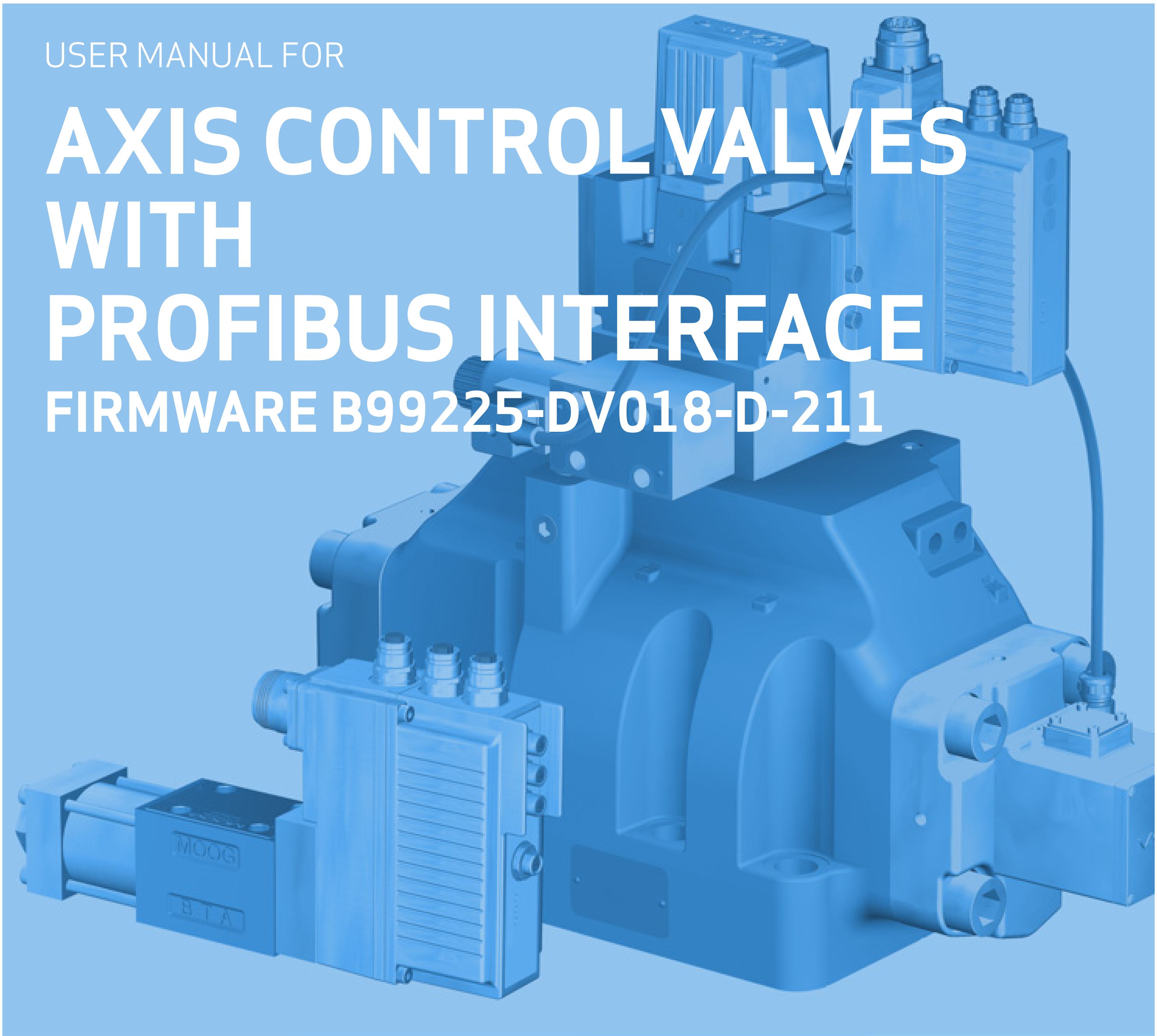
USER MANUAL FOR

AXIS CONTROL VALVES WITH PROFIBUS INTERFACE

FIRMWARE B99225-DV018-D-211



Rev. A, October 2018

OFFERING FLEXIBLE INTEGRATION AND ADVANCED MAINTENANCE FEATURES INCLUDING DIAGNOSTICS, MONITORING OF CHARACTERISTICS AND ABILITY TO DEFINE DYNAMIC BEHAVIORS

WHAT MOVES YOUR WORLD

MOOG

Copyright

Hanns-Klemm-Straße 28
71034 Boeblingen
Germany

Telephone:	+49 7031 622-0
Fax:	+49 7031 622-191
E-mail:	sales@moog.com
Internet:	http://www.moog.com/Industrial

Table of contents

List of tables

List of figures

1 General information

1.1 About this manual

This document describes the Profibus DP field bus interface of the Moog servo valves.

It describes and explains the general structure of the Profibus DP field bus interface and in a second part the device specific profile for hydraulic valves will be explained.

All parameters follow the common PROFIBUS-DP Profile for Fluid Power Technology, Version 1.5, released by the PROFIBUS Nutzerorganisation e.V. (PNO).

This manual is part of the set of documentation available for the servo valve.
⇨ Chapter "1.4 Further documentation for the servo valve", page 4

This document is not a replacement for the Profibus DP standards as listed in the references.
⇨ Chapter "1.5 References", page 5

This manual was prepared with great care and the contents reflect the author's best knowledge. However, the possibility of errors remains and improvements are possible.
Please feel free to submit any comments regarding errors or possibly incomplete information to Moog.

1.1.1 Reservation of changes and validity

The information contained in this manual is valid at the time of this version's release. See footer for version number and release date of this manual.
We reserve the right to make changes to this manual at any time without specified reasons.

1.1.2 Completeness

This manual is complete only when used in conjunction with the product related hardware and software documentation required for the relevant application.

1.1.3 Place of storage

This manual and all other associated documentation for hardware and software must always be kept in a location where they will be readily accessible and close to the servo valve or the equipment in which it is installed.

1.1.4 Warranty and liability

This manual only describes the functionality and influence of the parameters. The described software functionality can be used in various servo valve models which can be implemented in a vast range of applications. Hence it is not possible to assume liability for the influence of the parameters. Please refer to the safety instructions and remarks in the related operating instructions.

1.1.5 Typographical conventions

⚠ DANGER

Identifies safety instructions that are intended to warn of an immediate and impending danger to life and limb.
Failure to observe these safety instructions will inevitably lead to death, serious personal injury (disablement)!

⚠ WARNING

Identifies safety instructions that are intended to warn of potential danger to life and limb.
Failure to observe these safety instructions might lead to death, serious personal injury (disablement)!

⚠ CAUTION

Identifies safety instructions that are intended to warn of slight personal injury.
Failure to observe these safety instructions might lead to slight personal injury.

NOTICE

Failure to observe this safety notice can result in property damage!

ⓘ Identifies important information

• / -	Identifies listings
⇨	Identifies references to another chapter, page, table or figure in this manual
blue text	Identifies a hyperlink within the PDF file
1., 2., …	Identifies steps in a procedure that should be performed in consecutive order
'STATE'	Identifies states of a state machine
«MS»	Identifies LEDs of the servo valve (for example, «MS»)
< >	Identifies a parameter name
"…"	Used for references

1.2 Structure of warning notices

The warning notices in this user manual have the following structure:

1 ⚠ **DANGER** 2

Moving machine parts! 3

Entrapment hazard! 4

▸ Do not enter danger zone! 5

Figure 1: Structure of a warning notice

Legend

1 Warning symbol
2 Signal word
3 Type and source of hazard
4 Possible consequences if a potential hazard
5 Hazard prevention measures

1.3 Selection and qualification of personnel

Only qualified users may work with the servo valve. Qualified users are properly trained experts with the required knowledge and experience. In particular, these experts must have the authorization to bring into operation systems and power circuits in accordance with safety engineering standards. They must be familiar with safety concepts common in automation.

1.4 Further documentation for the servo valve

This manual is part of the complete set of documentation for the servo valve, which includes the following documents:

User manuals	
B97072-670	Product Installation Instruction Servo- Proportional Valves and Servovalves D67x Series
B97072-636	User Manual Mounting and Installation Notes D636/7/8/9 Series - Servovalves
B97072-630	User Manual Mounting and Installation Notes, Servovalves D630 Series
B97072-941	Mounting and Installation Notes, pQ-Proportional Valves D941/2/3/4 Series
CA63420-001	User Manual Electrical Interfaces Description of the electrical interfaces for the series D636, D637, D638, D639, D67x, D930, D94x and the RKP-D
B95872-001	Operating Instructions D636 and D637/D638 Series Direct-Operated Servovalves
C43357-001-en + de	Operating Instructions D941 Series Two-State pQ-Proportional Valves with Integrated Digital Electronics and CAN Bus Interface
CA45707-002	Betriebsanleitung Servoventile Serie D636/D638 Direktbetätigte Servoventile
CA61892-001	User Manual Direct Drive Servovalves with integrated Digital Electronics and Fieldbus, Size NG10 D637-R/D639-R Series
CA75181-002	Benutzerinformation Vorgesteuerte Proportionalventile, Größe NG10 - NG32 Baureihe D67x nach ISO 4401
Explosion proof valves	
CDS29587-en	User Manual for Direct Driven Servovalves with Integrated Digital Electronics (explosion proof) Series D636K and D638K, Size 03
CDS29577-en	User Manual for Direct Operated Servo- and Proportional Valves with Integrated Digital Electronics (explosion proof) Series D637K und FD639K, Size 05
CDS29588-en	User Manual for Pilot operated Proportional Valves with Integrated Digital Electronics (explosion proof) D67xK Series
CDS29589-en	User Manual for Pilot Operated Proportional valves with Integrated Digital Electronics (explosion proof) Series D94xK
Miscellaneous documents	
CA58437-001	Technical Note TN353 Protective Grounding and Electrical Shielding of Valves
CA48851-001	Technical Note TN494 Maximum Permissible Length of Electric Cables for Valves with Integrated Electronics
CDL28319-en	Catalog - D671-D672-D673-D674-D675 Series Drive Servo-Proportional Valves with Integrated Digital Electronics and CAN bus Interface

Visit http://www.moog.com/industrial/literature to download the desired documents.

1.5 References

In this chapter you will find information about standards for Profibus DP, CANopen and the used device profile. In the following table you see all relevant organizations for standardization.

ISO	International Organization for Standardization 1, ch. de la Voie-Creuse, Case postale 56 CH-1211 Geneva 20 http://www.iso.org
IEC	International Engineering Consortium 233 S. Wacker Drive, Suite 8400 Chicago, IL 60606-6338 USA http://www.iec.org
CiA	CAN in Automation Kontumazgarten 3 DE-90429 Nürnberg http://www.can-cia.org
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V. Lyoner Strasse 18 60528 Frankfurt/Main http://www.vdma.org
PI / PNO	PI Support Center Haid-und-Neu-Strasse 7 76131 Karlsruhe http://www.profibus.com

1.5.1 Profibus field bus

The Profibus DP field bus interface provides a connection to the servo valves using Profibus DP frames according to IEC 61158.

IEC 61158-2-3 IEC 61784-5-3 Anhang A	Profibus DP Physical Layer specification Installing profiles for communication networks
IEC 61158-3-3 IEC 61158-4-3	Profibus DP Data-link service definition Profibus DP Data-link protocol specification
PIEC 61158-5-3 IEC 61158-6-3	Profibus DP Application layer service definition Profibus DP Application layer protocol specification

1.5.2 Device Profile

VDMA Profile Fluid Power	Device profile for Proportional Valves and Hydrostatic Transmissions VDMA Profile Fluid Power Technology Version 1.6
CiA 408 or Device Profile Fluid Power	CiA 408 Device profile for fluid power technology proportional valves and hydrostatic transmissions, Version 1.5.2
Device Profile Fluid Power	PNO Profibus DP Profile "Fluid Power Technology", Version 1.5, in accordance to CiA 408

1.6 Definitions

1.6.1 Internal resolution (iR)

The internal resolution is 16384 (0x4000) at 100 % and –16384 (0xC000) at –100 % of the value range.

1.6.2 Volume flow direction

A positive spool position demand value will result in a volume flow from connection P to A.

1.6.3 Servo valve position and stage names

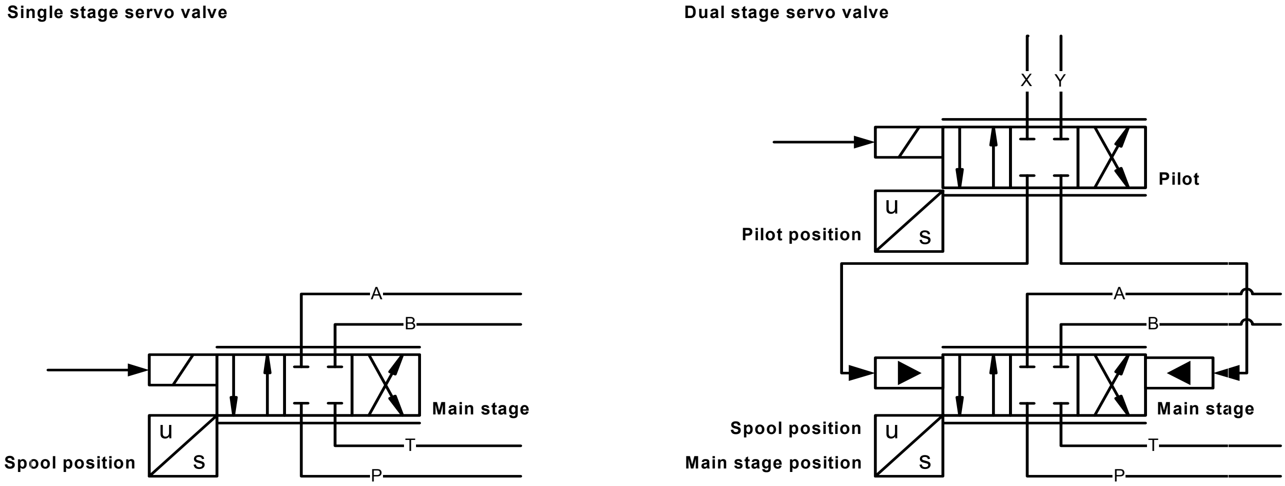


Figure 2: Servo valve position and stage names

1.7 Abbreviations

Abbreviation	Explanation
AC	Alternating Current
CAN	Controller Area Network
CANopen	ISO/OSI Layer 7 protocol, specified by CAN in Automation (CiA)
CiA	CAN in Automation
COB-ID	Communication Object Identifier
DC	Direct Current
DCV	Moog Digital Control Valve
DSM	Device State Machine
DSP	Digital Signal Processor
EDS	Electronics Datasheet, containing a description of the CANopen object dictionary
EEPROM	Electrically erasable programmable read-only memory
iR	Internal resolution defined by CiA 408
ISO	International Engineering Consortium
LED	Light Emitting Diode
LVDT	Linear Variable Differential Transformer used to measure the valves spool position
NMT	Network management according CANopen
NS	Network Status
OD	Object Dictionary
OSI	Open Systems Interconnection
P	Proportional gain element
PD	Proportional derivative element
PDO	Process Data Object
PE	Protective earth / Electrical grounding
PI	PROFIBUS & PROFINET International (PI)
PNO	PROFIBUS Nutzerorganisation e.V. (PNO) / Profibus User Organisation
PPT1	Proportional first order lag element
ro	Read only

Table 1: Abbreviations (part 1 of 2)

Abbreviation	Explanation
rw	Read write
RxPDO	Receive Process Data Object
RxPDO remote	Receive Process Data Object remote
RxSDO	Receive Service Data Object
SDO	Service Data Object
TR	State transmission of the valve application state machine
TxPDO	Transmit Process Data Object
TxPDO remote	Transmit Process Data Object remote
TxSDO	Transmit Service Data Object
URL	Uniform Resource Locator / Internet address
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V.
wo	Write only
Xn	Physical connector n for electrical connection

Table 1: Abbreviations (part 2 of 2)

1.8 Trademarks

Moog and Moog Authentic Repair® are registered trademarks of Moog Inc. and its subsidiaries.

All the product and company names mentioned in this document are possibly proprietary names or trademarks of the respective manufacturers. The use of these names by third parties for their own purposes may infringe the rights of the manufacturers.
It cannot be inferred from the absence of the ® or ™ symbol that the designation is a free brand name.

2 Access over Profibus DP

2.1 Introduction

Profibus (Process Field Bus) is a universal field bus, which is used primarily in manufacturing, process and building automation. It was developed by Siemens and the Profibus User Organization (PNO) and standardized in the IEC 61158 series of international standards. There are 3 Profibus variants: DP, FMS and PA. Moog servo valves support the DP variant and are designed for the usual transfer rates up to 12 Mbaud. Profibus enables the communication of devices of different manufacturers without special interface adjustments. Profibus specifies the technical characteristics of a serial field bus system which can be used to network distributed digital automation devices from the field level to the cell level. Profibus is a multi-master system which enables the shared operation of several automation, engineering or visualization systems with the distributed peripheral devices on the same bus. The protocol architecture is oriented to the OSI (Open System Interconnection) reference model, conforming to the international standard ISO 7498. Specific tasks are assigned to each transmission layer. Layer 1 (Physical Layer) defines the physical transmission. Layer 2 (Data Link Layer) defines the bus access method and layer 7 (Application Layer) the interface to the application. Profibus is suitable for both fast, time-critical applications and for complex communication tasks.

2.2 Slave reference model

The architecture of the Profibus DP stack with Physical Layer, Data Link Layer and Application Layer was taken from the ISO/OSI Reference Model. Layers three to six of this 7 layer reference model are not required in real time field bus communication, hence they are not implemented for Profibus.

Layer	Description	References Profibus DP-V1
	Application-/Device-profiles	PNO "Fluid Power Technology"
Layer 7	Application Layer	IEC 61158-5-3 IEC 61158-6-3
Layer 6	Presentation Layer (not implemented)	
Layer 5	Session Layer (not implemented)	
Layer 4	Transport Layer (not implemented)	
Layer 3	Network Layer (not implemented)	
Layer 2	Data Link Layer	IEC 61158-3-3 IEC 61158-4-3
Layer 1	Physical Layer	IEC 61158-2-3 IEC 61784-5-3 annex A

Table 2: Slave reference model

2.3 Device Profile

Profiles are pre-defined configurations of functions and features available from Profibus for use in specific devices or applications. They are specified by PNO working groups and published by PI. Profiles are important for openness, interoperability and interchangeability, so that the end user can be sure that similar equipment from different vendors perform in a standardized way.

There are three different profile standards used for the implementation of the servo valve. The mandatory objects of the following profile standards are completely implemented in the servo valve.

- PNO Profibus DP Profile "Fluid Power Technology"
- CiA 408 "Device profile for fluid power technology proportional valves and hydrostatic transmissions"

Both are derived from the fieldbus independent VDMA "Profile for fluid power technology".

2.4 Profibus DP overview

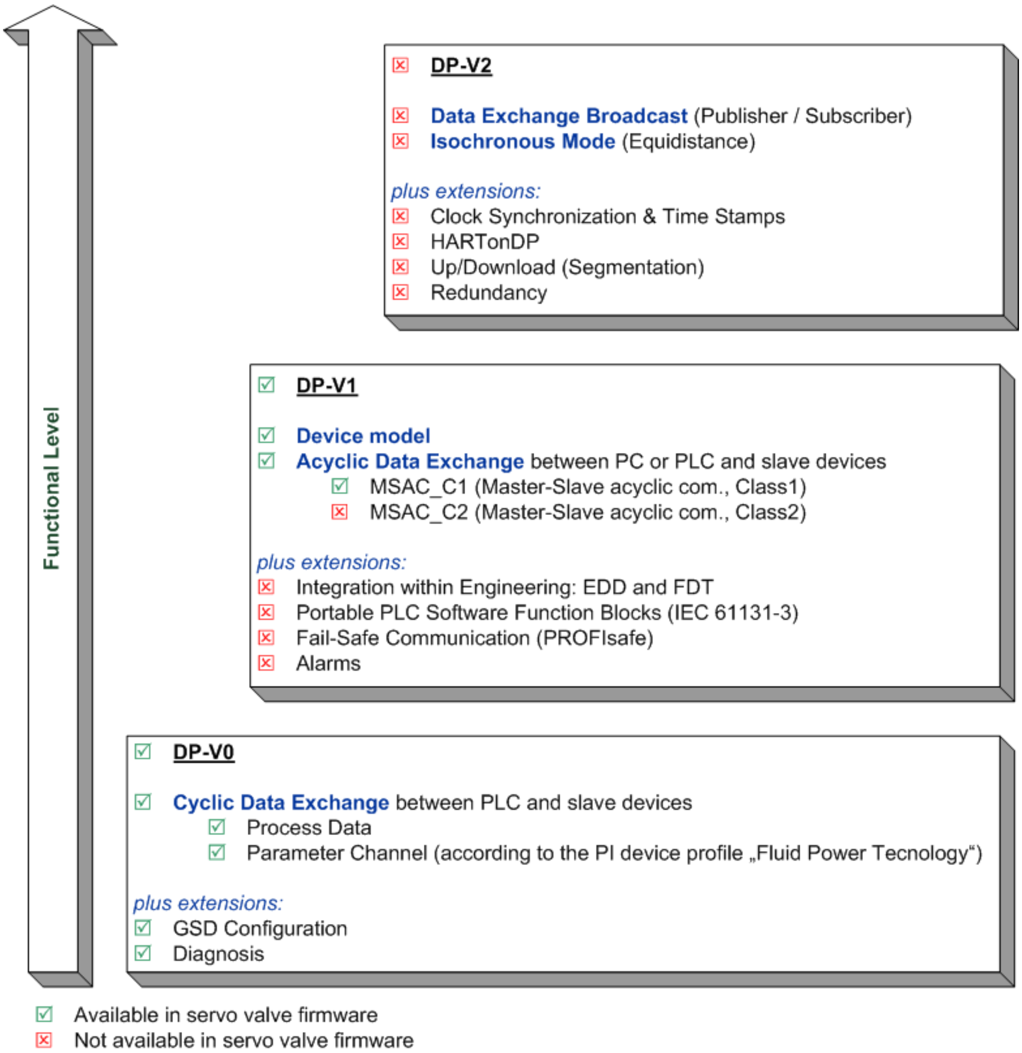


Figure 3: Profibus DP overview

The communication protocol DP (Decentralized Peripherals) has been designed for fast data exchange at field level. This is where central programmable controllers, such as PLCs, PCs or process control systems communicate with distributed field devices, such as I/Os, drives, servo valves, transducers or analysis devices, over a fast serial connection. Data exchange with the distributed devices is primarily cyclic. The communication functions required for this are specified through the DP basic functions (version DP-V0). Geared towards the special demands of the various areas of application, these basic DP functions have been expanded step-by-step with special functions. DP is now available in three versions; DP-V0, DP-V1 and DP-V2, whereby each version has its own special key features. This breakdown into versions largely reflects the chronological sequence of specification work as a result of the ever-increasing demands of applications. Versions V0 and V1 contain both "characteristics" (binding for implementation) and options, while version V2 only specifies options. The key contents of the three versions are as follows.

