub-index supported		SDO	Unsigned8	5	ro
	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					

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
	0	Largest sub-index 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 sub-index 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 sub-index 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 sub-index supported		SDO	Unsigned8	5	ro
	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					

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
	0	Largest sub-index 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 sub-index 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 sub-index 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**					
	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**

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
	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
	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	6074 hex	rw
	3	PDO mapping for the third application object to be mapped	p8732.2	SDO	Unsigned32	0	rw

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
	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
	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

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
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
	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					

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Predefined connection set	Read/ write
	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.

6.3 Free objects

OD index (hex)	Description	Data type per PZD	Pre-set values:	Can be written to/read
5800 to 580F	16 freely-interconnectable receive process data	Integer16	0	rw
5810 to 581F	16 freely-interconnectable transmit process data	Integer16	0	ro
5820 to 5827	8 freely-interconnectable receive process data	Integer32	0	rw
5828 to 582F	Reserved			
5830 to 5837	8 freely-interconnectable transmit process data	Integer32	0	ro
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.

e.g.: If drive object 2 starts from 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): 4000hex corresponds to 100%
- 32-bit (word): 4000000hex corresponds to 100%

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

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

See also

Free objects (Page 28)

6.4 Objects in drive profile DSP402

Overview

The following table lists the object directory with the index of the individual objects for the drives. The column "SINAMICS parameter" shows the parameter range in which they are located for SINAMICS S120.

CANopen currently supports profile velocity mode for SINAMICS S120.

Table 6- 4 Objects in drive profile DSP402

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Default values	Read/ write
Predefinitions							
67FF		Single device type		SDO			
Common entries in the object dictionary							
6007		Abort connection option code	p8641	SDO	Integer32	0	rw
6502		Supported drive modes		SDO	Integer32		
6504		Drive manufacturer		SDO	String		
Device control							
6040		Control word	p8890	PDO/SDO	Unsigned16	–	rw [1)]
6041		Status word	r8784	PDO/SDO	Unsigned16	–	ro
6060		Modes of operation	p1300	SDO	Integer16	–	rw
6061		Modes of operation display	p1300	SDO	Integer16	–	rw
Factor group							
6094		Velocity encoder factor					
	01	Velocity encoder factor numerator		SDO	Integer32		
	02	Velocity encoder factor denumerator		SDO	Integer32		
Profile Position Mode							
6083		Profile acceleration	p1082/p1120	SDO			
6084		Profile deceleration	P1082/p1121	SDO	0	0	0
6085		Quick stop deceleration	P1082/p1135	SDO	0	0	0
6086		Motion profile type	p1115/p1113	SDO			
Profile velocity mode							
6069		Velocity sensor actual value	r0061	SDO/PDO	Integer32	–	ro
6063		Actual position value	r0482	SDO/PDO	Integer32	–	ro
606B		Velocity demand value	r1170	SDO/PDO	Integer32	–	ro

OD index (hex)	Sub-index (hex)	Object name	SINAMICS parameter	Transmission	Data type	Default values	Read/ write
606C		Velocity actual value Actual velocity	r0063	SDO/PDO	Integer32	–	ro
6071		Target torque Set torque	p1513[0]	SDO/PDO	Integer16	–	rw 1)
6072		Max. torque	P1520/P1521	SDO	0	0	0
6074		Torque demand value Actual torque	r0080	SDO/PDO	Integer16	–	ro
60FF		Target velocity Set velocity	Without ramp-function generator -> p1155[0] With ramp-function generator -> p1070	SDO/PDO	Integer32	–	rw 1)

1) SDO access is only possible after mapping the objects and the BICO interconnection to r parameters.

Note

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

Parameters, faults and alarms, terminology

A

Section content

Note

All the parameters, faults, alarms, and function diagrams for CANopen in the SINAMICS drive line-up are described in: /LH1/ SINAMICS S List Manual.

A.1 Glossary

Active Line Module

Controlled, self-commutating infeed/regenerative feedback unit (with -"IGBTs" in infeed/regenerative feedback direction), which supplies the DC link voltage for the -> "Motor Module".

Drive

The drive includes the motor (electric or hydraulic), the actuator (converter, valve), the control, measuring system, and supply components (line infeed module, pressure reservoir). For electric drives, a distinction is made between a converter system and an inverter system. With a converter system (e.g. -> "MICROMASTER 4"), from the point of view of the user the line infeed, actuator, and control component form a single device; with an inverter system (e.g. -> "SINAMICS S"), the supply is ensured by means of -> "Line Module", thereby realizing a DC link to which the -> "Inverters" (-> "Motor Module"s) are connected. The (–> "Control Unit") is implemented as a separate device and connected to the other components by means of –> "DRIVE-CLiQ".