- **DP-V0**
  DP-V0 provides the basic functionality of DP, including cyclic data exchange as well as station diagnosis, module diagnosis and channel-specific diagnosis.
- **DP-V1**
  DP-V1 contains enhancements for process automation, in particular acyclic data communication for parameter assignment, operation, visualization and alarm handling of intelligent field devices, parallel to cyclic user data communication. This permits online access to stations using engineering tools. In addition, DP-V1 defines alarms like status alarms, update alarms and manufacturer-specific alarms.

- **DP-V2**
  DP-V2 contains further enhancements and is geared primarily towards the demands of electrical drive technology. Due to additional functionalities, such as isochronous slave mode and slave-to-slave communication (DXB, Data eXchange Broadcast) etc., the DP-V2 can also be implemented as a drive bus for controlling fast movement sequences in drive axes.

The various versions of DP are specified in detail in the IEC 61158.

2.5 Profibus DP Device model DP-V1

A Profibus-DP slave can have many configuration parameters. These parameters are grouped in slots. Every slot can be addressed by a slot number. Each slot can contain up to 255 parameters addressed by an index.

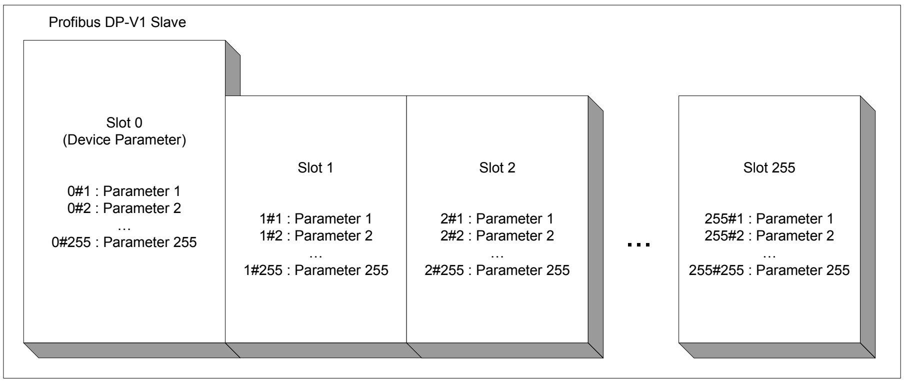


Figure 4: Profibus DP Device model DP-V1

2.5.1 Slot definitions

According to the Device Profile Fluid Power the slots are used to group the parameters thematically.

Slot	Description
0	Device block
1	Reserved for drive actual value conditioning transducer block
2	Valve actual value conditioning transducer block
3	Reserved for drive output processing block
4…10	Reserved for transducer blocks
11	Reserved for drive open loop control function block
12	Reserved for drive position control loop function block
13	Reserved for drive speed control loop function block
14	Reserved for drive force pressure control function block
15…20	Reserved for drive control blocks
21	Valve spool position control function block
22	Valve pressure control function block
23	Reserved for valve PQ control function block
24…30	Reserved for valve controller blocks
31…63	Reserved for future profile extensions
64…255	Vendor specific blocks

Table 3: Slot definitions

2.5.2 Parameter objects

Parameter objects consist of one or more parameters and their values. Furthermore an object can be a part of a block. The relation between these terms will be explained in this chapter.

2.5.2.1 Parameter value

A parameter value is a real value stored in the servo valve.

2.5.2.2 Parameter and their attributes

A Parameter is an abstract representation of a particular parameter value within the object dictionary in a device. Parameters are described in this document in the following tabular form:

Block name							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default

In the parameter list in chapter "10 Object dictionary" you will find a form like the following one in which the attributes are explained in detail:

Slot # Index	**Short name**	**Specification**	**Block, object and parameter name**	**CANopen SDO**	**Data type**	**Access**	**Persistence**	**Value range**	**Default**

<table>
<tr><th>Column name</th><th colspan="2">Meaning</th></tr>
<tr><td>Block name</td><td colspan="2">Describes the family of the object. If the object does not belong to a block, the object name is taken as block name.</td></tr>
<tr><td>Object name</td><td colspan="2">Defined name of the object.</td></tr>
<tr><td>Slot # Index</td><td colspan="2">This pair of values addresses the Profibus DP-V1 device model parameter with slot and index (Slot # Index).</td></tr>
<tr><td>Parameter name</td><td colspan="2">Defined name of the parameter.</td></tr>
<tr><td>CANopen SDO</td><td colspan="2">This pair of values addresses the CANopen parameter with index and sub-index (Index # Subindex).</td></tr>
<tr><td rowspan="7">Data type</td><td colspan="2">Data type of the parameter.</td></tr>
<tr><td>INTn</td><td>Integer with n bits</td></tr>
<tr><td>FLOAT32</td><td>Floating point with 32 bit</td></tr>
<tr><td>char</td><td>Character (ASC II)</td></tr>
<tr><td>STRING</td><td>String of characters</td></tr>
<tr><td>UINTn</td><td>Unsigned integer with n bits</td></tr>
<tr><td>DOMAIN</td><td>Application specific data block</td></tr>
<tr><td rowspan="4">Access</td><td colspan="2">Access permission for the parameter.</td></tr>
<tr><td>rw</td><td>Read and write allowed</td></tr>
<tr><td>wo</td><td>Write only</td></tr>
<tr><td>ro</td><td>Read only</td></tr>
<tr><td>Persistence</td><td colspan="2">Defines whether the parameter can be saved in non-volatile memory. If the persistence is set to "Y", the saved value stays in memory even after the device is turned off. Parameters not marked as persistent ("N") lose their settings after the device is turned off. The parameters with the access type "read only" are marked with "-". This means that the parameter cannot be changed by the user.</td></tr>
<tr><td>Value range</td><td colspan="2">Allowed value range for the object.</td></tr>
<tr><td>Default</td><td colspan="2">Default values:
The default values in this document are firmware preset values. These values can be changed during calibration or set up with model specific parameters during production of the servo valve.
Factory settings:
The factory settings are values which are set up specific to the model during production of the servo valve. These parameters no longer contain the firmware default preset values.
⇨ Chapter "9 Storing / restoring parameters", page 285</td></tr>
<tr><td rowspan="5">Specification</td><td colspan="2">Related (field bus) standard defining the parameter.
Possible entries:</td></tr>
<tr><td>IEC 61158-x-3</td><td>Parameters correspond to IEC 61158-x-3 Profibus DP specification.</td></tr>
<tr><td>PNO Fluid Power Profile</td><td>Parameters correspond to the Profibus DP profile "Fluid Power Technology".</td></tr>
<tr><td>CiA 408</td><td>Parameters correspond to CiA 408 "Device profile for fluid power technology proportional valves and hydrostatic transmissions".</td></tr>
<tr><td>Moog DCV</td><td>Moog defined parameters for digital control valves.</td></tr>
<tr><td>Short name</td><td colspan="2">Unique short name.</td></tr>
</table>

Table 4: Field bus independent attributes

The listed default values contain the firmware preset values and not necessarily the configuration of the delivered servo valve.

2.5.3 Units and prefix parameter

This chapter describes the coding of units and prefix parameters according to CiA 303-2. Some objects provide unit and prefix in the sub-indices 2 and 3 to allow the master controller the correct visualization.

Name of unit	International symbol	Notation index (hex)
none	dimensionless or iR	0x00
meter	m	0x01
second	s	0x03
hertz	Hz	0x20
liter	l or L	0x44
minute (time)	min	0x47
hour	h	0x48
day	d	0x49
year	a	0x4A
bar	bar	0x4E
meter per square second	m/s^2	0x55

Table 5: Unit representation

Prefix	Factor	Symbol	Notation index (hex)
none	10^{-0}		0x00
deci	10^{-1}	d	0xFF
centi	10^{-2}	c	0xFE
milli	10^{-3}	m	0xFD
	10^{-4}		0xFC

Table 6: Prefix representation

2.5.4 Grouping parameters in objects and blocks

An object is a set of parameters which are grouped by an object name. The objects are grouped in blocks.

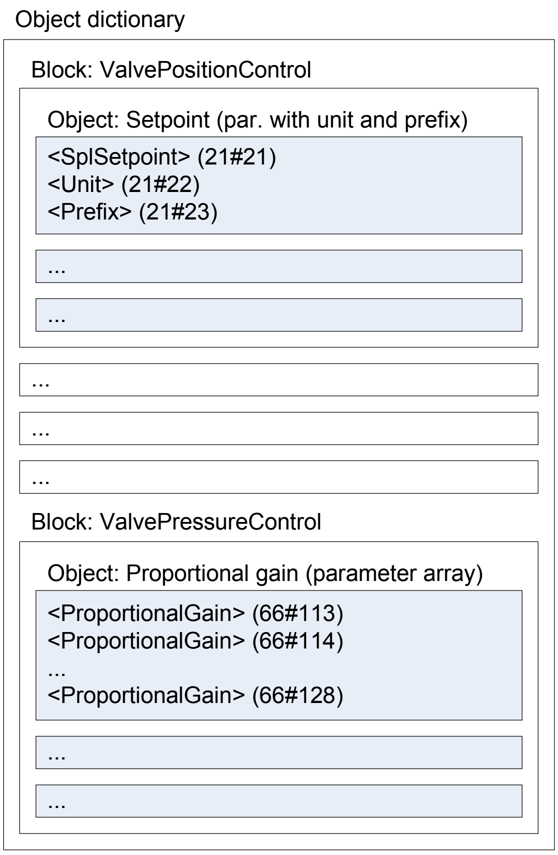


Figure 5: Grouping parameters in objects and blocks

2.6 Information objects

2.6.1 Object 0#21: Ident number

This parameter contains the Profibus identity number which is used to identify the GSD file for the Profibus device.

Profibus							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#21	IdentNumber	0x4460#0	UINT16	ro	-	UINT16	0x07F4

2.6.2 Object 64#1: Bit rate

This parameter contains the effective Profibus bit rate. The bit rate is set by the master.

Profibus							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#1	Bitrate	0x4447#0	UINT32	ro	-	0…12000000	None

2.6.3 Object 0#36: Error code

This parameter contains the Profibus error code.

Profibus							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#36	ErrorCode	0x4461#0	UINT16	ro	-	UINT16	None

2.6.4 Object 64#12: DP-V1 status

If this parameter is set to 1, one or more bits (bit 3…7 of byte 7 or any bit of byte 8 and 9) of the user parameter data telegram (SAP 61) are switched on. These bits indicate that at least one DPV1 functionality is switched on.

Profibus							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#12	DPV1Status	0x444B#0	UINT8	ro	-	0…1	None

Value description

<DPV1Status>	
Value	
0	Servo valve Profibus interface does not use any DP-V1 functionality.
1	Servo valve Profibus interface uses at least one DP-V1 functionality.

2.6.5 Object 64#11: VPC3+b Status

This parameter contains the Profibus chip status register.

Profibus							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#11	VPC3+b_Status	0x4448#0	UINT16	ro	-	UINT16	None

Value description

<VPC3+b_Status>	
Bit	**Description**
0	Offline/Passive_Idle state: 0. VPC3+C is in Offline state. 1: VPC3+C is in Passive_Idle state.
2	Status of the Diagnosis buffer: 0: The Diagnosis buffer is read from the DP-Master. 1: The Diagnosis buffer is not read from the DP-Master yet.
4…5	State of the Network State Machine: 00: WAIT-PRM state 01: WAIT-CFG state 10: DATA-EXCH state 11: Not possible
6…7	State of the Watchdog State Machine: 00: BAUD_SEARCH state 01: BAUD_CONTROL state 10: DP_CONTROL state 11: Not possible
8…11	The baud rate found by VPC3+D: 0000: 12.00 Mbit/s 0001: 6.00 Mbit/s 0010: 3.00 Mbit/s 0011: 1.50 Mbit/s 0100: 500.00 Kbit/s 0101: 187.50 Kbit/s 0111: 45.45 Kbit/s 1000: 19.20 Kbit/s 1001: 9.60 Kbit/s 1111: After reset and during baud rate search Rest: Not possible
12…15	Release number for VPC3+: 0000: Step A 1011: Step B 1100: Step C 1101: Step D Rest: Not possible

2.7 Slave Network state machine (DP-V0)

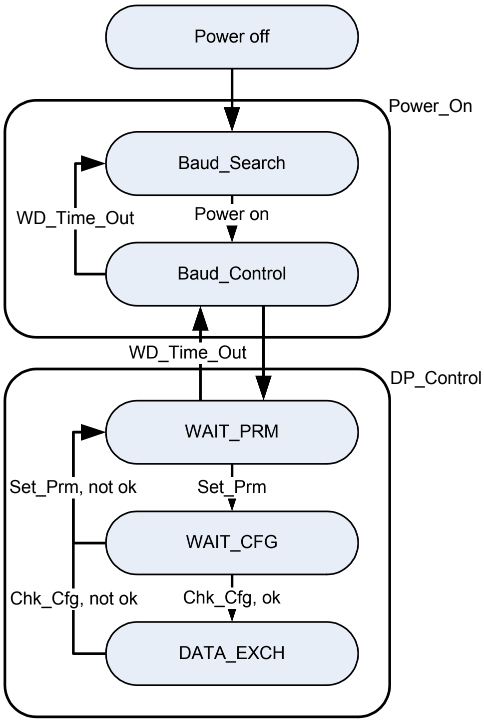


Figure 6: Grouping parameters in objects and blocks

Start up routine:
The startup sequence is:
1. Request diagnosis
2. Change station address (if necessary)
3. Set parameterization
4. Check configuration
5. Request diagnosis (to check the preceding commands)
6. Data exchange

The network state machine state can be read from the parameter <VPC3+b_Status> (64#11) bit 4…7.
⇨ Chapter "2.6.5 Object 64#11: VPC3+b Status", page 16

Power_On:
Only in Power_On state (Baud_Search and Baud_Control) the slave accepts an address change request.

Baud_Search:
The servo valve is capable of detecting the Profibus baud rate set by the master. If the electronics is in the Baud_Search state, it cannot accept any messages. The electronics searches for the baud rate by checking different baud rates, trying to detect a correct telegram. After identifying the baud rate the controller switches to Baud_Control state.

Baud_Control:
The detected baud rate is constantly monitored in the Baud_Control state. Each error free detected telegram to the own station address resets the internal watchdog. The watchdog timer can be set by the master. If the watchdog expires, the servo valve drops to the Baud_Search state and starts searching the baud rate again.

DP_CONTROL:

WAIT_PRM:
After the startup phase, the slave expects a parameter telegram which defines the behavior of the slave. The parameter telegram contains information about the indent number, the sync/freeze capability, master address and watchdog time.

WAIT_CFG:
In this state the slave waits for the configuration telegram. The configuration telegram defines the configuration of inputs and outputs. The master transmits the IO configuration to the servo valve. After receiving the Chk_Cfg telegram the servo valve compares the configuration with its own configuration. If the configurations match, the servo valve will acknowledge this by setting the corresponding bit in the diagnosis state and enter the DATA_EXCH state automatically.

DATA_EXCH:
The servo valve is in DATA_EXCH state and exchanges cyclic data with the master.

2.8 Setting the slave address (DP-V0)

The factory setting for the Profibus slave address is 126. There are three ways to change the slave address:

- Change the slave address by the Master
- Change the slave address with the Moog Valve and Pump Configuration Software
- Change the slave address via the parameter channel

All servo valves have the address 126 configured as the factory setting. Therefore, it is recommended to install the servo valves one by one and configure the address, or switch them on one after another. This allows the master to establish a peer to peer connection to the servo valve in order to configure the address. If more than one new servo valve is attached to the bus, the master cannot identify that there is more than one unconfigured slave with address 126. It is also possible to pre-configure the servo valve with the "Moog Valve and Pump Configuration Software" before connecting it to the Profibus.

2.8.1 Setting the slave address by the master

The slave address can be changed by the master via the Set_Slave_Adr telegram. The slave address will be changed directly in the Profibus chip without restart and will be saved for the next restart.

2.8.2 Setting the slave address with the "Moog Valve and Pump Configuration Software" or by using the parameter channel

After changing the value of the parameter <ModuleIdentifier> (64#0), the new slave address must be saved by the "store/restore" parameter. The new address becomes effective after the next boot up of the servo valve. The effective slave address can be read from the parameter <ActualModuleIdentifier> (64#33).

⇨ Chapter "9 Storing / restoring parameters", page 285

2.8.2.1 Object 64#0: Module identifier

This parameter contains the preset Profibus slave address. The parameter can be changed and stored to set the Profibus slave address which will become effective as <ActualModuleIdentifier> (64#33) after the next power on.

Profibus							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#0	ModuleIdentifier	0x4446#0	UINT8	rw	Y	1…126	125

2.8.2.2 Object 64#33: Actual module identifier

This parameter contains the effective Profibus slave address.

Profibus							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#33	ActualModuleIdentifier	0x4449#0	UINT8	rw	Y	1…126	125

2.9 Cyclic master slave communication (DP-V0)

2.9.1 Configuration

Process data is exchanged between master and the slave devices by using cyclic telegrams. The data length and the transmission type of the telegram can be configured. This configuration is stored in the slave devices configuration file named GSD file. This file is created by the slave manufacturer and must be provided to the Profibus master. The chosen servo valve module in the master must match the chosen telegram in the servo valve. The servo valve telegram can be selected with the parameter <TelegramSelection> (0#46). The result of the Profibus configuration is saved in the parameter <TelegramConfgurationBytes> (64#3…9).

2.9.1.1 Object 0#46: Telegram selection

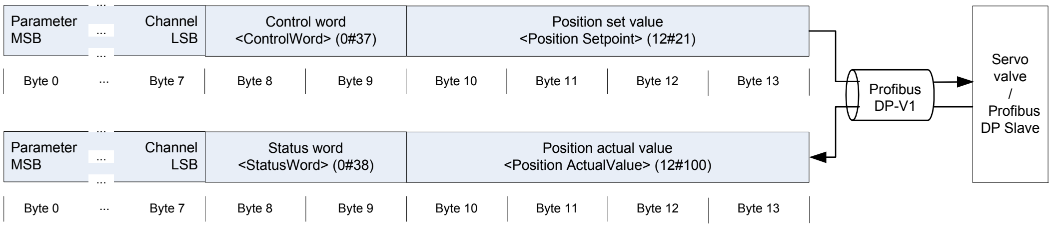
The parameter can be changed and stored to set the Profibus telegram which will become effective after the next power on.

Profibus							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#46	TelegramSelection	0x444F#0	UINT8	rw	Y	UINT8	3

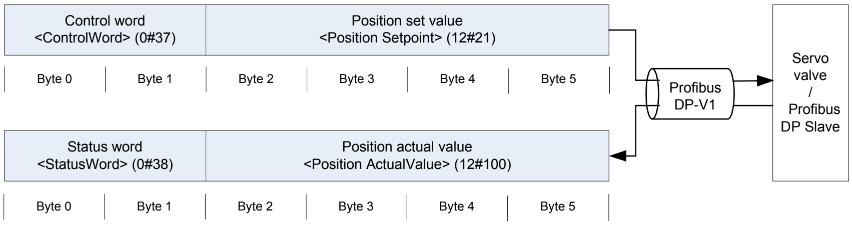
Value description

<TelegramSelection>	**Module name (content of the <TelegramConfigurationBytes> (64#3…9))**	**I/O telegram content**
1	Std. Tel. 1 | pos + Par.Chn. (0xF3,0xE2,0xD2)	Parameter channel Control word Axis position (32 bit)
2	Std. Tel. 2 | pos (0xE2,0xD2)	Control word Axis position (32 bit)
3	Std. Tel. 3 | Q + Par.Chn. (0xF3, 0xE1, 0xD1)	Parameter channel Control word Spool position
4	Std. Tel. 4 | Q (0xE1, 0xD1)	Control word Spool position
5	Std. Tel. 5 | p/Q + Par.Chn. (0xF3, 0xE2, 0xD2)	Parameter channel Control word Pressure Spool position
6	Std. Tel. 6 | p/Q (0xE2, 0xD2)	Control word Pressure Spool position
100	MOOG Tel. 100 | p + Par.Chn. (0xF3, 0xE1, 0xD1)	Parameter channel Control word Pressure
101	MOOG Tel. 101 | p (0xE1, 0xD1)	Control word Pressure
255	MOOG Tel. 255 | p/Q + pos + Par.Chn. (0xF3,0xE4,0xD4)	Parameter channel Control word Pressure Spool position Axis position

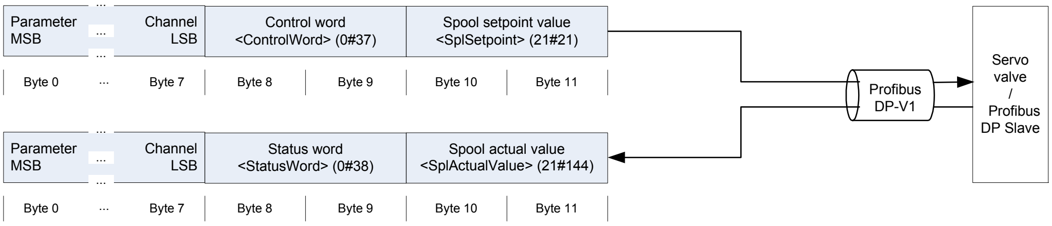
Module 1: <TelegramSelection> (0#46)=1



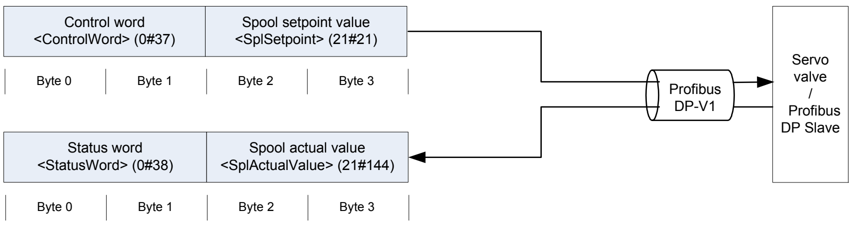
Module 2: <TelegramSelection> (0#46)=2



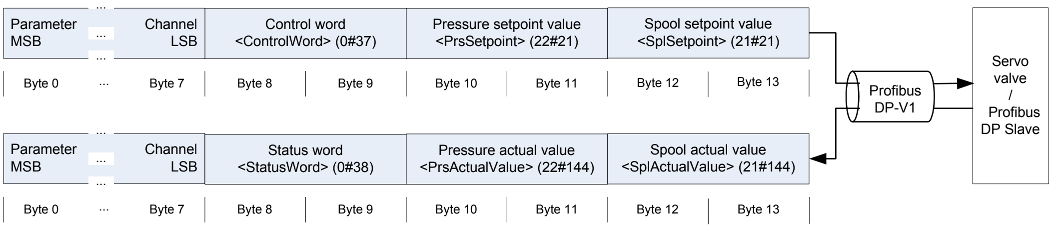
Module 3: <TelegramSelection> (0#46)=3



Module 4: <TelegramSelection> (0#46)=4



Module 5: <TelegramSelection> (0#46)=5



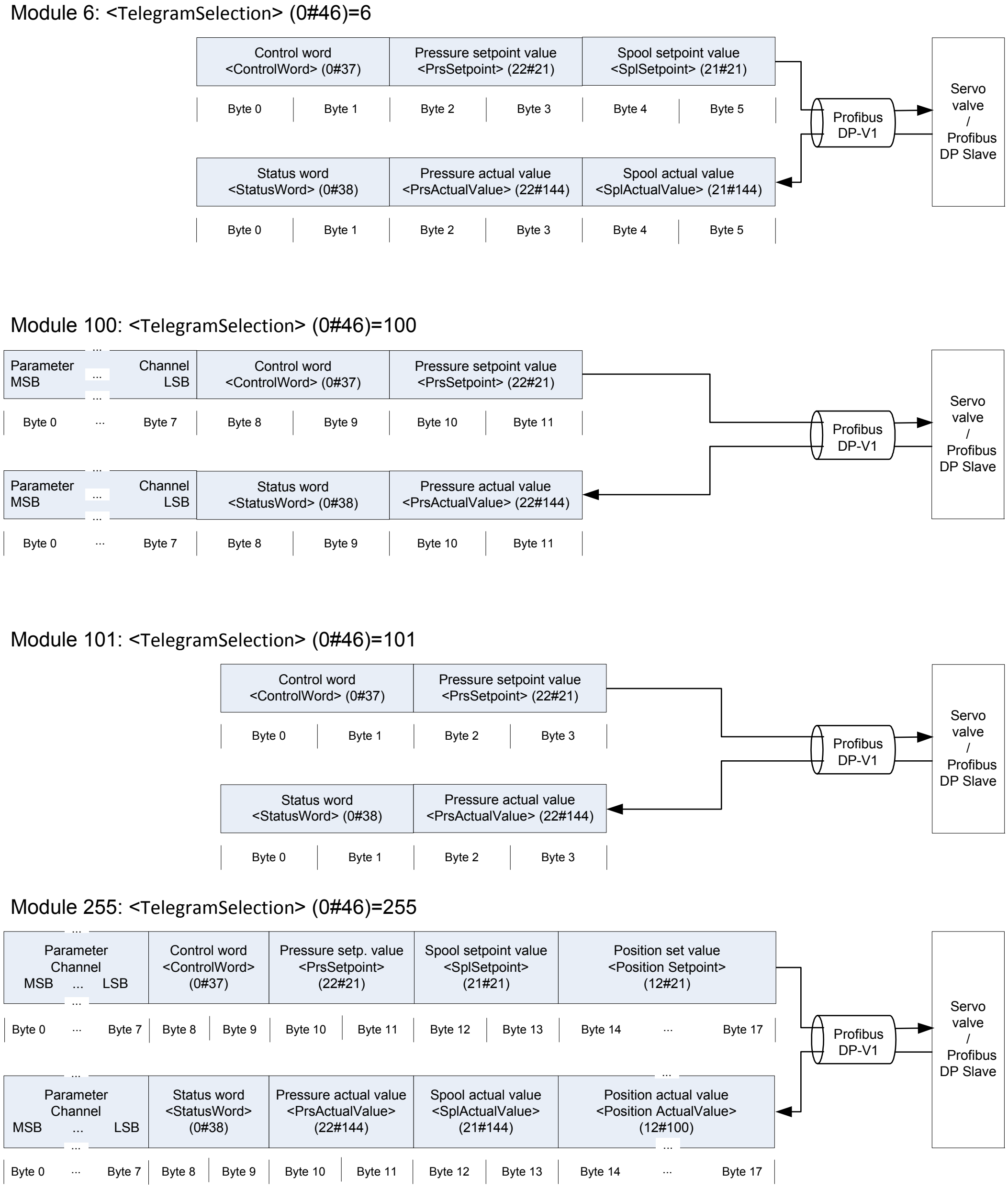


Figure 7: Selection of servo valve telegram

2.9.1.2 Object 64#2: Telegram configuration length

This parameter contains the number of the configured modules in the cyclic telegram. The servo valve can set up maximum seven cyclic transfer data modules defined in the parameter <TelegramConfigurationBytes> (64#3…9).

Profibus							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#2	TelegramConfigurationLength	0x444D#0	UINT8	ro	-	0…7	None

2.9.1.3 Object 64#3…9: Telegram configuration bytes

This parameter contains the effective Profibus cyclic data configuration.

Profibus							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#3…9	TelegramConfigurationBytes	0x444E#1…7	UINT8	ro	-	UINT8	None

Value description

<TelegramConfigurationBytes>	
Bit	**Description**
0…3	Length of data: 0x00: 1 byte/word 0x0F: 16 byte/word
4…5	Input / output: 00: Special format 01: Input (cyclic process data) 10: Output (cyclic process data) 11: Input and output (acyclic parameter channel)
6	Data type: 0: Byte 1: Word
7	Consistence: 0: Consistent over data type 1: Consistent over whole data length

2.9.2 Master Watchdog

The watchdog function is implemented in the master. If the master does not receive a requested telegram within a defined watchdog time, the master responds with a watchdog error. The watchdog time can be set within the master.

2.9.3 Slave Watchdog

If the slave loses the connection to the Profibus master, the servo valve generates the fault code number 118 (General Profibus Error).

⇨ Chapter "8.1.2 Possible fault codes", page 258

2.9.4 Global control

The global control sends commands to all slaves in the same group. The group identifier of each slave is set by the master (parameter telegram). The following commands are supported:

- Freeze
- Unfreeze
- Synch
- Unsync

With these global control commands it is possible to synchronize the inputs and outputs for all slaves in a group.

2.9.4.1 Object 64#34: Last global control telegram

The last global control command received by the servo valve is saved in this parameter.

Profibus							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#34	LastGlobalControlTelegram	0x4457#0	UINT8	ro	-	UINT8	None

Value description

<LastGlobalControlTelegram>	
Bit	**Description**
0	Reserved
1	Clear data (not supported)
2	Unfreeze
3	Freeze
4	Unsync
5	Synch
6	Reserved
7	Reserved

2.9.4.2 Freeze / Unfreeze

When the master sends a "freeze" global control broadcast command to a slave group, all slaves in this group buffer the input data virtually at the same time. In the next bus cycles (one per slave), the master can read the frozen input data of all slaves in the group. After all data is read, the master sends the global control broadcast command "unfreeze" to the group. Now the slaves in the group release their input buffer and update it cyclically again. By using this procedure all input values can be read synchronously.

2.9.4.3 Sync / Unsync

When the master sends a "sync" broadcast command to a slave group, all slaves in this group buffer the output data and hold the buffered data. So the outputs maintain their values. Now the master can send new data for the outputs (one telegram per slave). After all data is written, the master sends the broadcast command "unsync". All slaves in the group change the output buffer almost simultaneously. So all outputs switch the values synchronously (output from master = input to slave).

2.9.5 Parameter channel (DP-V0 / DP-V1)

If a parameter channel is defined between master (module with parameter channel) and slave (telegram with parameter channel defined with the parameter <TelegramSelection> (0#46)), a cyclic parameter exchange is possible.

2.9.5.1 Structure of the parameter channel

The coding of the parameter channel is in accordance with the parameter channel description in the Device Profile Fluid Power.

The following figure shows the input and output structure of the parameter channel.

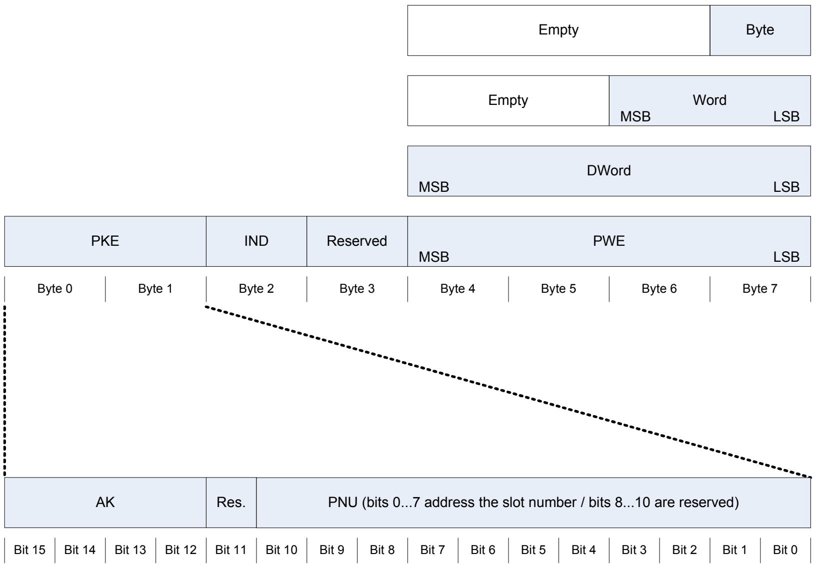


Figure 8: Structure of the parameter channel

Abbreviations	Meaning
PKE	Parameter signature value
IND	Index within a slot (slot number is coded in PKW -> PNU)
PWE	Process value
AK	Instruction / response signature
PNU	Parameter number (bits 0…10) Bits 0…7 code the slot number of the function block Bits 8…10 are reserved

2.9.5.2 Read a parameter value

To initiate a read request, write the slot (PNU) and the index (IND) of the desired parameter to the parameter channel structure and set the instruction (AK) to 0001b.

Read instruction (AK)	
Value	**Description**
0000b	No request
0001b	Parameter value read

In the case of a communication error, the signature (AK) contains the value 0111b and the bytes 6…7 (PWE) hold the error code.
⇨ Chapter "2.9.5.4 Parameter channel read / write errors", page 26

Otherwise the signature (AK) acknowledges the command and the requested value (PWE) can be processed. If the read parameter value is one byte, it is transferred in byte 7 of PWE, if the parameter value consists of two bytes (word), it is transferred using the bytes 6 and 7 of PWE, if the parameter value consists of four bytes (double word), it is transferred using the bytes 4 to 7 of PWE. The parameter value (PWE) is read during every cycle from the servo valve as long as the instruction (AK), slot (PNU) and index (IND) do not change.

Response signature (AK)	
Signature	**Description**
0000_b	No response
0001_b	Parameter value transmitted (word)
0010_b	Parameter value transmitted (double word)
0111_b	Instruction not processable (error code)
1011_b	Parameter value transmitted (byte)

2.9.5.3 Write a parameter value

To initiate a write request, write the slot (PNU) and the index (IND) of the desired parameter to the parameter channel structure. If a byte should be written to the servo valve, the byte parameter must be written to byte 7. If a word should be written to the servo valve, the word parameter must be written to the bytes 6…7. If a double word should be written to the servo valve, the double word parameter must be written to the bytes 4…7. To write the parameter value, set the instruction (AK) to one of the following values.

Write instruction (AK)	
Signature	**Description**
0000_b	No request
0010_b	Parameter value write (word)
0011_b	Parameter value write (double word)
1010_b	Parameter value write (byte)

In case of a communication error, the signature (AK) contains the value 0111b and the bytes 6…7 hold the error code.
⇨ Chapter "2.9.5.4 Parameter channel read / write errors", page 26

Otherwise the signature acknowledges the requested write command with one of the following values.

Response signature (AK)	
Signature	**Description**
0000_b	No response
0001_b	Parameter value transmitted (word)
0010_b	Parameter value transmitted (double word)
0111_b	Instruction not processable (error code)
1011_b	Parameter value transmitted (byte)

2.9.5.4 Parameter channel read / write errors

If a write or read error occurs, an error code is set in the bytes 6…7 (PWE) in the parameter channel structure. The following table shows the possible error codes.

Error code (byte 6…7 of the parameter channel structure)	
Error code	**Description**
0	Undefined index
1	Parameter not changeable
2	Lower or upper value range limit overflow
3	Sub-index error
4	No array
5	Data type error
6	Setting not allowed (only resettable)
7	Description element not changeable
8	Reserved
9	Reserved
10	Access group error
11	No operation sovereignty
12	Password error
13	Text not readable in cyclic data transfer
14	Name not readable in cyclic data transfer
15	No text array existent
16	Reserved
17	Instruction not processable caused by bad operation state
18	Other errors
19	Data not readable in cyclic error
20…100	Reserved for all PNO profiles
101…200	Reserved for future profile extensions
210…255	Vendor specific

Table 7: Error codes of the parameter channel structure

2.10 Diagnostic (DP-V0)

In the Profibus DP-V0 standard three diagnostic telegrams are defined:

- Module diagnostic (not implemented)
- Channel diagnostic (not implemented)
- Device diagnostic

The diagnostic data can be read from the Profibus DP field bus interface in the master. The way to retrieve this diagnostic data depends on the master. For information how to retrieve the device diagnostic data refer to the master documentation.

2.10.1 Device diagnostic data structure

The diagnostic data structure is a combination of a Profibus DP-V0 standard part (the first 6 bytes) and one or more diagnostic message blocks. The servo valve supports only the device diagnostic with a Moog specific 9 byte diagnostic message.

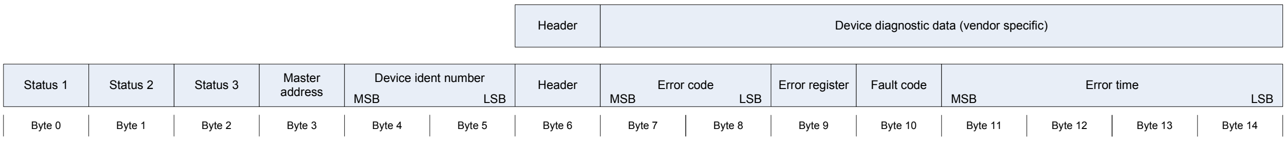


Figure 9: Device diagnostic data structure

2.10.1.1 Status bytes

The first three bytes contain status information about the slave.

Status 1 (Byte 0)	
Bit	Description
0	Station_Non_Existent. Diagnostic station does not exist. (set by master)
1	Station_Not_ready. Slave not ready for data exchange.
2	Cfg_Fault. Configuration data does not match.
3	Ext_Diag (1 for device diagnostic). Slave has external diagnosis.
4	Not_Supported
5	Invalid_Slave_Response
6	Prm_Fault. Slave not ready for data exchange.
7	Master_lock. Slave was configured by other master. (set by master)

Status 2 (Byte 1)	
Bit	**Description**
0	Prm_Req. Slave requires new configuration.
1	Stat_Diag. Statistic diagnosis.
2	Fixed to 1
3	WD_On. Watchdog active.
4	Freeze_Mode. Freeze command was received.
5	Sync_Mode. Sync command was received.
6	Reserved
7	Deactivated (set by master)

Status 3 (Byte 2)	
Bit	**Description**
0	Reserved
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Ext_Diag_Overflow

2.10.1.2 Master address byte

If the slave is parameterized successfully, the master address byte contains the master address. Otherwise the master address byte holds the default value 255.

Master address (Byte 3)	
Value	**Description**
0…125	Valid master address
255	Slave not parameterized

2.10.1.3 Device indent number word

The Profibus indent number is used to identify the appropriate GSD file for the servo valve.

⇨ Chapter "2.6.1 Object 0#21: Ident number", page 14

Device indent number (Bytes 4…5)		
Byte	**Value**	**Description**
0	0x07	Ident number high byte
1	0xF4	Ident number low byte

2.10.1.4 Device diagnostic data

The diagnostic message contains a header byte with the diagnostic type selector and the message length.

Header (Byte 6)		
Bit	**Value**	**Description**
0…1	00_b	Diagnostic type: 00: Device diagnostic 01: Module diagnostic (not used) 10: Channel diagnostic (not used)
2…7	001001_b	Length of device diagnostic data (8 bytes data + 1 byte header = 9 bytes)

The error codes and the error register comply with the Device Profile Fluid Power.

⇨ Chapter "8.1.4 Error codes depending on fault codes", page 262
⇨ Chapter "8.1.6 Error register", page 265

Error code (Bytes 7…8)	
Byte	**Description**
0	Device profile error code high byte
1	Device profile error code low byte

Error register (Byte 9)	
Bit	**Description**
0	Generic error (any error)
1	Current error
2	Voltage error
3	Temperature error
4	Communication error
5	Reserved
6	Reserved
7	Reserved

The fault codes are vendor specific.

⇨ Chapter "8.1.2 Possible fault codes", page 258

Fault code (Byte 10)	
Byte	**Description**
0	Moog fault code

The error time stamp contains the time in minutes when the error occurred since the last servo valve reset.

Error time in minutes since power on (Bytes 11…14)	
Byte	**Description**
0	Error time high byte
1	
2	
3	Error time low byte

2.11 Acyclic master slave communication (DP-V1)

The Profibus DP extension version 1 (DP-V1) extends the Profibus DP capability to allow exchange of parameter data acyclic between master and slave. There are two master to slave communications defined in the Profibus DP-V1 standard:

- Acyclic data exchange with a class 1 master (MSAC_C1).
  The class 1 master is the control master.
- Acyclic data exchange with a class 2 master (MSAC_C2) (not supported).
  The class 2 master is for diagnostic purpose.

The acyclic data exchange with a class 1 master (MSAC_C1) is used to implement a parameter channel. This parameter channel is different to the parameter channel defined in the Device Profile Fluid Power. In the following table the differences between the MSAC_C1 (Master Slave Acyclic Communication Class 1) and the DP-V0 (Decentralized Periphery Version 0) cyclic parameter channel are explained:

	DP-V0/Parameter channel	**DP-V1/MSAC_C1**
Reference	Device Profile Fluid Power	IEC 61158
Communication type	Cyclic	Acyclic
Implementation	The Parameter channel is a 8 byte process data input/output structure. This structure can be mapped to a PLC input and output variable.	This parameter channel is implemented as separate Profibus telegram. The data transfer between PLC and the Profibus slave depends on the Profibus master implementation.
Data length	One parameter with maximum 4 byte data referenced by slot and index.	Block of parameters referenced by slot, index and length in bytes up to 4…244 bytes.