Drive unit

The drive unit includes all the components connected via –> "DRIVE-CLiQ" that are required for carrying out a drive task: -> "Motor Module" –> "Control Unit" –> "Line Module" and the required –> "Firmware" and –> "Motor"s, but not additional components, such as filters or reactors.
Several –> "Drive"s can be implemented in a drive unit.
See –> "Drive system".

Drive component

Hardware components connected to a –> "Control Unit" via –> "DRIVE-CLiQ", for example. Drive components are for instance: -> "Motor Module"s, -> "Line Module"s, -> "Motor"s, -> "Sensor Module"s and -> "Terminal Module"s.
The overall layout of a Control Unit together with the connected drive components is called -> "Drive unit".

Drive object

A drive object is an autonomous, individual software function with its own -> "Parameter"s and may also have its own -> "Fault"s and -> "Alarm"s. Drive objects may exist by default (e.g. On Board I/O), can be created individually (e.g. –> "Terminal Board" 30, TB30) or also as multiples (e.g. –> "Servo control"). As a rule, each drive object has its own window for parameterization and diagnostic purposes.

Drive parameter

Parameters of a drive axis that include, for example, the parameters of the corresponding controllers, as well as the motor and encoder data. The parameters of the higher-level technology functions (positioning, ramp-function generator), however, are called –> "Application parameter"s.
See –> "Basic unit system".

Drive system

The drive system includes all the components in a product family that belong to a drive (e.g. SINAMICS). A drive system comprises, for example, "Line Module"s, -> "Motor Module"s, -> "Encoder"s, -> "Motor"s, -> "Terminal Module"s and -> "Sensor Module"s as well as additional components (reactors, filters, cables, etc.).
See -> "Drive unit"

Drive line-up

A drive line-up comprises a -> "Control Unit" and the -> "Motor Module"s and -> "Line Module"s connected via -> "DRIVE-CLiQ".

Basic Infeed

Overall functionality of an infeed with –> "Basic Line Module" including the required additional components (filters, switching devices, etc.).

Basic Line Module

Unregulated line infeed unit (diode bridge or thyristor bridge, without feedback) for rectifying the line voltage of the -> "DC link".

CompactFlash card

Memory card for non-volatile storage of the drive software and corresponding –> "Parameter"s. The memory card can be plugged into the –> "Control Unit" from the outside.

Control Unit

Central control module in which the closed-loop and open-loop control functions for one or more -> "SINAMICS" -> "Line Module"s and/or -> "Motor Module"s are implemented.
There are three types of Control Unit:
- SINAMICS Control Units, e.g. -> "CU320"
- SIMOTION Control Units, e.g. -> "D425" and -> "D435"
- SINUMERIK solution line Control Units, e.g. NCU710, NCU720 and NCU730

Double Motor Module

Two motors can be connected to and operated with a Double Motor Module.
See –> "Motor Module" –> "Single Motor Module"
Former term: –> "Double-axis module"

DRIVE-CLiQ

Abbreviation for "Drive Component Link with IQ".
Communication system for connecting the various components in a SINAMICS drive system (e.g. -> "Control Unit", -> "Line Module"s, -> "Motor Module"s, -> "Motor"s, and speed/position encoders.
The DRIVE-CLiQ hardware is based on the Industrial Ethernet standard and uses twisted-pair lines. The DRIVE-CLiQ line provides the transmit and receive signals, as well as the +24 V power supply.

Infeed

Input component of a converter system for generating a DC link voltage to supply one or more -> "Motor Module"s, including all the required components (e.g. -> "Line Module"s, fuses, reactors, line filters, and firmware, as well as proportional computing power (if required) in a -> "Control Unit".

External encoder

Position encoder that is not built in or mounted on the –> "Motor", but is instead fitted outside via a mechanical transmission element or mechanical intermediate element.
The external encoder (see –> "Externall mounted encoder") is used for –> "Direct position detection".

Encoder

An encoder is a measuring system that captures actual values for the speed and/or angular/position values and makes them available for electronic processing. Depending on the mechanical design, encoders can be integrated in the –> "Motor" (–> "Motor encoder") or mounted on the external mechanical equipment (– "External encoder"). Depending on the type of movement, a distinction is made between rotary encoders ("rotary transducers") and translatory encoders (e.g. - Linear Sensors). In terms of measured value provision, a distinction is made between –> "Absolute encoders" (code encoders) and –> "Incremental encoders".
See –> "Incremental encoder TTL/HTL" –> "Incremental encoder sin/cos 1 Vpp" –> "Resolver".