The MSAC_C1 data can be read from and written to the Profibus DP master. The way to transfer data depends on the master. For information on how to access the MSAC_C1 data channel refer to the master documentation.

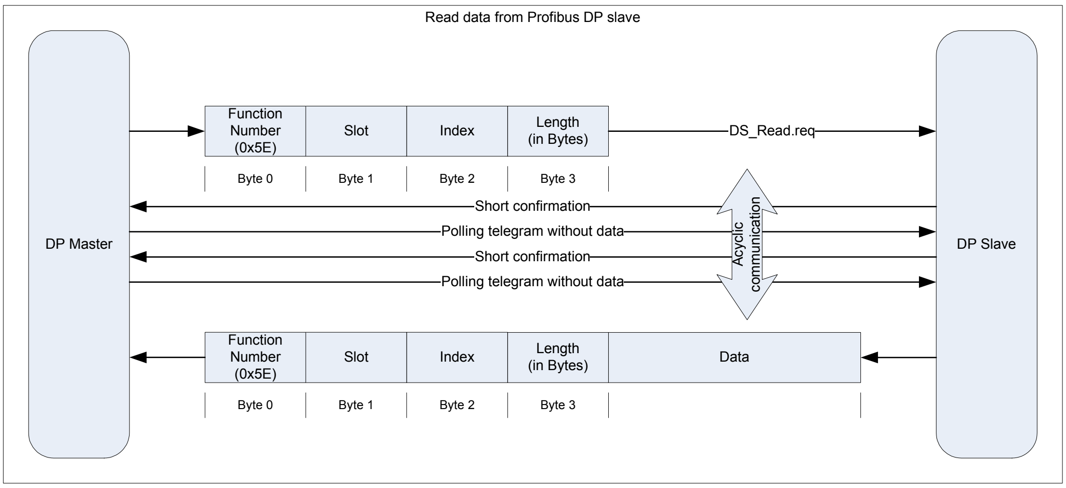


Figure 10: Master slave communication (DP-V1)

2.12 The GSD file

The GSD file contains data for configuration of the Profibus DP servo valve and must be provided to the Profibus master. The GSD file is provided by Moog or can be downloaded from the Moog website http://www.moogsoftwaredownload.com/. Search for "gsd" under "Configuration Files for BUS master".

3 Device structure

3.1 Overview

The servo valve functionality is based on the Device Profile Fluid Power. This profile defines the device functionality and the object dictionary of the parameters. The following figure shows the general architecture.

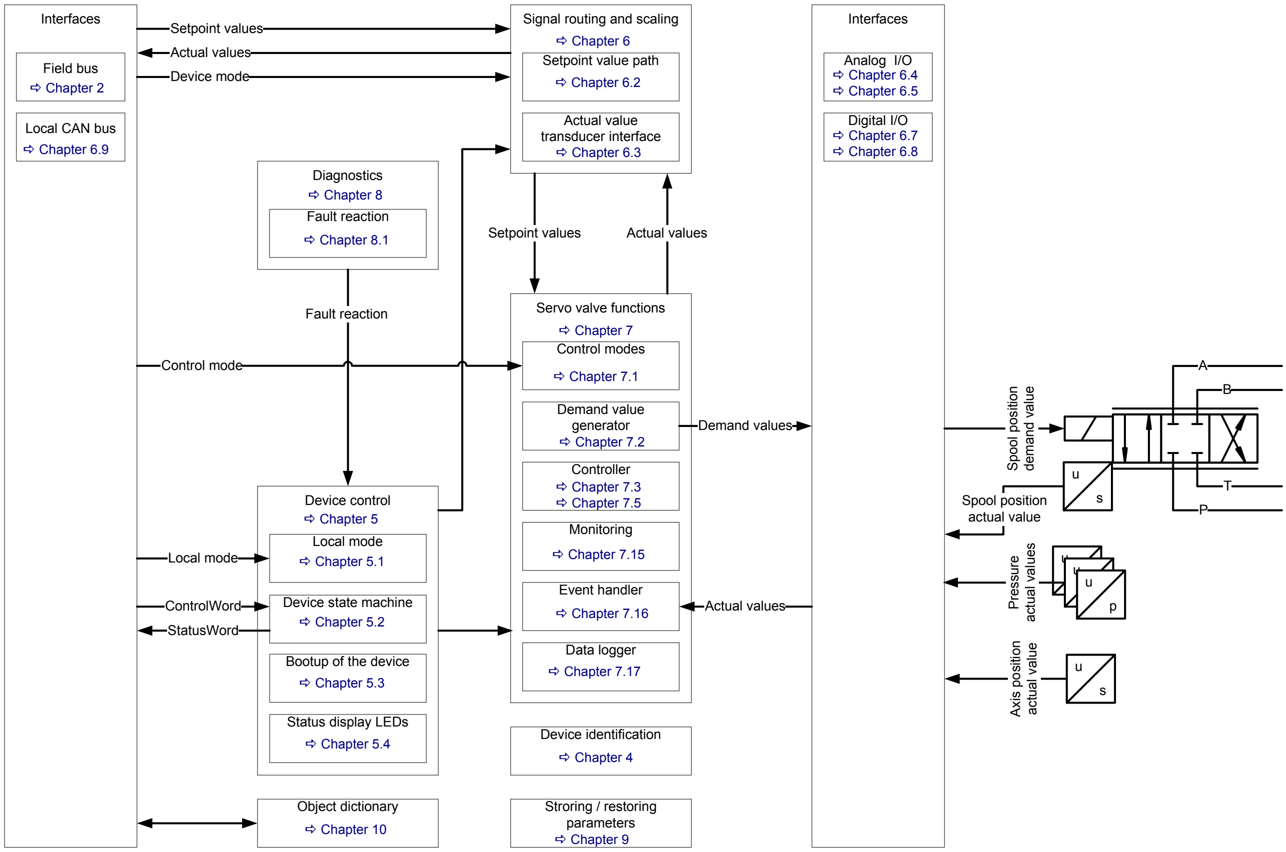


Figure 11: Device structure

3.1.1 Device identification

For the device identification (model number, serial number, device capability …) a special set of parameters is defined.

3.1.2 Device control

The device control block controls all functions of the device and contains the device state machine by which the device functions can be activated or deactivated.

3.1.3 Signal routing and scaling

This block conditions the raw input signals for the signal processing and maps these to the internal servo valve functions. This mapping depends among other things on the <DeviceMode> (0#39) and the device state machine state.

3.1.4 Servo valve functions

All signals in the servo valve are processed by the internal 'servo valve' function block. The demand value generator prepares the setpoint value before it is used as input to the controller. Depending on the valve, the controller can control spool position, or pressure or a combination of both or the position, or velocity or the force of an axis. The pressure controller can be tuned by the end-user to optimize the specific closed loop response. The control error (the difference between setpoint and feedback signals) can be monitored.

3.1.5 Diagnostics

The diagnostic module detects faults and initiates a response according to how the module parameters have been configured by the user. For example, this can result in an error message being sent to the master or a change of state in the device state machine.

3.1.6 Storing / restoring parameters

Save and load several servo valve parameters to the internal non volatile memory of the device.

3.1.7 Object dictionary

The object dictionary holds all servo valve parameters. These parameters can be read or written using the field bus or the local CAN interface and the Moog Valve and Pump Configuration Software.

3.2 Device controller structure

The following figure shows the device structure with focus on the signal flow. Depending on the <ControlMode> (0#40) only a subset of the device structure may be used.

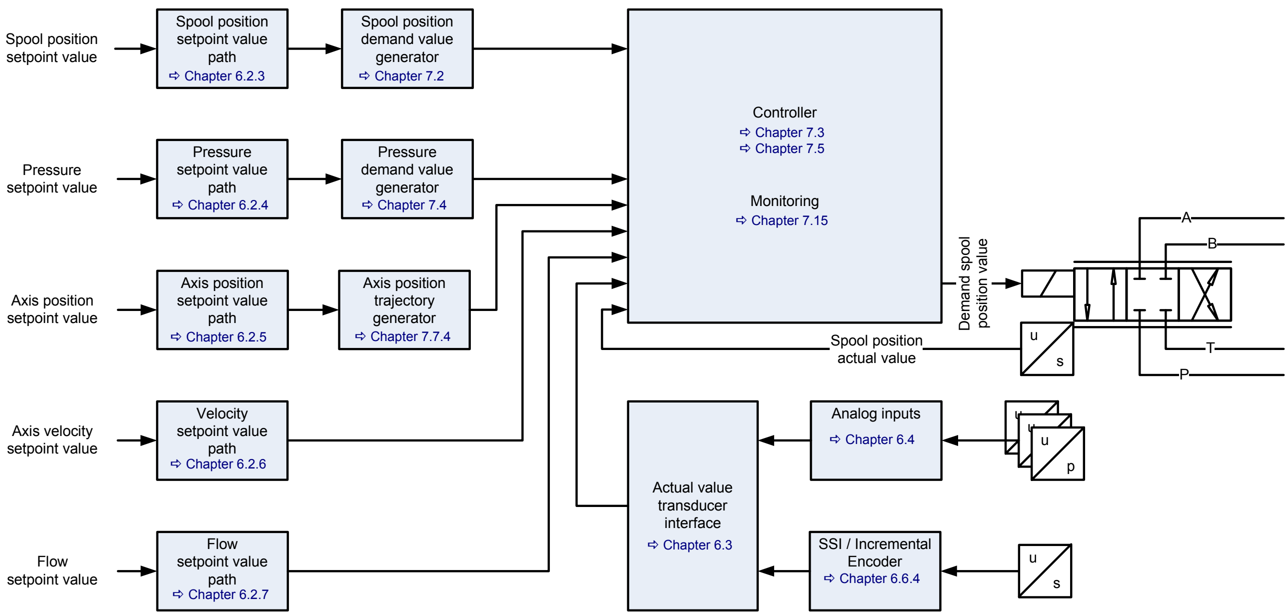


Figure 12: Device controller structure

4 Device identification

The device (servo valve) has parameters that both identify the device and configure the device for operation with the specific machine. The following chapter includes the descriptions of the parameters providing this information.

4.1 Objects of the PROFIBUS-DP Profile for Fluid Power Technology defined by PNO

4.1.1 Object 0#26: Manufacturer hardware version

This parameter indicates the current hardware version of the servo valve electronics.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#26	ManufacturerHardwareVersion	0x1009#0	STRING	ro	-	None	""

4.1.2 Object 0#24: Manufacturer software version

This parameter indicates the current software version of the servo valve.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#24	ManufacturerSoftwareVersion	0x100A#0	STRING	ro	-	None	Device-specific value

4.1.3 Object 0#22: Version

This parameter contains the model range of the Moog servo valve.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#22	DeviceVersion	0x6050#0	STRING	ro	-	None	""

4.1.4 Object 0#32: Code number

The user can use this parameter to set a user defined code number.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#32	CodeNumber	0x6051#0	UINT16	rw	-	UINT16	0

4.1.5 Object 0#28: Serial number

This parameter indicates the serial number of the Moog servo valve.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#28	SerialNumber	0x6052#0	STRING	ro	-	None	""

4.1.6 Object 0#33: Description

The user can use this parameter to set a user defined description.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#33	Description	0x6053#0	STRING	rw	Y	None	""

4.1.7 Object 0#30: Model description

This parameter contains the model number of the Moog digital servo valve.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#30	ModelDescription	0x6054#0	STRING	ro	-	None	""

4.1.8 Object 0#20: Vendor name

This parameter indicates the name of the device vendor.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#20	VendorName	0x6057#0	STRING	ro	-	None	"MOOG, Hanns-Klemm-Strasse 28, D-71034 Boeblin-gen, Germany"

4.1.9 Object 0#50: Capability

This object provides information on the capabilities of the used device, e.g. the control modes. The value depends on the ordered functionality (type designator position 16).

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#50	Capability	0x605F#0	UINT32	ro	-	UINT32	0x3F3F0000

Value description

<Capability>																	
Description	Mod	Servo valve or pump application							Drive application								Reserved
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15…0
	MSB																LSB

Table 8: Possible values of parameter <Capability> (0#50)

Bit	**Description**
0…15	Reserved
16	Supports drive
17	Supports control mode drive open loop movement
18	Supports control mode drive velocity control
19	Supports control mode force control
20	Supports control mode drive position control
21	Supports control mode flow control
22, 23	Reserved
24	Supports servo valve or pump
25	Supports control mode spool position open loop
26	Supports control mode spool position closed loop
27	Supports control mode pressure open loop
28	Supports control mode pressure closed loop
29	Supports control mode p/Q closed loop
30	Reserved
31	Supports modular device according CiA 301

Table 9: Bit values of parameter <Capability> (0#50)

4.2 Vendor-specific objects

4.2.1 Object 64#13…16: Identity object

These parameters represent a worldwide unique identification of any Profibus DP/CANopen slave device.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#13	VendorID	0x1018#1	UINT32	ro	-	UINT32	40
64#14	ProductCode	0x1018#2	UINT32	ro	-	UINT32	0
64#15	RevisionNumber	0x1018#3	UINT32	ro	-	UINT32	0
64#16	SerialNumber	0x1018#4	UINT32	ro	-	UINT32	0

The identification object (VendorID, ProductCode, RevisionNumber, SerialNumber) is printed on the name plate of the servo valve as shown in the following figure.

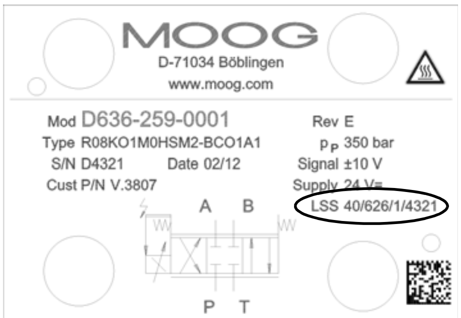


Figure 13: Name plate of the device with identification object address

Value description

Sub-index	**Parameter**	**Meaning**
1	<VendorId>	Unique vendor ID, 0x28 reserved for Moog.
2	<ProductCode>	Product code of the Moog servo valve. Each valve model number has its own product code.
3	<RevisionNumber>	Revision number of the Moog servo valve.
4	<SerialNumber>	Serial number of the Moog servo valve (digits without leading character) as on the name plate. ⇨ Chapter "4.1.5 Object 0#28: Serial number", page 34

Table 10: Possible values of parameter Identity object (64#1…4)

<VendorId>				
Description	Department		Company (0x000028)	
Bit	31	24	23	0
	MSB			LSB

4.2.2 Object 72#1: Model URL

This parameter holds the Internet address where additional information about the device is available.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#1	ModelURL	0x6055#0	STRING	ro	-	None	"www.moog.com"

4.2.3 Object 72#2: Parameter set code

The user can use this parameter to set a user defined parameter configuration identification number.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#2	ParameterSetCode	0x6056#0	UINT8	rw	Y	0…254	0

5 Device control

The device control contains a device state machine (DSM) which activates or deactivates the servo valve. The states of the DSM can be changed externally by the <ControlWord> (0#37) which can be set via bus using a PDO or locally by the parameter value <LocalControlWord> (0#206). It is possible to configure a power on delay for the communication. The status LEDs display the current network and device states.

5.1 Local mode

The source of the control word, acting on the device state machine, is defined by the parameter <Local> (0#41) as shown in the following figure.

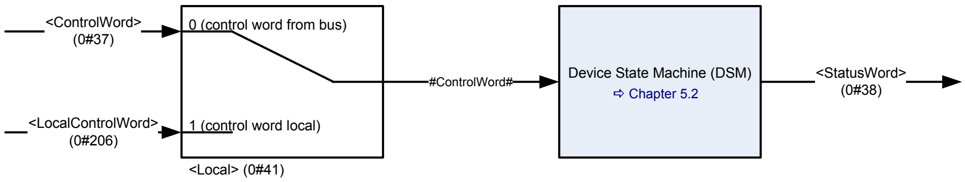


Figure 14: Local mode

5.1.1 Object 0#41: Local

By writing the value 1 to this parameter, the <LocalControlWord> (0#206) is used as input signal #ControlWord# for the device state machine (DSM). Setting the <Local> (0#41) parameter to 0, the <ControlWord> (0#37) is used as input signal #ControlWord#.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#41	Local	0x604F#0	INT8	rw	Y	0…1	0

Value description

<Local>	Description
0	DSM controlled via bus by the <ControlWord> (0#37). The <ControlWord> (0#37) is used as input signal #ControlWord# for the DSM.
1	DSM controlled by the <LocalControlWord> (0#206). The <LocalControlWord> (0#206) is used as input signal #ControlWord# for the DSM.

Table 11: Possible values of parameter <Local> (0#41)

The actual setting of the <Local> (0#41) parameter is indicated in bit 4 of the <StatusWord> (0#38).

Bit 4	Description
1	<LocalControlWord> is active.
0	<ControlWord> is active.

⇨ Chapter "5.2.3 Object 0#38: Status word", page 48

5.1.2 Object 0#37: Control word

The functionality of the <LocalControlWord> (0#206) and the <ControlWord> needs to be distinguished. If the parameter <Local> (0#41) is set, the DSM input #ControlWord# comes from the local parameter <LocalControlWord> (0#206) otherwise it comes from <ControlWord> (0#37). This bit-coded parameter controls the DSM states.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#37	ControlWord	0x6040#0	UINT16	rw	N	UINT16	None

Value description

<ControlWord>				
Bit	**Description**	**<Control Mode> (0#40) is set to 1…4, 7…14**	**<Control Mode> (0#40) is set to 5**	**Specification**
0	Bit Disabled (D)			CiA 408
1	Bit Hold (H)			
2	Bit Active (M)			
3	Bit Reset Faults (R)			
4…7	Reserved			
8	<ControlMode> (0#40) specific	Reserved	Enable pressure controller	CiA 408
9	Axis install mode positive direction			MOOG DCV
10	Axis install mode negative direction			MOOG DCV
11…14	Reserved			
15	Ramp stop			Moog DCV

Table 12: Possible values of parameter <ControlWord> (0#37)

Bits 0, 1, 2, 3: 'DISABLED', 'HOLD', 'ACTIVE', 'RESET FAULTS'
The lower four bits within the control word represent the device state machine's (DSM) control command.
⇨ Chapter "5.2.2.1 DSM state transitions caused by the control word", page 45

Bit 8: Enable pressure controller
This bit activates the pressure controller in the p/Q control mode (<ControlMode> (0#40) is set to 5).

0: Disables the pressure controller
1: Enables the pressure controller

⇨ Chapter "7.1 Control modes", page 118

Bit 9: Axis install mode positive direction
If the servo valve is in <DeviceMode> (0#39) = 3 (Install mode) or <DeviceMode> (0#39) = 4 (reference mode), then the hydraulic axis can be moved with bit 9 of the <ControlWord> (0#37) in positive direction. The velocity of the movement is defined in the parameter <ManualMaxSpeed> (69#2). If an incremental encoder is used to measure the axis position, then <DeviceMode> (0#39) = 4 (reference mode) is used to run the homing position to calibrate the incremental encoder.
⇨ Chapter "6.6.4 Incremental Encoder", page 89
⇨ Chapter "7.7 Axis position setpoint conditioning / demand value generator", page 181

Bit 10: Axis install mode negative direction
If the servo valve is in <DeviceMode> (0#39) = 3 (Install mode) or <DeviceMode> (0#39) = 4 (reference mode), then the hydraulic axis can be moved with bit 10 of the <ControlWord> (0#37) in negative direction. The velocity of the movement is defined in the parameter <ManualMaxSpeed> (69#2). If an incremental encoder is used to measure the axis position then <DeviceMode> (0#39) = 4 (reference mode) is used to run the homing position to calibrate the incremental encoder.
⇨ Chapter "6.6.4 Incremental Encoder", page 89
⇨ Chapter "7.7 Axis position setpoint conditioning / demand value generator", page 181

Bit 15: Ramp stop
If this bit is set, the spool position ramp and the pressure ramp output are frozen.

Spool position demand value generator: ⇨ Chapter "7.2.5 Ramp", page 131
Pressure demand value generator: ⇨ Chapter "7.4.5 Ramp", page 152

5.1.3 Object 0#206: Local control word

Parameter description: ⇨ Chapter "5.1.2 Object 0#37: Control word", page 40

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#206	LocalControlWord	0x4040#0	UINT16	rw	N	UINT16	<LocalControl-WordDefault> (0#205)

Value description

Same values as <ControlWord> (0#37)
⇨ Table 12, page 40

5.1.4 Object 0#205: Local control word default

The <LocalControlWordDefault> (0#205) defines the control word after power up of the digital servo valve. During startup of the servo valve, the parameter <LocalControlWordDefault> (0#205) is copied to the parameter <LocalControlWord> (0#206).

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#205	LocalControlWordDefault	0x403F#0	UINT16	rw	Y	UINT16	0x0107

5.2 Device state machine (DSM)

The device state machine (DSM) describes the states of the servo valve and the transitions between them. Any state represents a certain internal and external behavior. State changes result from DSM input and other events (for example switching on the supply voltage or on the appearance of a device fault). The current device state can be read by means of the <StatusWord> (0#38) (bits 0…3 of the status word indicate the device condition).

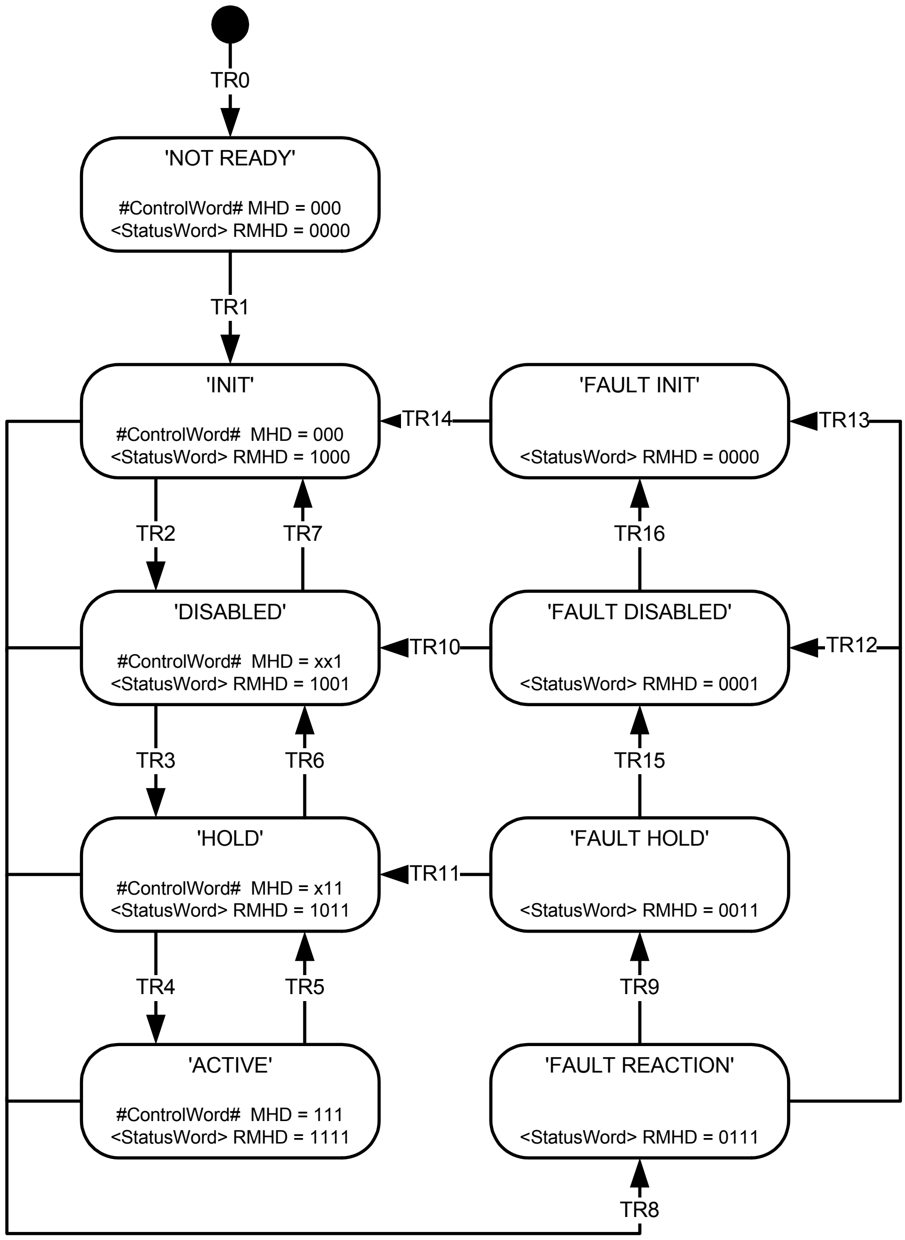


Figure 15: Device state machine

<StatusWord> (0#38)	**#ControlWord#**
(Bit 3) R: Ready	(Bit 3) R: Reset fault
(Bit 2) M: Active	(Bit 2) M: Active
(Bit 1) H: Hold	(Bit 1) H: Hold
(Bit 0) D: Disabled	(Bit 0) D: Disabled

5.2.1 DSM states

The <StatusWord> (0#38) indicates the DSM state. The following DSM states are possible:

'NOT_READY':

- The electronics circuit has power.
- Device initialization running (e.g. communication interface, hardware, software).
- Device function disabled.
- Hydraulic failsafe spool position (spool position depends on ordered servo valve model).

'INIT':

- Device parameters can be set.
- Device function disabled.
- Communication enabled.
- Hydraulic failsafe spool position (spool position depends on ordered servo valve model).

'DISABLED':

- Device parameters can be set.
- Device function disabled.
- Actual values are available.
- Hydraulic failsafe spool position (spool position depends on ordered servo valve model).

'HOLD':

- Device parameters can be set.
- Actual values are available.
- The selected <ControlMode> (0#40) is active.
  ⇨ Chapter "7.1 Control modes", page 118
- The setpoint values from the bus or from the analog input according to the chosen <DeviceMode> (0#39) are not effective.
  ⇨ Chapter "6.2.1 Object 0#39: Device mode", page 53
- The control loop depending on the selected <ControlMode> (0#40) is active and the corresponding pre-defined hold setpoint is used, for example, the <SplHoldSetPoint> (21#30…32) for the spool control or the <PrsHoldSetPoint> (22#30…32) for the pressure control.
  ⇨ Chapter "6.2.3.3 Object 21#30…32: Spl hold setpoint", page 56
  ⇨ Chapter "6.2.4.3 Object 22#30…32: Prs hold setpoint", page 58
  ⇨ Chapter "6.2.5.3 Object 12#33: Hold setpoint", page 59
  ⇨ Chapter "6.2.6.3 Object 13#23: Hold setpoint", page 63
  ⇨ Chapter "6.2.7.3 Object 70#6: Hold setpoint", page 66

'ACTIVE':

- Device parameters can be set.
- Actual values are available.
- The setpoint values from the bus or from the analog input according to the chosen <DeviceMode> (0#39) are active.

'FAULT_INIT':

- Device parameters can be set.
- Device function disabled.
- Communication enabled.
- Hydraulic failsafe spool position (spool position depends on ordered servo valve model).

'FAULT_DISABLED':

- Device parameters can be set.
- Actual values are available.
- A fault reaction has occurred.
- Hydraulic failsafe spool position (spool position depends on ordered servo valve model).

'FAULT_HOLD':

- Device parameters can be set.
- Actual values are available.
- A fault reaction has occurred.
- The selected <ControlMode> (0#40) is active.
  ⇨ Chapter "7.1 Control modes", page 118
- The setpoint values from the bus or from the analog input according to the chosen <DeviceMode> (0#39) are not effective.
  ⇨ Chapter "6.2.1 Object 0#39: Device mode", page 53
- The control loop depending on the selected <ControlMode> (0#40) is active and the corresponding predefined hold setpoint is used, for example, the <SplHoldSetPoint> (21#30…32) for the spool control or the <PrsHoldSetPoint> (22#30…32) for the pressure control.
  ⇨ Chapter "6.2.3.3 Object 21#30…32: Spl hold setpoint", page 56
  ⇨ Chapter "6.2.4.3 Object 22#30…32: Prs hold setpoint", page 58
  ⇨ Chapter "6.2.5.3 Object 12#33: Hold setpoint", page 59
  ⇨ Chapter "6.2.6.3 Object 13#23: Hold setpoint", page 63
  ⇨ Chapter "6.2.7.3 Object 70#6: Hold setpoint", page 66

'FAULT_REACTION':

- This state is assumed when the device detects an error.
- A fault dependent vendor specific action is executed.
  ⇨ Chapter "8.1.3 Fault reaction type", page 261
- The resulting fault state depends on the vendor specific <FaultReactionType> (72#102…219).
- Important condition for transitions 9, 12, 13:
  The RMHD bits of the #ControlWord# do not increase the state of the DSM.

Coming from	Meaning
'INIT'	'FAULT_INIT'
'DISABLED'	'FAULT_INIT', 'FAULT_DISABLED'
'HOLD'	'FAULT_INIT', 'FAULT_DISABLED', 'FAULT_HOLD'
'ACTIVE'	'FAULT_INIT', 'FAULT_DISABLED', 'FAULT_HOLD'
'FAULT_HOLD'	'FAULT_INIT', 'FAULT_DISABLED', 'FAULT_HOLD'
'FAULT_DISABLED'	'FAULT_INIT', 'FAULT_DISABLED'

The default state after power on if <LocalMode> (0#41) is set to 1 (Control Word Local) and enable signal on is defined by the parameter <LocalControlModeDefault> (0#205).
⇨ Chapter "5.1.4 Object 0#205: Local control word default", page 41

WARNING

Moving machine parts!

The word "failsafe" means not a personnel safety. Parts of the machine can move if the servo valve has a fault.

- If a personnel safety is needed, some additional electrical and hydraulic parts are necessary!

5.2.2 State transitions

State transitions are caused by

- The control word #ControlWord#
- Enable signal (digital input 0)
- Internal events

5.2.2.1 DSM state transitions caused by the control word

The following table lists the transitions depending on the #ControlWord#.
⇨ Chapter "5.2 Device state machine (DSM)", page 42

The device control commands, which cause a state transition, are formed by the four low-order bits of the #ControlWord#.
⇨ Chapter "5.1 Local mode", page 39

Every transition between the actual state and the requested state will be processed.

Transition (TR)	Control Word	Control word bit								Comments/Conditions
		7	6	5	4	3	2	1	0	
						R	M	H	D	
TR2	Activate 'DISABLED'	x	x	x	x	x	x	x	1	
TR3	Activate 'HOLD'	x	x	x	x	x	x	1	1	Depending on enable signal ⇨ Chapter "5.2.2.4 Enable behavior", page 46
TR4	Activate 'ACTIVE'	x	x	x	x	x	1	1	1	Depending on enable signal ⇨ Chapter "5.2.2.4 Enable behavior", page 46
TR5	Deactivate 'ACTIVE	x	x	x	x	x	0	x	X	
TR6	Deactivate 'HOLD'	x	x	x	x	x	0	0	X	
TR7	Deactivate 'DISABLED'	x	x	x	x	x	0	0	0	
TR10	Reset 'FAULT_DISABLED'	x	x	x	x	0	0	0	1	This transition is executed if the reset bit changes from 0 to 1 (rising edge) or the enable signal toggles from 0 to 1. ⇨ Chapter "5.2.2.4 Enable behavior", page 46 Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 48
		change to								
		x	x	x	x	1	0	0	1	
TR11	Reset 'FAULT_HOLD'	x	x	x	x	0	0	1	1	This transition is executed if the reset bit changes from 0 to 1 (rising edge) or the enable signal toggles from 0 to 1. ⇨ Chapter "5.2.2.4 Enable behavior", page 46 Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 48
		change to								
		x	x	x	x	1	0	1	1	
TR14	Reset 'FAULT_INIT'	x	x	x	x	0	0	0	0	This transition is executed if the reset bit changes from 0 to 1 (rising edge) or the enable signal toggles from 0 to 1. ⇨ Chapter "5.2.2.4 Enable behavior", page 46 Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 48
		change to								
		x	x	x	x	1	0	0	0	
TR15	'FAULT_HOLD' to 'FAULT_DISABLED'	x	x	x	x	x	0	0	1	
TR16	'FAULT_DISABLED' to 'FAULT_INIT'	x	x	x	x	x	0	0	0	

5.2.2.2 DSM state transitions caused by the enable signal

The following events will lead automatically to state changes.

Transition (TR)	Description	Comments/Conditions
TR2	Activate 'DISABLED'	Depending on ControlWord RMHD ≥ x001
TR3	Activate 'HOLD'	Depending on ControlWord RMHD ≥ x011
TR4	Activate 'ACTIVE'	Depending on ControlWord RMHD ≥ x111
TR5	Deactivate 'ACTIVE'	
TR6	Deactivate 'HOLD'	
TR7	Deactivate 'DISABLED'	
TR9	Transition from 'FAULT_REACTION' to 'FAULT_HOLD'	Depending on the enable behavior
TR10	Reset 'FAULT_DISABLED'	Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 48
TR11	Reset 'FAULT_HOLD'	Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 48
TR14	Reset 'FAULT_INIT'	Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 48
TR15	'FAULT_HOLD' to 'FAULT_DISABLED'	Depending on the enable behavior

The enable behavior is depending on the servo valve hardware configuration.

5.2.2.3 DSM state transitions caused by internal events

The following table shows the internal events which automatically lead to a state change.

Transition (TR)	Description	Comments/Conditions
TR0	Power up	
TR1	Device init successful.	Initialization of device parameters with stored values. Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 48
TR8	A fault was detected. On entering 'FAULT_REACTION' state an emergency message is sent out.	If state is 'DISABLED' or 'FAULT_DISABLED', state transitions to 'FAULT_HOLD' will be redirected to 'FAULT DISABLED'. Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 48
TR9	Transition from 'FAULT_REACTION' to 'FAULT_HOLD' (fault reaction successful).	Depending on enable signal and the enable behavior. ⇨ Chapter "5.2.2.4 Enable behavior", page 46
TR12	Transition from 'FAULT_REACTION' to 'FAULT_DISABLED' (fault reaction successful).	Depending on enable signal and the enable behavior. ⇨ Chapter "5.2.2.4 Enable behavior", page 46
TR13	Transition from 'FAULT_REACTION' to 'FAULT_INIT' (fault reaction successful).	Depending on enable signal and the enable behavior. ⇨ Chapter "5.2.2.4 Enable behavior", page 46

5.2.2.4 Enable behavior

The enable signal comes from the connector X1. The enable signal influences the device state machine and can be used to acknowledge faults.

⇨ Chapter "6.7.1 Digital input 0 (enable signal)", page 92

5.2.2.4.1 DSM state transitions depending on the enable signal

The enable signal can cause different valve responses ('HOLD' or 'DISABLE'). The response to the enable signal is specified by the servo valve type designation number 13 within the order code. The configuration is set by Moog during the production and cannot be changed by software. These hardware dependent options are shown in the following tables.

If the enable signal changes from 1 to 0, a state machine transition is triggered to either state 'HOLD' (closed loop) or state 'DISABLED' (open loop).

State transitions do not necessarily refer to a specific spool position or change of spool position. The specific spool position also depends on the chosen failsafe function on type designator position 6, the pilot connection type designator position 7 and the availability of pilot pressure.

⚠ WARNING

Moving machine parts!

The word "failsafe" means not a personnel safety. Parts of the machine can move if the servo valve has a fault.

- If a personnel safety is needed, some additional electrical and hydraulic parts are necessary!

Servo valves with the type designator 13 in the order code of O:
There is no reaction to the enable signal at all. With power ON, the valve always will switch to state 'ACTIVE'.

Servo valves with the type designator 13 in the order code of B, D, F, H, L and R:
If the enable signal is switched off (0 V), the state machine will switch to state 'DISABLED'.

Enable signal	Transition (TR)	Old DSM state	New DSM state	Comments/Conditions
1 -> 0	TR5, TR6	'HOLD', 'ACTIVE'	'DISABLED'	RMHD ≤ 1001
	TR15	'FAULT_HOLD'	'FAULT_DISABLED'	

Servo valves with the type designation number 13 in the order code A, C, E, G, J, K, M, S and T:
If the enable signal is switched off (0 V), the state machine will switch to state 'HOLD'. Depending on the selected control mode, a specific hold position is controlled.

➪ Chapter "5.2.1 DSM states", page 43
➪ Chapter "6.2.3 Spool position setpoint value path", page 54
➪ Chapter "6.2.4 Pressure setpoint value path", page 56

Enable signal	Transition (TR)	Old DSM state	New DSM state	Comments/Conditions
1 -> 0	TR5	'ACTIVE'	'HOLD'	RMHD ≤ 1001
	-	'FAULT_HOLD'	No change	

5.2.2.4.2 Fault confirmation with the enable signal

Toggling the enable signal from low to high causes the device state machine to erase all errors. If no error is pending, the state machine will exit the fault state.

Enable signal	Transition (TR)	Old DSM state	New DSM state
0 -> 1	TR10	'FAULT_DISABLED'	Depending on the #ControlWord#
	TR11	'FAULT_HOLD'	Depending on the #ControlWord#

5.2.2.5 Error output pin

The error output (digital output 1) is used to indicate fault states (negative logic) according to the Device Profile Fluid Power.

- Digital output 1 is set to 1 on power on (TR1) of the servo valve.
- When a fault is detected (TR8) the digital output 1 is set to 0 to indicate a fault (negative logic).
- When a fault state is left (TR10, TR11) the digital output 1 is set to 1.

To enable this behavior on the digital output 1, the parameter <DigitalOutputType1> (0#221) must be set to 2.
⇨ Chapter "6.8.2 Object 0#220…221: Digital output configuration", page 94

5.2.3 Object 0#38: Status word

The bit-coded <StatusWord> (0#38) indicates the current device status.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#38	StatusWord	0x6041#0	UINT16	ro	-	UINT16	None

Value description

<StatusWord>				
Bit	**<StatusWord> bit**	**<Control Mode> (0#40) is set to 1…4, 7…14**	**<Control Mode> (0#40) is set to 5**	**Specification**
0	Bit Disabled (D)			CiA 408
1	Bit Hold (H)			
2	Bit Active (M)			
3	Bit Ready (R)			
4	Indicates that bit <Local> (0#41) is set			CiA 408
5…7	Reserved			
8	<ControlMode> (0#40) specific	Reserved	Pressure controller effective	CiA 408
9	Ramp running			CiA 408
10	Limit touched (c)			CiA 408
11	Control deviation			CiA 408
12…14	Reserved			
15	Ramp frozen			Moog DCV

Table 13: Possible values of parameter <StatusWord> (0#38)

Bits 0, 1, 2, 3: 'DISABLED', 'HOLD', 'ACTIVE', 'READY'
These bits indicate the state of the device state machine (DSM).
⇨ Chapter "5.2 Device state machine (DSM)", page 42

Bit 4: Indicates that bit <Local> (0#41) is set
The <LocalControlWord> (0#206) is the active control word.
⇨ Chapter "5.1 Local mode", page 39

Bit 8: Enable pressure controller
This bit indicates whether the pressure controller is effective or not. In this case the pressure controller limits the spool position (flow).

0: Output of spool position controller limits the flow.
1: Output of pressure controller limits the flow.

⇨ Chapter "7.1 Control modes", page 118

This bit is only active if the <ControlMode> (0#40) is set to 5 (p/Q control).

Bit 9: Ramp running
This bit is set if the following conditions are true:

- Spool position and/or pressure ramp function is active and
- Spool position and/or pressure ramp function is running and
- #ControlWord# bit 15 is set to false.

Spool position demand value generator: ⇨ Chapter "7.2.5 Ramp", page 131
Pressure demand value generator: ⇨ Chapter "7.4.5 Ramp", page 152

Bit 10: Limit value reached
This bit indicates that one of the setpoint values is limited by the corresponding limit function set with the demand value generator functions.

Spool position demand value generator: ⇨ Chapter "7.2.3 Limit function", page 128
Pressure demand value generator: ⇨ Chapter "7.4.3 Limit function", page 150

Bit 11: Control deviation
This bit indicates a control deviation, detected by one of the control monitoring functions, e.g. the control deviation has been outside the tolerance band for longer than the specified delay time.
⇨ Chapter "7.15 Monitoring", page 222

Bit 15: Ramp frozen
This bit is set if the following conditions are true:

- Spool position and/or pressure ramp function is active and
- #ControlWord# bit 15 is set to true.

Spool position demand value generator: ⇨ Chapter "7.2.5 Ramp", page 131
Pressure demand value generator: ⇨ Chapter "7.4.5 Ramp", page 152

5.2.4 Object 64#52: Manufacturer Status Register

The <ManufacturerStatusRegister> (64#52) indicates the current status of the digital input 0 (enable signal).

⇨ Chapter "6.7.1 Digital input 0 (enable signal)", page 92

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#52	ManufacturerStatusRegister	0x1002#0	UINT32	ro	-	UINT32	None

5.3 Bootup of the device

The bootup sequence needs about 500 ms. It can be delayed with the <PowerOnDelay> (0#202) parameter.

5.3.1 Object 0#202: Power On Delay

The parameter <PowerOnDelay> (0#202) allows delaying the bootup procedure before establishing the communication and servo valve functions. The <PowerOnDelay> (0#202) time is provided in seconds.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#202	PowerOnDelay	0#202#0	UINT8	rw	Y	0…10	0

5.4 Status display LEDs

The network and the servo valve's states are indicated by multicolor light emitting diodes (status display LEDs) on the electronics housing.

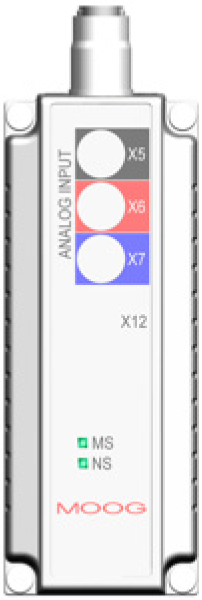


Figure 16: Status display LEDs

5.4.1 Module status LED «MS»

The module status LED displays the state of the device state machine (DSM).