Line Module

A Line Module is a power unit that generates the DC link voltage for one or more –> "Motor Module"s from a 3-phase line voltage.
In SINAMICS, the following three types of Line Module are available:
–> "Basic Line Module", –> "Smart Line Module" –> "Active Line Module".
The overall function of an infeed, including the required additional components such as a –> "Line Reactor", proportional computing power in a –> "Control Unit", switching devices, etc. is called –> "Basic Infeed", –> "Smart Infeed", and –> "Active Infeed".

Motor

For the electric motors that can be driven by –> "SINAMICS", a basic distinction is made between rotary and linear motors with regard to their direction of motion, and between synchronous and induction motors with regard to their electromagnetic operating principle. In SINAMICS, the motors are connected to a –> "Motor Module".
See –> "Synchronous motor" –> "Induction motor" –> "Built-in motor" –> "Motor encoder" -> "External encoder" –> "Third-party motor".

Motor Module

A Motor Module is a power unit (DC-AC inverter) that supplies the power for the connected motor(s).
Power is supplied through the –> "DC link" of the –> "Drive unit".
A Motor Module must be connected to a –> "Control Unit" via –> "DRIVE-CLiQ". The open-loop and closed-loop control functions for the Motor Module are stored in the Control Unit.
–> "Single Motor Module"s and –> "Double Motor Module"s are available.

Motor encoder

An –> "Encoder" (e.g. –> "Resolver", –> "Incremental encoder TTL/HTL", or –> "Incremental encoder sin/cos 1 Vpp" that is integrated in or attached to the motor.
The encoder detects the motor speed. In the case of synchronous motors, it can also detect the rotor position angle (of the commutation angle for the motor currents).
For drives without an additional –> "Direct position measuring system", it is also used as a –> "Position encoder" for position control.
In addition to the motor encoders, –> "External encoders" for –> "Direct position sensing" are available.

Option slot

Slot for an optional module (e.g. in the –> "Control Unit").

Parameter

Variable quantity within the drive system that the user can read and, in some cases, write.
For –> "SINAMICS", all specifications defined in the –> "PROFIdrive" profile are defined by a parameter.
See –> "Visualization parameter"s and –> "Adjustable parameter"s.

PROFIBUS

Field bus to IEC 61158, Sections 2 to 6.
The abbreviation "DP" is no longer included because PROFIBUS FMS is not standardized and PROFIBUS PA (for Process Automation) is now part of the "general" –> "PROFIBUS".

Sensor Module

Hardware module for evaluating speed/position encoder signals and providing detected actual values as numerical values at a -> DRIVE CLiQ socket".
Three mechanical Sensor Module variants are available:
- SMCxx = Sensor Module Cabinet-Mounted
- SME = Sensor Module Externally Mounted (with a high degree of protection)

Servo drive

An electric servo drive comprises a motor, a –> "Motor Module", a –> "Servo Control" and, in most cases, a speed and position –> "Encoder"
Electric servo drives are normally extremely precise and have a high dynamic response. They are designed for cycle times to less than 100 ms, and often have a short-time overload capacity, which enables quick acceleration. Servo drives are available as rotary and linear drives and are used for machine tools, handling robots, and packaging machines.

Servo control

For -> "Motor"s equipped with a -> "Motor encoder", this control type allows operation with a high level of -> "Accuracy" and -> "Dynamic response".
In addition to speed control, position control can also be implemented.

SITOP power

Components for –> "Electronics power supply".
Example: 24 V DC

Smart Line Module

Unregulated infeed/regenerative feedback unit with a diode bridge for the infeed and stall-protected, line-commutated regenerative feedback via -> "IGBTs".
The Smart Line Module supplies the DC link voltage for the -> "Motor Module"s.

Memory card

An external memory card for non-volatile storage of the drive software and corresponding –> "Parameter"s. The memory card can be plugged into the –> "Control Unit" from the outside. Different types of cards can be used depending on the device -> SD Card, MMC Card, "CompactFlash Card".

Control word

Bit-coded -"Process data" word, transmitted by -> "PROFIdrive" at cyclic intervals to control the drive states.

Vector control

Vector control (field-oriented control) is a high-performance control type for induction machines. It is based on an exact model calculation of the motor and two current components that simulate and accurately control the flux and torque by means of software algorithms, thereby enabling predefined speeds and torques to be observed and limited accurately and with a good dynamic response.
Two vector control types exist:
Frequency control (–> "Sensorless vector control") and speed-torque control with speed feedback (–> "Encoder").

Status word

Bit-coded -> "Process data" word, transmitted by -> "PROFIdrive" at cyclic intervals to control the drive states.

Index

L

M

N

O

P

R

S

T

U

V

W

X

Siemens AG
Industry Sector
Drive Technologies
Motion Control Systems
Postfach 3180
91050 ERLANGEN
GERMANY

www.siemens.com/motioncontrol