⇨ Chapter "5.2 Device state machine (DSM)", page 42

Module status LED «MS»	**Valve State Machine (status word) (according to Device Profile Fluid Power)**	**Description**
Off		No supply power.
Green blinking	'INIT' or 'DISABLED'	Servo valve standby mode.
Green	'HOLD' or 'ACTIVE'	Normal operation.
Red blinking	'FAULT_DISABLED' or 'FAULT_HOLD'	Recoverable error. This state can be caused by fault reactions 'FAULT_DISABLED' or 'FAULT_HOLD'. ⇨ Chapter "8.1.3 Fault reaction type", page 261
Red	'NOT_READY'	Unrecoverable error. This state can be caused by fault reactions 'FAULT_INIT' or 'FAULT_STOP'. ⇨ Chapter "8.1.3 Fault reaction type", page 261

5.4.2 Network status LED «NS»

The network status LED displays the state of the slave network state machine.

⇨ Chapter "2.7 Slave Network state machine (DP-V0)", page 17

Network status LED «NS»	Slave network state machine	Description
Off	Stopped	No power supply or not connected.
Green	'DATA_EXCH'	Valve is in data exchange state.
Green blinking	'WAIT_CFG'	Valve is waiting for configuration telegram.
Orange	'Baud_Control' or 'WAIT_PRM'	Valve has detected baud rate. Valve is waiting for parameter telegram.
Orange blinking	'Baud_Search'	Valve is searching for correct baud rate.

6 Signal routing and scaling

6.1 Signal routing structure

The following picture shows the structure of the signal routing for the setpoint values and the physical actual values of the servo valve depending on the <ControlMode> (0#40) used. The blocks with gray backgrounds are described in detail in this chapter.

➪ Chapter "7.1 Control modes", page 118

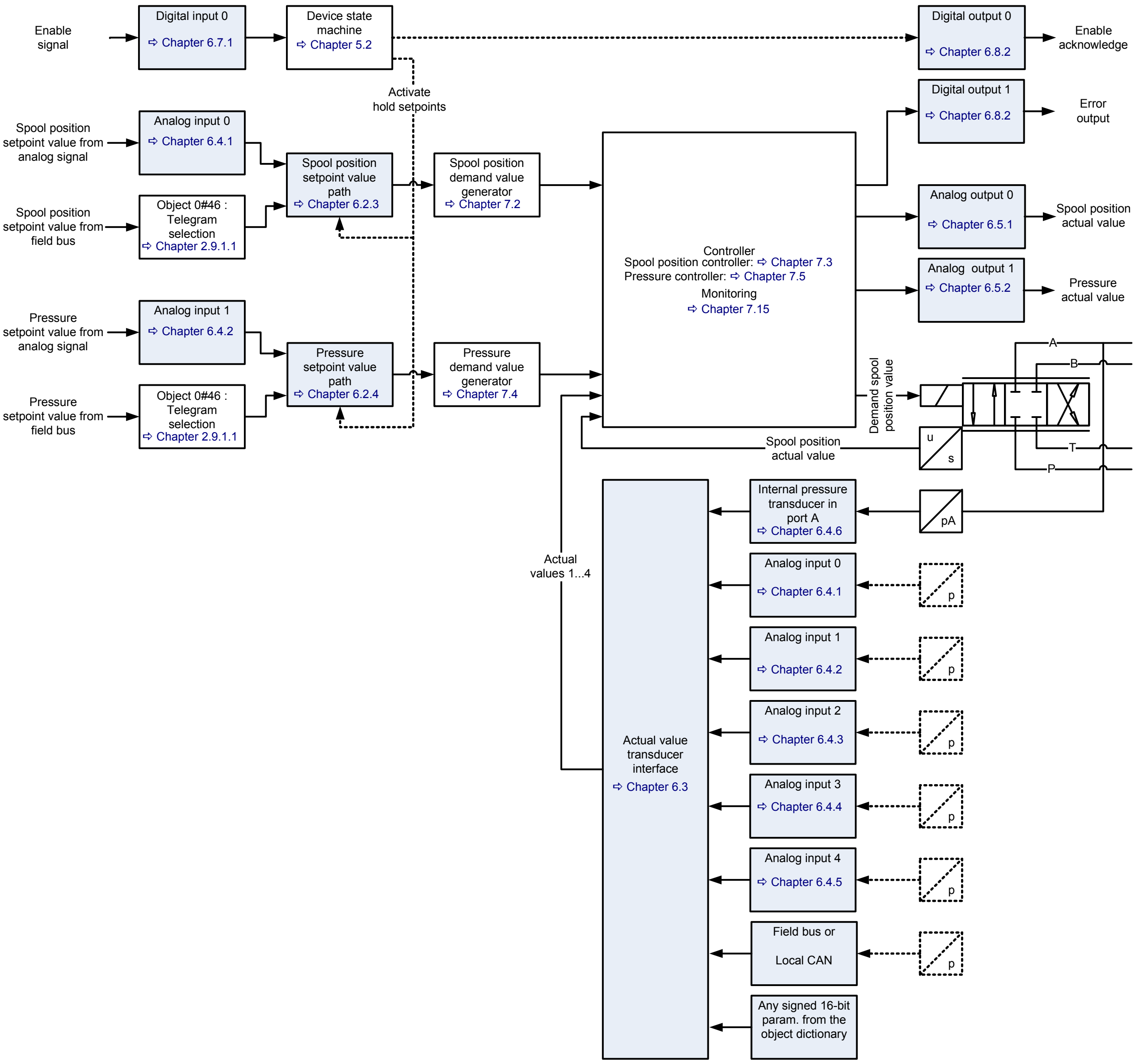


Figure 17: Signal routing

6.2 Setpoint value path

The setpoint values for pressure, spool position, axis position, axis velocity and flow can be received via the field bus or via the analog/encoder inputs. The axis setpoints (position, velocity and flow) are only available in axis control valves.

- The spool position setpoint value received via bus is provided by the parameter <SplSetpoint> (21#21…23).
- The pressure setpoint value received via bus is provided by the parameter <PrsSetpoint> (22#21…23).
- The axis position setpoint value received via bus is provided by the parameter <PosSetpoint> (12#21).
- The axis velocity setpoint value received via bus is provided by the parameter <VelSetpoint> (13#21).
- The flow setpoint value received via bus is provided by the parameter <FlwSetpoint> (70#2).
- The spool position setpoint value from local source is coming from the analog input 0 <AnaInpActualValue0> (74#1).
- The pressure setpoint value from local source is coming from the analog input 1 <AnaInpActualValue1> (74#3).
- The axis position setpoint, axis velocity setpoint and the flow setpoint values from local source are not referenced to a special hardware input.

Which setpoint is in effect depends on the parameter <DeviceMode> (0#39) and the <StatusWord> (0#38):

- <DeviceMode> (0#39) is set to 1 (setpoint input via bus), <StatusWord> (0#38) is 0111_b ('ACTIVE'): The setpoint value received via field bus is forwarded to the demand value generator.
- <DeviceMode> (0#39) is set to 2 (setpoint input locally), <StatusWord> (0#38) is 0111_b ('ACTIVE'): The setpoint value coming from the analog input is forwarded to the demand value generator.
- <StatusWord> (0#38) is 0011_b ('HOLD'):
  The hold setpoint values are taken as setpoint values. This hold setpoint is in effect regardless if the device mode <DeviceMode> (0#39) is set to 1 (setpoint input via bus) or <DeviceMode> (0#39) is set to 2 (setpoint input locally).
  The spool position hold setpoint value is stored in the parameter <SplHoldSetpoint> (21#30…32). The pressure hold setpoint value is stored in the parameter <PrsHoldSetpoint> (22#30…32). In the same manner the setpoints for the axis position, velocity and flow are stored in the parameter <PosHoldSetpoint> (12#33), <VelHoldSetpoint> (13#23) and <FlwHoldSetpoint> (70#6…8).

6.2.1 Object 0#39: Device mode

The device mode is used to switch the setpoint value source from local input (e.g., an analog input) to setpoint value input via bus.

The <DeviceMode> (0#39) = 3 (install mode) and = 4 (reference mode) has no direct influence to the position setpoint path. These two device modes are enabling the trajectory generator for manual movement of the hydraulic axis. The axis can be moved in positive or negative direction with bit 9 and 10 of the <ControlWord> (0#37). The <DeviceMode> (0#39) = 4 (reference mode) is used to enable the referencing run for an incremental encoder.

⇨ Chapter "5.1.2 Object 0#37: Control word", page 40
⇨ Chapter "6.6.4 Incremental Encoder", page 89
⇨ Chapter "7.7 Axis position setpoint conditioning / demand value generator", page 181

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#39	DeviceMode	0x6042#0	INT8	rw	N	1…4	<DeviceModeDefault> (0#207)

Value description

<DeviceMode>	Type of analog input
0	Reserved
1	Setpoint input via the bus
2	Setpoint input locally
3	Install mode
4	Reference mode
All other	Reserved

Table 14: Possible values of parameter <DeviceMode> (0#39)

The effective <DeviceMode> (0#39) after power up is defined by the parameter <DeviceModeDefault> (0#207).
⇨ Chapter "6.2.2 Object 0#207: Device mode default", page 54

6.2.2 Object 0#207: Device mode default

The <DeviceModeDefault> (0#207) defines the active device mode after power up of the servo valve. This is achieved by automatically copying the parameter <DeviceModeDefault> (0#207) to the parameter <DeviceMode> (0#39) during the startup procedure of the servo valve

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#207	DeviceModeDefault	0x4042#0	INT8	rw	Y	1…2	1

6.2.3 Spool position setpoint value path

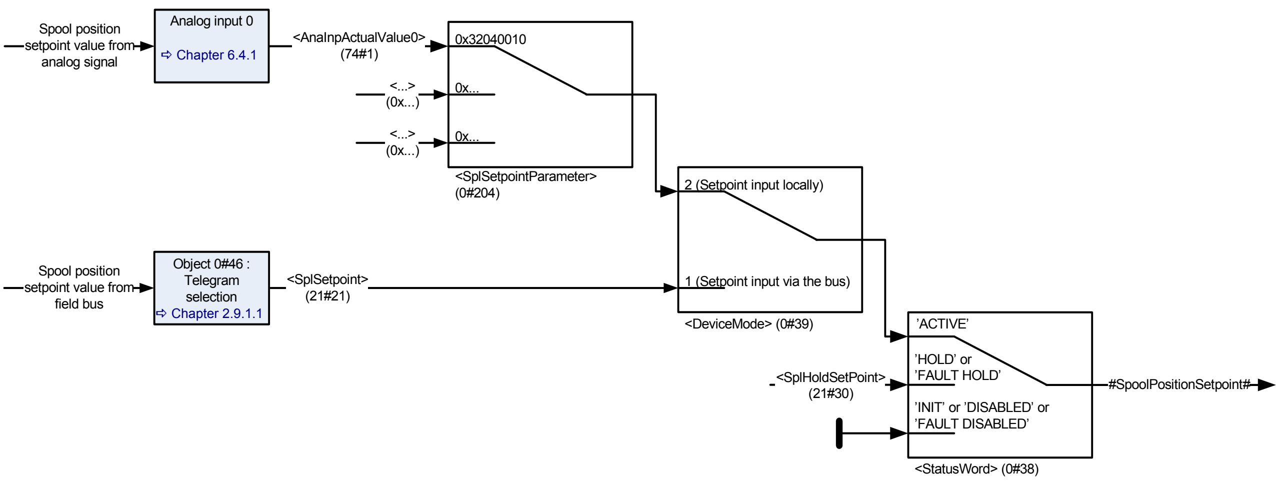


Figure 18: Spool position setpoint value path

6.2.3.1 Object 21#21…23: Setpoint

This parameter contains the spool position setpoint value which is received from the field bus. Depending on the <DeviceMode> (0#39), this parameter is in effect for the following three control modes stored in the parameter <ControlMode> (0#40):

1 Spool position control open loop
2 Spool position control closed loop
5 p/Q control
7 Axis velocity control
9 Axis position control
13 Flow control
14 p/flow control

The setpoint value <SplSetpoint> (21#21…23) takes only effect if the <StatusWord> (0#38) is 0111_b ('ACTIVE') and the <DeviceMode> (0#39) is set to 1 (setpoint input via bus).

➪ Chapter "5.2.3 Object 0#38: Status word", page 48
➪ Chapter "6.2.1 Object 0#39: Device mode", page 53

ValvePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#21	SplSetpoint	0x6300#1	INT16	rw	N	INT16	None
21#22	Unit	0x6300#2	UINT8	ro	-	UINT8	0
21#23	Prefix	0x6300#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

6.2.3.2 Object 0#204: Setpoint parameter

The spool position setpoint value parameter <SplSetpointParameter> (0#204) points to the input where the spool position setpoint value <SplSetpoint> (21#21…23) comes from.

The spool position setpoint value <SplSetpoint> (21#21…23) is only effective in case the <StatusWord> (0#38) is 0111_b ('ACTIVE') and the <DeviceMode> (0#39) is set to 1 (setpoint input via bus).

➪ Chapter "5.2.3 Object 0#38: Status word", page 48
➪ Chapter "6.2.1 Object 0#39: Device mode", page 53

ValvePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#204	SplSetpointParameter	0x3320#0	INT32	rw	-	INT32	0x63000110

Value description

<SetpointParameter>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length: 0x10
Default	0x63	0x00	0x01	0x10

Table 15: Possible values of parameter <SplSetpointParameter> (0#204)

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

The default value is 0x63000110, which refers to the <SplSetpoint> (21#21…23), with the CANopen index 0x6300 and the CANopen sub-index 0x01 with a length of 16 bit (16=0x10).

6.2.3.3 Object 21#30…32: Spl hold setpoint

This parameter defines the spool position hold setpoint value for the <ControlMode> (0#40):

1 Spool position control open loop
2 Spool position control closed loop
5 p/Q control
7 Axis velocity control
9 Axis position control
13 Flow control
14 p/flow control

⇨ Chapter "7.1.1 Object 0#40: Control mode", page 119

The <SplHoldSetpoint> (21#30…32) acts as setpoint value in case of <StatusWord> (0#38) is 1011_b ('HOLD') or 0011_b ('FAULT_HOLD').

⇨ Chapter "5.2.3 Object 0#38: Status word", page 48

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#30	SplHoldSetpoint	0x6314#1	INT16	rw	Y	INT16	0
21#31	Unit	0x6314#2	UINT8	ro	-	UINT8	0
21#32	Prefix	0x6314#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

6.2.4 Pressure setpoint value path

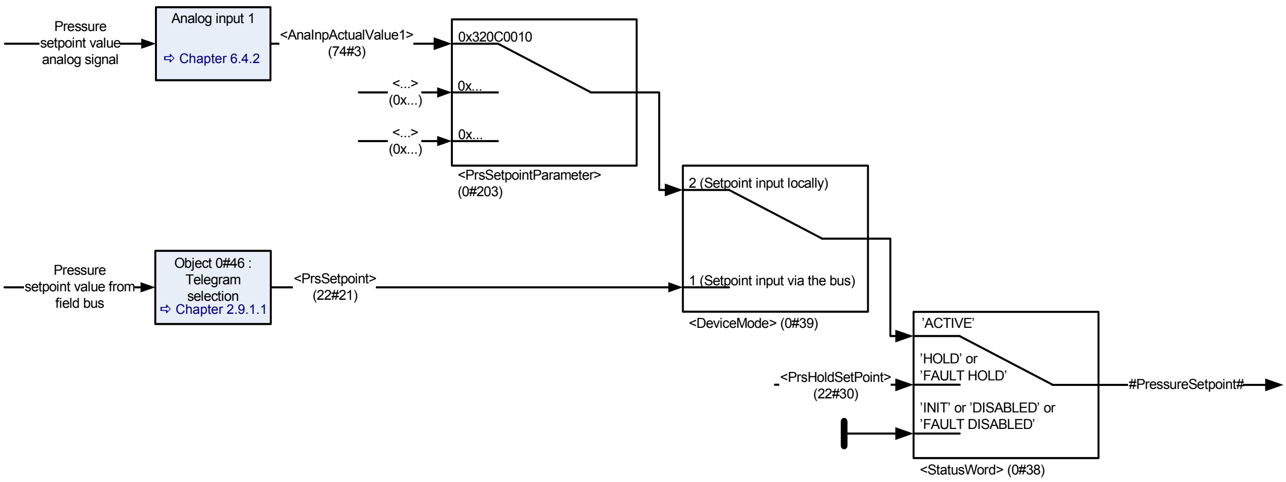


Figure 19: Pressure setpoint value path

6.2.4.1 Object 22#21…23: Setpoint

This parameter contains the pressure setpoint value which is received from the field bus. Depending on the <DeviceMode> (0#39), this parameter is in effect for the following three control modes stored in the parameter <ControlMode> (0#40):

3 Pressure control open loop
4 Pressure control closed loop
5 p/Q control
14 p/flow control

➪ Chapter "7.1.1 Object 0#40: Control mode", page 119

The setpoint value takes only effect in case the <StatusWord> (0#38) is 1111_b ('ACTIVE') and the <DeviceMode> (0#39) is set to 1 (setpoint input via bus).

➪ Chapter "5.2.3 Object 0#38: Status word", page 48
➪ Chapter "6.2.1 Object 0#39: Device mode", page 53

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
22#30	PrsSetpoint	0x6380#1	INT16	rw	N	INT16	None
22#31	Unit	0x6380#2	UINT8	ro	-	UINT8	0
22#32	Prefix	0x6380#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

6.2.4.2 Object 0#203: Setpoint parameter

The pressure setpoint value parameter <PrsSetpointParameter> (0#203) points to the input where the pressure setpoint value <PrsSetpoint> (22#21…23) comes from.

The pressure setpoint value <PrsSetpoint> (22#21…23) is only effective in case the <StatusWord> (0#38) is 1111_b ('ACTIVE') and the <DeviceMode> (0#39) is set to 1 (setpoint input via bus).

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#203	PrsSetpointParameter	0x3310#0	INT32	rw	Y	INT32	0x63000110

Value description

<SetpointParameter>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length: 0x10
Default	0x63	0x80	0x01	0x10

Table 16: Possible values of parameter <PrsSetpointParameter> (0#203)

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

The default value is 0x63800110, which refers to the <PrsSetpoint> (22#21…23), with the CANopen index 0x6300 and the CANopen sub-index 0x01 with a length of 16 bit (16=0x10).

6.2.4.3 Object 22#30…32: Prs hold setpoint

This parameter defines the pressure hold setpoint value. It is effective for the following <ControlMode> (0#40):

3 Pressure control open loop
4 Pressure control closed loop
5 p/Q control
8 Force control
14 p/flow control

➪ Chapter "7.1.1 Object 0#40: Control mode", page 119

The <PrsHoldSetpoint> (22#30…32) acts as setpoint value in case of <StatusWord> (0#38) equals 'HOLD' or 'FAULT_HOLD'.

➪ Chapter "5.2.3 Object 0#38: Status word", page 48

ValvePressureControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
22#30	PrsHoldSetpoint	0x6394#1	INT16	rw	Y	INT16	0
22#31	Unit	0x6394#2	UINT8	ro	-	UINT8	0
22#32	Prefix	0x6394#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

6.2.5 Axis position setpoint value path

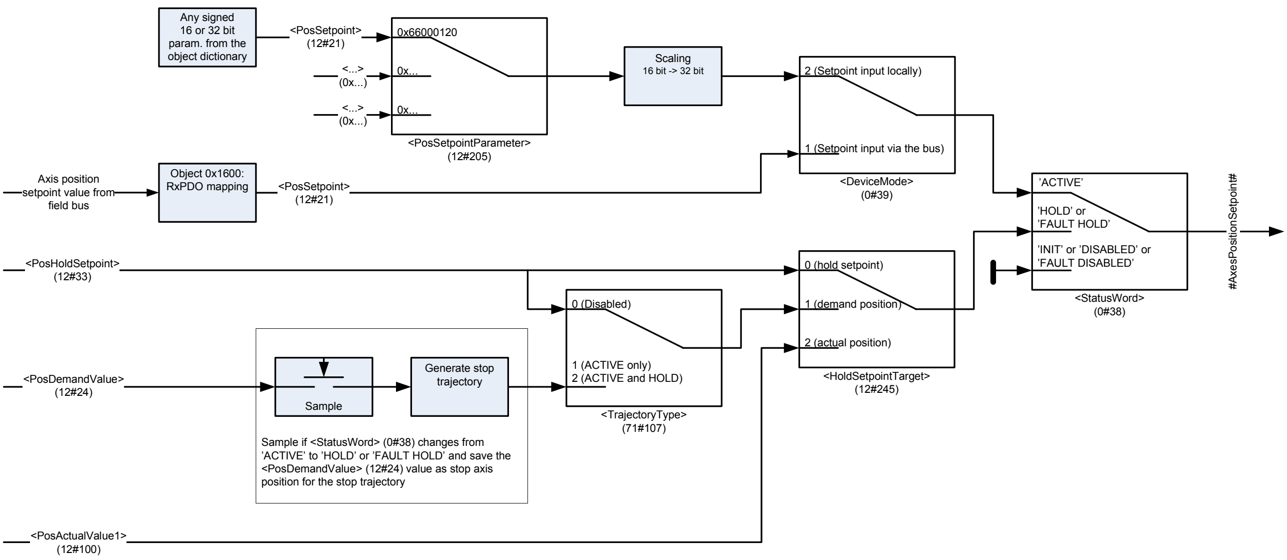


Figure 20: Axis position setpoint value path

➪ Chapter "7.7.1 Object 12#24: Position demand value", page 181

6.2.5.1 Object 12#21: Setpoint

This parameter contains the axis position setpoint value which is received from the field bus. Depending on the <DeviceMode> (0#39), this parameter is in effect only for the <ControlMode> (0#40) = 8 (axis position control).
➪ Chapter "7.1.1 Object 0#40: Control mode", page 119

The setpoint value takes only effect in case the <StatusWord> (0#38) is 1111_b ('ACTIVE') and the <DeviceMode> (0#39) is set to 1 (setpoint input via bus).

➪ Chapter "5.2.3 Object 0#38: Status word", page 48
➪ Chapter "6.2.1 Object 0#39: Device mode", page 53

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#21	PosSetpoint	0x6600#1	INT32	rw	N	INT32	None
12#22	Unit	0x6600#2	UINT8	ro	-	UINT8	0
12#23	Prefix	0x6600#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

6.2.5.2 Object 12#205: Setpoint parameter

The axis position setpoint value parameter <PosSetpointParameter> (12#205) points to the input where the axis position setpoint value <PosSetpoint> (12#21) comes from.

The axis position setpoint value <PosSetpoint> (12#21) is only effective in case the <StatusWord> (0#38) is 1111_b ('ACTIVE') and the <DeviceMode> (0#39) is set to 1 (setpoint input via bus).

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#205	PosSetpointParameter	0x3300	INT32	rw	Y	INT32	0x66000120

Value description

<PosSetpointParameter>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen index	Parameter bit length
Default	0x63	0x80	0x01	0x20

Table 17: Possible values of parameter <PosSetpointParameter> (12#205)

This pointer contains a combination of index, sub-index and length of the parameter to be used.

The default value is 0x66000120, which refers to the <PosSetpoint> (12#21), sub-index 0x01 with the length of 32 bit (32 = 0x20).

6.2.5.3 Object 12#33: Hold setpoint

This parameter defines the axis position hold setpoint value. This parameter is in effect only for the <ControlMode> (0#40) = 8 (axis position control).
➪ Chapter "7.1.1 Object 0#40: Control mode", page 119

The <PosHoldSetpoint> (12#33) acts as setpoint value in case of <StatusWord> (0#38) equals 'HOLD' or 'FAULT_HOLD'.
➪ Chapter "5.2.3 Object 0#38: Status word", page 48

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#33	PosHoldSetpoint	0x6614#1	INT32	rw	Y	INT32	0
12#34	Unit	0x6614#2	UINT8	ro	-	UINT8	0
12#35	Prefix	0x6614#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

6.2.5.4 16 to 32 bit scaling

Only if a 16 bit parameter with the <PosSetpointParameter> (12#205) is mapped, a scaling of 16 bit input value is available.

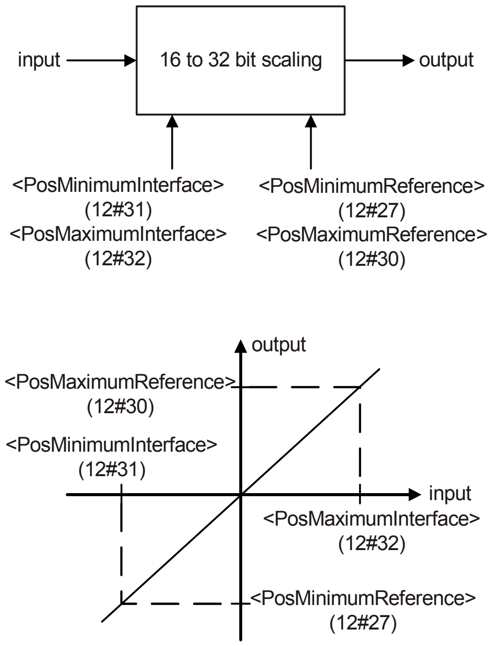


Figure 21: Axis position setpoint value path - 16 to 32 bit scaling

6.2.5.4.1 Object 12#27: Minimum reference

This parameter together with the parameter <PosMinimumInterface> (12#31) results in the first point of the linear scaling function.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#27	PosMinimumReference	0x4230	INT32	rw	Y	INT32	–16384

6.2.5.4.2 Object 12#30: Maximum reference

This parameter together with the parameter <PosMaximumInterface> (12#32) results in the second point of the linear scaling function.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#30	PosMaximumReference	0x4231	INT32	rw	Y	INT32	16384

6.2.5.4.3 Object 12#31: Minimum interface

This parameter together with the parameter <PosMinimumReference> (12#27) results in the first point of the linear scaling function.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#31	PosMinimumInterface	0x5509	INT32	rw	Y	INT32	–16384

6.2.5.4.4 Object 12#32: Maximum interface

This parameter together with the parameter <PosMaximumReference> (12#30) results in the second point of the linear scaling function.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#32	PosMaximumInterface	0x5510	INT32	rw	Y	INT32	16384

6.2.5.5 Object 71#107: Trajectory Type

With the type it is possible to switch the trajectory off or enable it in the state machine state 'ACTIVE' or 'HOLD'.

DrivePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#107	TrajectoryType	0x5700	INT8	rw	Y	0…2	0

Value description

<TrajectoryType>	**Behavior of preload output**
0 (no trajectory)	The trajectory signals for axis velocity and acceleration are calculated as derivation of the axis position signal. ➪ Chapter "7.7.4 Axis position trajectory generator", page 182
1 (only 'ACTIVE')	The trajectory signals for axis position, velocity and acceleration are calculated as realtime trajectory based on the axis position signal. The trajectory generator is effective, if the device state machine state <StatusWord> (0#38) is in the state 'ACTIVE'. ➪ Chapter "7.7.4 Axis position trajectory generator", page 182
2 ('ACTIVE' and 'HOLD')	The trajectory signals for axis position, velocity and acceleration are calculated as realtime trajectory based on the axis position signal. The trajectory generator is effective, if the device state machine state <StatusWord> (0#38) is in the state 'ACTIVE' or 'HOLD'. ➪ Chapter "7.7.4 Axis position trajectory generator", page 182

Table 18: Possible values of parameter <TrajectoryType> (71#107)

6.2.5.6 Object 12#245: Hold setpoint target

DrivePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#245	HoldSetpointTarget	0x585C	INT8	rw	Y	0…2	0

Value description

<HoldSetpointTarget>	Target position of the trajectory
0 (hold position)	The target position of the trajectory is jumping to the <PosHoldSetpoint> (12#33) while the <StatusWord> (0#38) is changing from 'ACTIVE' to 'HOLD'.
1 (demand position)	The target position of the trajectory is jumping to the <DemandValue> (12#24) while the <StatusWord> (0#38) is changing from 'ACTIVE' to 'HOLD'. The axis will be stopped with maximal acceleration.
2 (actual position)	The target position of the trajectory is jumping to the <PosActualValue1> (12#100) while the <StatusWord> (0#38) is changing from 'ACTIVE' to 'HOLD'.

Table 19: Possible values of parameter <HoldSetpointTarget> (12#245)

6.2.6 Axis velocity setpoint value path

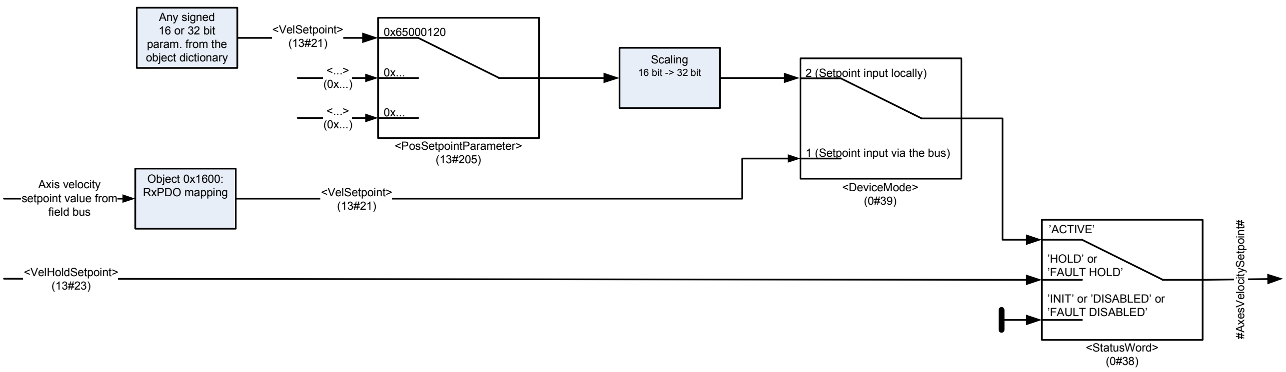


Figure 22: Axis velocity setpoint value path

6.2.6.1 Object 13#21: Setpoint

This parameter contains the axis velocity setpoint value which is received from the field bus. Depending on the <DeviceMode> (0#39), this parameter is in effect only for the <ControlMode> (0#40) = 6 (axis velocity control).

➪ Chapter "7.1.1 Object 0#40: Control mode", page 119

The setpoint value takes only effect in case the <StatusWord> (0#38) is 1111_b ('ACTIVE') and the <DeviceMode> (0#39) is set to 1 (setpoint input via bus).

➪ Chapter "5.2.3 Object 0#38: Status word", page 48
➪ Chapter "6.2.1 Object 0#39: Device mode", page 53

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#21	VelSetpoint	0x6500#1	INT32	rw	N	INT32	None
13#22	Unit	0x6500#2	UINT8	ro	-	UINT8	0
13#23	Prefix	0x6500#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

6.2.6.2 Object 13#205: Setpoint parameter

The axis velocity setpoint value parameter <VelSetpointParameter> (13#205) points to the input where the axis velocity setpoint value <VelSetpoint> (13#21) comes from.

The axis velocity setpoint value <VelSetpoint> (13#21) is only effective in case the <StatusWord> (0#38) is 1111_b ('ACTIVE') and the <DeviceMode> (0#39) is set to 1 (setpoint input via bus).

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#205	VelSetpointParameter	0x5520	INT32	rw	Y	INT32	0x65000120

Value description

<VelSetpointParameter>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen index	Parameter bit length
Default	0x65	0x00	0x01	0x20

Table 20: Possible values of parameter <VelSetpointParameter> (13#205)

This pointer contains a combination of index, sub-index and length of the parameter to be used.

The default value is 0x65000120, which refers to the <VelSetpoint> (13#21), sub-index 0x01 with the length of 32 bit (32 = 0x20).

6.2.6.3 Object 13#23: Hold setpoint

This parameter defines the axis velocity hold setpoint value. This parameter is in effect only for the <ControlMode> (0#40) = 6 (axis velocity control).
⇨ Chapter "7.1.1 Object 0#40: Control mode", page 119

The <VelHoldSetpoint> (13#23) acts as setpoint value in case of <StatusWord> (0#38) equals 'HOLD' or 'FAULT_HOLD'.
⇨ Chapter "5.2.3 Object 0#38: Status word", page 48

DriveVelocityControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#23	VelHoldSetpoint	0x5527#1	INT32	rw	N	INT32	0
13#24	Unit	0x5527#2	UINT8	ro	-	UINT8	0
13#25	Prefix	0x5527#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

6.2.6.4 16 to 32 bit scaling

Only if a 16 bit parameter with the <VelSetpointParameter> (13#205) is mapped, a scaling of 16 bit input value is available.

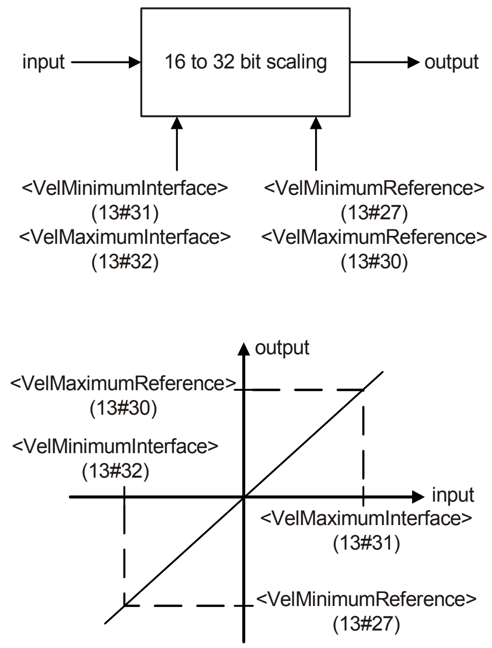


Figure 23: Axis velocity setpoint value path - 16 to 32 bit scaling

6.2.6.4.1 Object 13#27: Minimum reference

This parameter together with the parameter <VelMinimumInterface> (13#30) results in the first point of the linear scaling function.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#27	VelMinimumReference	0x5523	INT32	rw	Y	INT32	–16384

6.2.6.4.2 Object 13#30: Maximum reference

This parameter together with the parameter <VelMaximumInterface> (13#32) results in the second point of the linear scaling function.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#30	VelMaximumReference	0x5524	INT32	rw	Y	INT32	16384

6.2.6.4.3 Object 13#31: Minimum interface

This parameter together with the parameter <VelMinimumReference> (13#27) results in the first point of the linear scaling.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#31	VelMinimumInterface	0x5525	INT32	rw	Y	INT32	–16384

6.2.6.4.4 Object 13#32: Maximum interface

This parameter together with the parameter <VelMaximumReference> (13#30) results in the second point of the linear scaling function.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#32	VelMaximumInterface	0x5526	INT32	rw	Y	INT32	16384

6.2.7 Flow setpoint value path

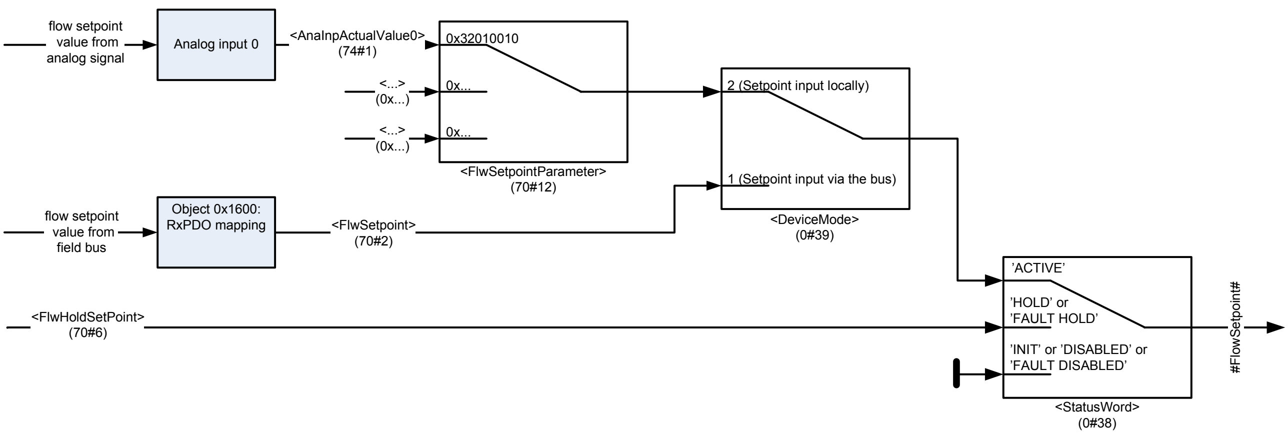


Figure 24: Flow setpoint value path

6.2.7.1 Object 70#2: Setpoint

This parameter contains the flow setpoint value which is received from the field bus. Depending on the <DeviceMode> (0#39), this parameter is in effect for the following control modes stored in the parameter <ControlMode> (0#40):

13 flow control
14 p/flow control

⇨ Chapter "7.1.1 Object 0#40: Control mode", page 119

The setpoint value takes only effect in case the <StatusWord> (0#38) is 1111_b ('ACTIVE') and the <DeviceMode> (0#39) is set to 1 (setpoint input via bus).
⇨ Chapter "5.2.3 Object 0#38: Status word", page 48
⇨ Chapter "6.2.1 Object 0#39: Device mode", page 53

DriveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
70#2	FlwSetpoint	0x5300#1	INT16	rw	N	INT16	None
70#3	Unit	0x5300#2	UINT8	ro	-	UINT8	0
70#4	Prefix	0x5300#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

6.2.7.2 Object 70#12: Setpoint parameter

The flow setpoint value parameter <FlwSetpointParameter> (70#12) points to the input where the flow setpoint value <FlwSetpoint> (70#2) comes from.

The flow setpoint value <FlwSetpoint> (70#2) is only effective in case the <StatusWord> (0#38) is 1111_b ('ACTIVE') and the <DeviceMode> (0#39) is set to 1 (setpoint input via bus).

DriveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
70#12	FlwSetpointParameter	0x5202#1	INT32	rw	N	INT32	0x53000110

Value description

<FlwSetpointParameter>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen index	Parameter bit length
Default	0x53	0x00	0x01	0x10

Table 21: Possible values of parameter <FlwSetpointParameter> (70#12)

This pointer contains a combination of index, sub-index and length of the parameter to be used.

The default value is 0x65000110, which refers to the <FlwSetpoint> (70#2), sub-index 0x01 with the length of 16 bit (16 = 0x10).

6.2.7.3 Object 70#6: Hold setpoint

This parameter defines the flow hold setpoint value. It is effective for the following <ControlMode> (0#40):

13 Flow control
14 p/flow control

⇨ Chapter "7.1.1 Object 0#40: Control mode", page 119

The <FlwHoldSetpoint> (70#6) acts as setpoint value in case of <StatusWord> (0#38) equals 'HOLD' or 'FAULT_HOLD'.
⇨ Chapter "5.2.3 Object 0#38: Status word", page 48

DriveFlowControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
70#6	FlwHoldSetpoint	0x5314#1	INT32	rw	Y	INT32	0
70#7	Unit	0x5314#2	UINT8	ro	-	UINT8	0
70#8	Prefix	0x5314#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

6.3 Drive transducer interface

There are seven available transducers interfaces which can be used to convert actual values to be forwarded to the controller:

- Analog input 0…4, used for pressure, position or other sensors
- Encoder
- SSI
- For special application any parameter can be mapped to an interface e.g. from a field bus.

The logic to select the interface is called transducer interface. The following figure shows how routing and scaling of the actual value is done for the available inputs.

With the interface type <Type> (1#22), the type of value conditioning can be selected depending on the sensor.

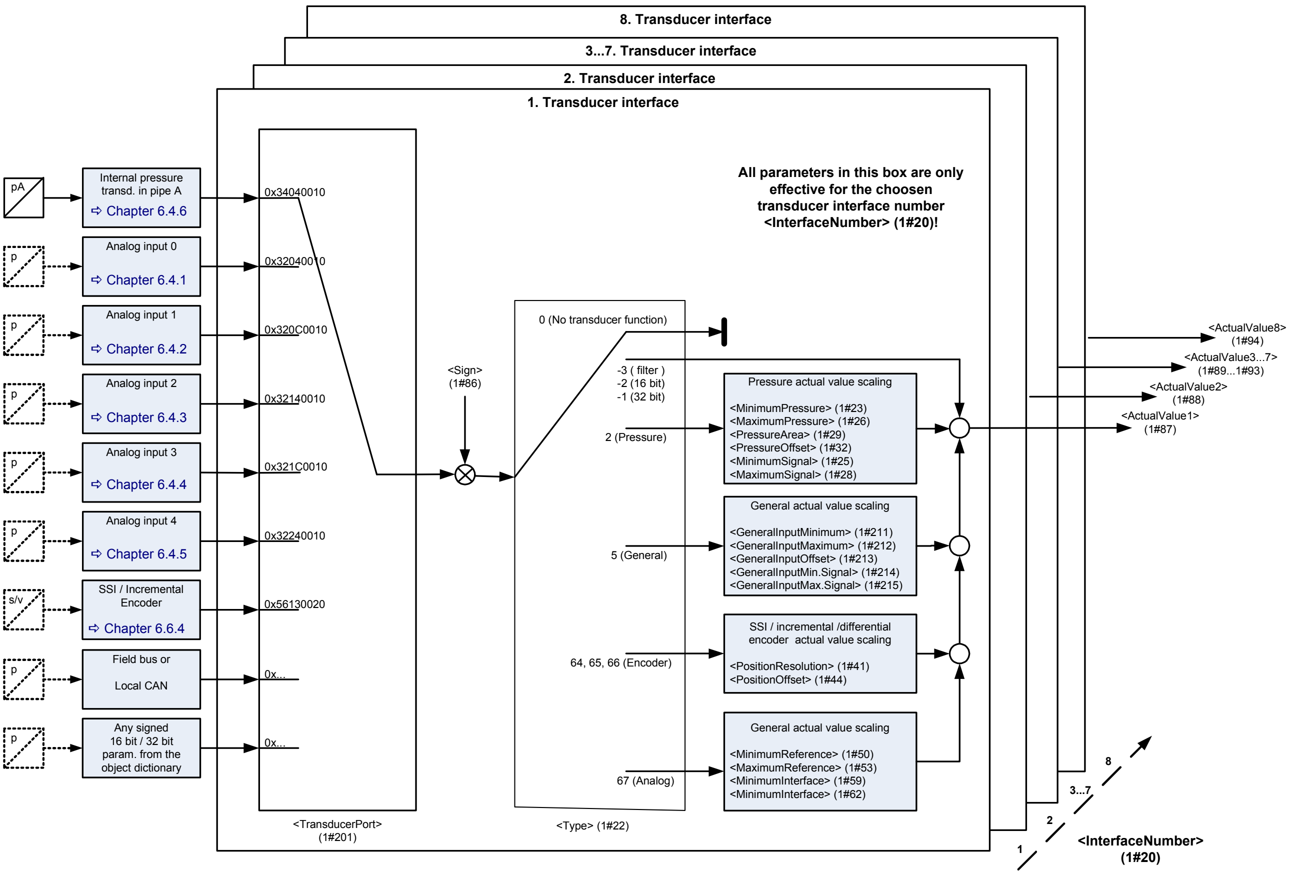


Figure 25: Drive actual value path

ⓘ Per default, analog input 0 and 1 are used for the analog setpoint values. Therefore the inputs 2…4 are normally used for actual values.
➪ Chapter "6.4 Analog inputs", page 81

ⓘ Before reading or writing configuration values of a particular transducer interface it is necessary to select the particular interface by setting the interface number <InterfaceNumber> (1#20).

When changing the <InterfaceNumber> (1#20), the following parameters represent the configuration of the selected interface.

Index	Object Name
1#21	Max interface number
1#20	Interface number
1#22	Type
1#86	Sign
1#83	Actual value
1#201	Transducer port
1#23	Minimum pressure
1#26	Maximum pressure
1#29	Pressure area
1#32	Pressure offset
1#25	Minimum transducer signal
1#28	Maximum transducer signal
1#41	Position resolution
1#44	Position offset
1#50	Minimum reference
1#53	Maximum reference
1#59	Minimum interface
1#62	Maximum interface

6.3.1 Object 1#87: Actual value 1

This parameter contains the output value of the transducer interface 1.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#87	ActualValue1	0x6210#1	INT32	ro	-	INT32	None

6.3.2 Object 1#88: Actual value 2

This parameter contains the output value of the transducer interface 2.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#88	ActualValue2	0x6211#1	INT32	ro	-	INT32	None

6.3.3 Object 1#89: Actual value 3

This parameter contains the output value of the transducer interface 3.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#89	ActualValue3	0x6212#1	INT32	ro	-	INT32	None

6.3.4 Object 1#90: Actual value 4

This parameter contains the output value of the transducer interface 4.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#90	ActualValue4	0x6213#1	INT32	ro	-	INT32	None

6.3.5 Object 1#91: Actual value 5

This parameter contains the output value of the transducer interface 5.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#91	ActualValue5	0x6214#1	INT32	ro	-	INT32	None

6.3.6 Object 1#92: Actual value 6

This parameter contains the output value of the transducer interface 6.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#92	ActualValue6	0x6215#1	INT32	ro	-	INT32	None

6.3.7 Object 1#93: Actual value 7

This parameter contains the output value of the transducer interface 7.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#93	ActualValue7	0x6216#1	INT32	ro	-	INT32	None

6.3.8 Object 1#94: Actual value 8

This parameter contains the output value of the transducer interface 8.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#94	ActualValue8	0x6217#1	INT32	ro	-	INT32	None

6.3.9 Transducer interface definition

The assignment of an actual value source to an interface is done by setting the parameter <TransducerPort> (1#201). The type of the actual value conditioning is configured by the parameter <Type> (1#22). All parameters configuring the interface only apply to the interface selected by the parameter <InterfaceNumber> (1#20).

6.3.9.1 Object 1#21: Max interface number

This parameter indicates the number of available transducer interfaces in the servo valve.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#21	MaxInterfaceNumber	0x6200#0	UINT8	ro	-	UINT8	8

6.3.9.2 Object 1#20: Interface number

This parameter defines the actual referenced interface.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#20	InterfaceNumber	0x6101#0	UINT8	rw	N	1…48	None

Value description

<InterfaceNumber>	**Description**
1	1st interface selected for configuration
2	2nd interface selected for configuration
3	3rd interface selected for configuration
4	4th interface selected for configuration
5	5th interface selected for configuration
6	6th interface selected for configuration
7	7th interface selected for configuration
8	8th interface selected for configuration
All other values	Reserved

Table 22: Possible values of parameter <InterfaceNumber> (1#20)

6.3.9.3 Object 1#22: Type

This interface type <Type> (1#22) defines the method of the value conditioning.
This parameter setting is effective for the interface selected by <InterfaceNumber> (1#20).

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#22	Type	0x6202#0	INT8	rw	N	INT8	0

Value description

<Type>	**Description**
0 (no transducer function)	Interface deactivated.
2 (pressure transducer)	Pressure sensor actual value conditioning active ➩ Chapter "6.3.10 Pressure actual value scaling", page 73
5 (general input)	General 16 or 32 bit input from the object dictionary ➩ Chapter "6.3.13 General input scaling", page 77
64 (position incremental)	Incremental encoder with A/B channels ➩ Chapter "6.3.11 Position digital encoder actual value scaling", page 75
65 (position SSI binary)	SSI possition encoder with binary coded signal ➩ Chapter "6.3.11 Position digital encoder actual value scaling", page 75
66 (position SSI gray)	SSI possition encoder with gray coded signal ➩ Chapter "6.3.11 Position digital encoder actual value scaling", page 75
67 (position analog)	Use analog input as position signal ➩ Chapter "6.3.12 Position analog encoder actual value scaling", page 76
–1 (position direct 32 bit)	32 bit signal with no further scaling ➩ Chapter "6.3.11 Position digital encoder actual value scaling", page 75
–2 (position direct 16 bit)	16 bit signal with no further scaling ➩ Chapter "6.3.11 Position digital encoder actual value scaling", page 75
All other values	Not used.

Table 23: Possible values of parameter <Type> (1#22)

6.3.9.4 Object 1#86: Sign

This parameter defines the sign of the actual value.
This parameter setting is effective for the interface selected by <InterfaceNumber> (1#20).

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#86	Sign	0x6203#0	INT8	rw	N	–1 or +1	1

Value description

<Sign>	**Description**
–1	Negative
1	Positive
All other values	Reserved

Table 24: Possible values of parameter <Sign> (1#86)

6.3.9.5 Object 1#83: Actual value

Compared to the objects <Actual value 1…8> (1#87…1#94), this parameter contains the output value of the actual selected interface.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#83	ActualValue	0x6204#0	INT32	ro	-	INT32	None

6.3.9.6 Object 1#201: Transducer port

This parameter defines the transducer port where the actual physical values are coming from. The port is defined by a CANopen index, sub-index and length.
This parameter setting is effective for the interface selected by <InterfaceNumber> (1#20).

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#201	TransducerPort	0x4032#0	UINT32	rw	N	UINT32	None

Value description

<TransducerPort>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length
Default	0x00	0x00	0x00	0x00

Table 25: Possible values of parameter <TransducerPort> (1#201)

This pointer contains a combination of index, sub-index and length of the parameter to be used.

There are 6 analog inputs (0…4 and the internal pressure transducer) and one input for a position encoder. Each input has its actual value parameter. Each actual value can be assigned to an interface by the <TransducerPort> (1#201) parameter as described in the examples below.

If an incremental encoder or an SSS encoder is selected (<Type> (1#22) = 64, 65, 66), then the <TransducerPort> (1#201) is fixed to 0x56130020 which is the index of the encoder input <EncoderValue> (69#10).

Connector	**Parameter**	**CANopen index of the parameter**	**CANopen sub-index of the parameter**	**Parameter length**	**Resulting <TransducerPort> value**
Analog input 0	<ActualValue0> (74#1)	0x3204	0x00	0x10	0x32040010
Analog input 1	<ActualValue1> (74#3)	0x320C	0x00	0x10	0x320C0010
Analog input 2	<ActualValue2> (75#1)	0x3214	0x00	0x10	0x32140010
Analog input 3	<ActualValue3> (75#4)	0x321C	0x00	0x10	0x321C0010
Analog input 4	<ActualValue4> (75#7)	0x3224	0x00	0x10	0x32240010
Internal pressure transducer	<ActualValue> (74#14)	0x3404	0x00	0x10	0x34040010
Position encoder	<Encoder-Value> (69#10)	0x5613	0x00	0x20	0x56130020

6.3.9.7 Object None: Drive transducer structure

The <DriveTransducerStructure> is only available via CAN SDO. There is no access via Profibus for the data type domain.

This parameter stores the interface parameterization for all four interfaces in an internal data structure. It is to be used only to store or transfer a configuration from one servo valve to another.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
None	DriveTransducerStructure	0x3271#0	DOMAIN	rw	Y	None	

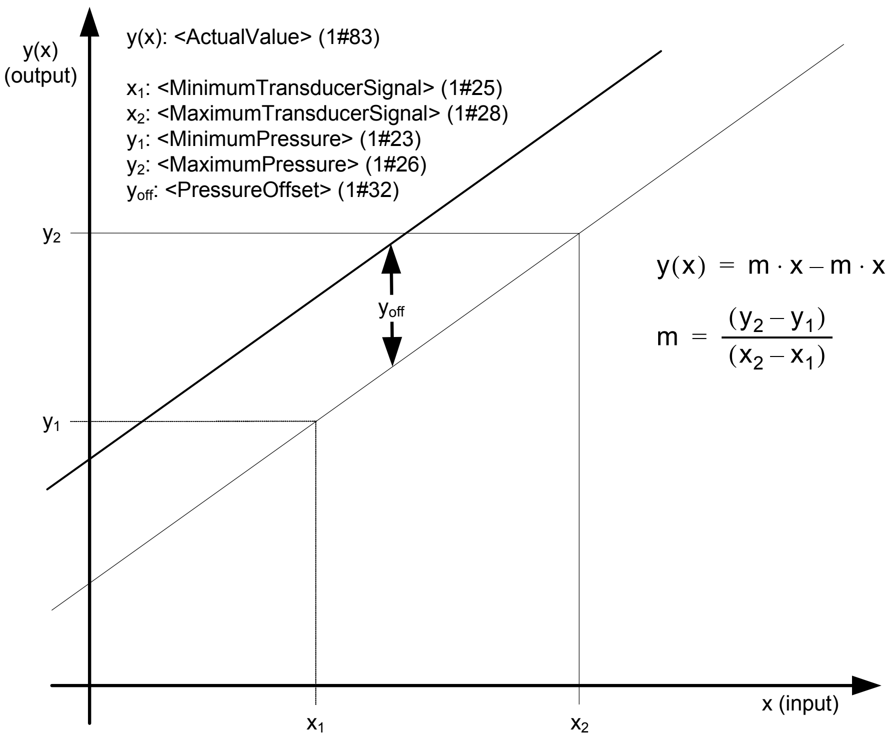
This parameter is for Moog internal use only.
The structure and length may change in the future.

6.3.10 Pressure actual value scaling

To activate the pressure actual value scaling, the interface type needs to be configured to "pressure transducer". This is done by writing the value 2 to the parameter <Type> (1#22).

This parameter setting is effective for the interface selected by <InterfaceNumber> (1#20).

⇨ Chapter "6.3.9.3 Object 1#22: Type", page 71



$$y(x) = m \cdot x - m \cdot x_1 + y_1 + y_{off}$$

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

Figure 26: Pressure actual value scaling

6.3.10.1 Object 1#23: Minimum pressure

This parameter defines the minimum transducer signal (when pressure offset equals 0).

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#23	MinimumPressure	0x6220#1	INT32	rw	N	INT16	16384

6.3.10.2 Object 1#26: Maximum pressure

This parameter defines the maximum transducer signal (when pressure offset equals 0).

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#26	MaximumPressure	0x6221#1	INT32	rw	N	INT16	16384

6.3.10.3 Object 1#25: Minimum transducer signal

This parameter defines the transducer signal when the pressure is minimal (when pressure offset equals 0).

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#25	MinimumTransducerSignal	0x6224#1	INT32	rw	N	INT16	16384

6.3.10.4 Object 1#28: Maximum transducer signal

This parameter defines the transducer signal when the pressure is maximal (when pressure offset equals 0).

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#28	MaximumTransducerSignal	0x6225#1	INT32	rw	N	INT16	16384

6.3.10.5 Object 1#29: Pressure area

This object shall provide the cylinder area corresponding to a pressure transducer. In Moog firmware this parameter consists, but is not used for any calculation.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#29	PressureArea	0x6222#1	INT32	rw	N	INT16	0

6.3.10.6 Object 1#32: Pressure offset

This parameter defines a pressure offset which is added to the two point scaling function defined by the four parameters before.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#32	PressureOffset	0x6223#1	INT32	rw	N	INT16	0

6.3.11 Position digital encoder actual value scaling

To activate the position digital encoder actual value scaling, the interface type needs to be configured to "position direct 16 bit","position direct 32 bit","position incremental", "position SSI binary" or "position SSI gray". This is done by writing the value –2, –1, 64, 65 or 66 to the parameter <Type> (1#22).

This parameter setting is effective for the interface selected by <InterfaceNumber> (1#20).

⇨ Chapter "6.3.9.3 Object 1#22: Type", page 71

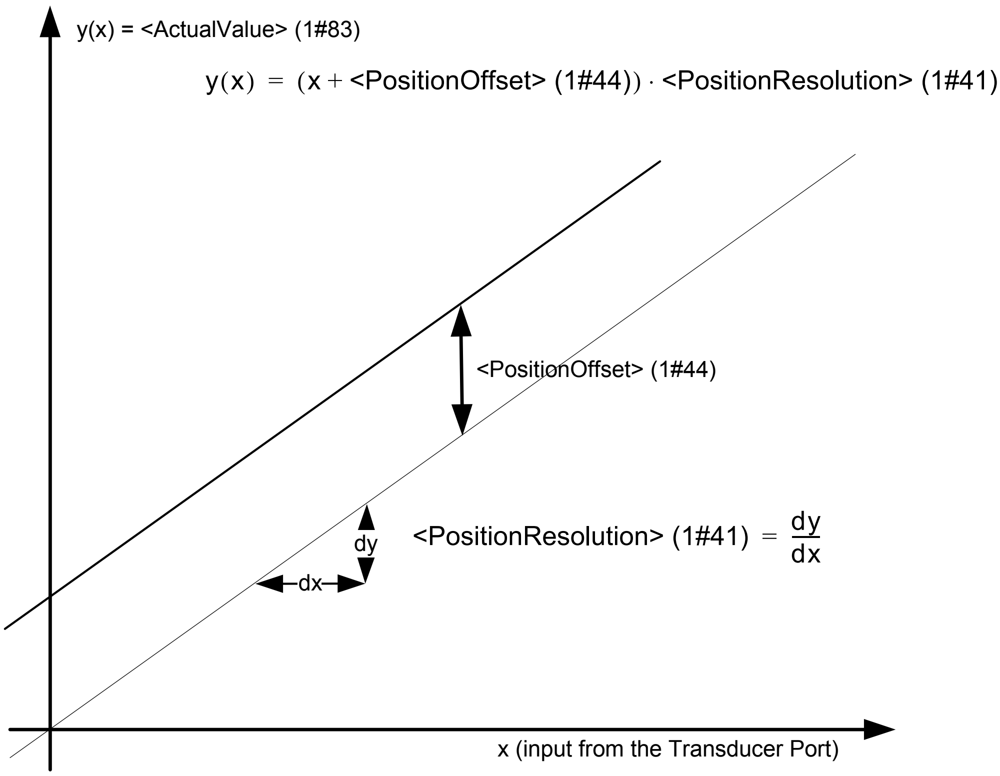


Figure 27: Position digital encoder actual value scaling

6.3.11.1 Object 1#41: Position resolution

This parameter defines the transducer siganl when the position is maximal.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#41	PositionResolution	0x6240#1	INT32	rw	N	INT32	1

6.3.11.2 Object 1#44: Position offset

This parameter defines a position offset which is added to the two point scaling function defined by the four parameters before.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#44	PositionOffset	0x6241#1	INT32	rw	N	INT32	0

6.3.12 Position analog encoder actual value scaling

To activate the position analogue encoder actual value scaling the interface type needs to be configured to "position analog". This is done by writing the value 67 to the parameter <Type> (1#22).

This parameter setting is effective for the interface selected by <InterfaceNumber> (1#20).

⇨ Chapter "6.3.9.3 Object 1#22: Type", page 71

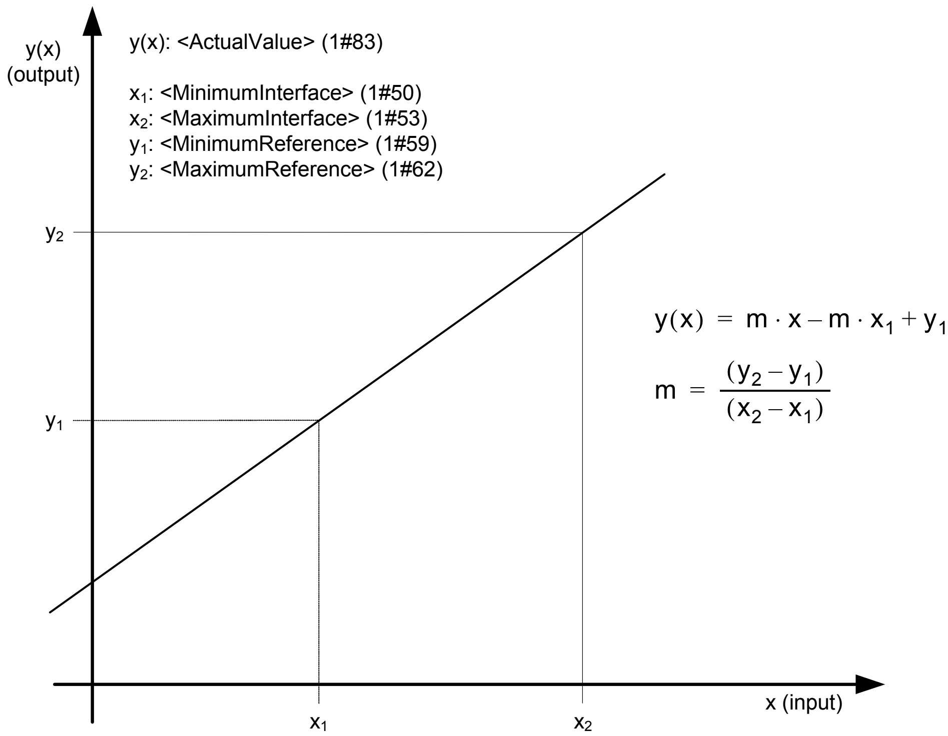


Figure 28: Position analog encoder actual value scaling

6.3.12.1 Object 1#59: Minimum reference

This parameter defines the transducer signal when position is minimal.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#59	MinimumReference	0x6233#1	INT32	rw	N	INT32	–1

6.3.12.2 Object 1#62: Maximum reference

This parameter defines the transducer signal when position is maximal.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#62	MaximumReference	0x6234#1	INT32	rw	N	INT32	1

6.3.12.3 Object 1#50: Minimum interface

This parameter defines the minimum interface signal.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#50	MinimumInterface	0x6230#1	INT32	rw	N	INT32	–1

6.3.12.4 Object 1#53: Maximum interface

This parameter defines the maximum interface signal.

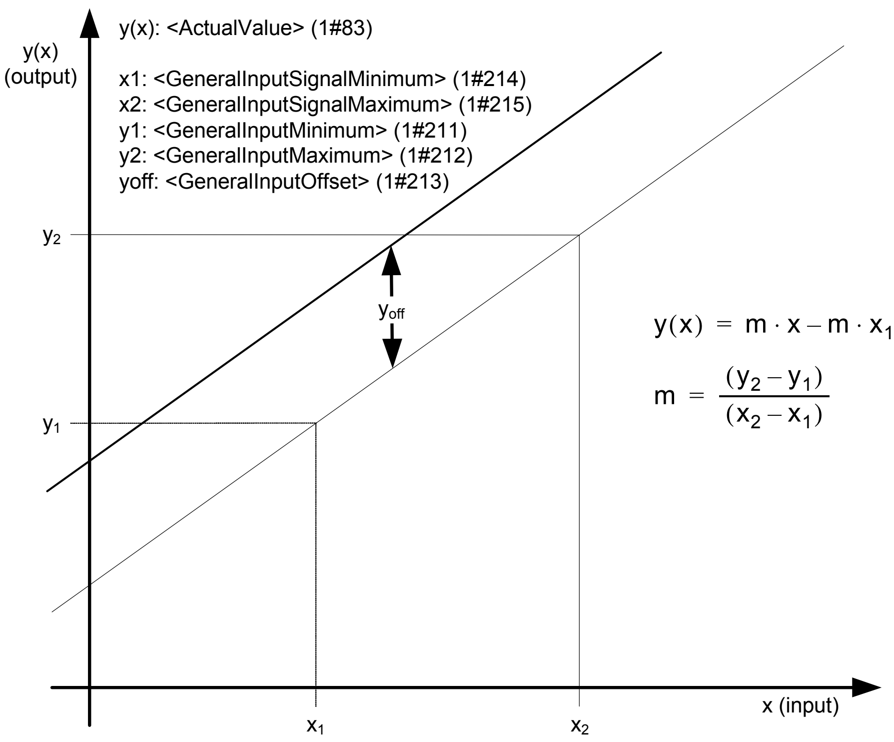
Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#53	MinimumInterface	0x6231#1	INT32	rw	N	INT32	1

6.3.13 General input scaling

The general input is used to scale an INT32 input or an INT16 input. Example: an external pressure transducer with CAN interface is mapped to the receive PDO <Integer32> (71#113). To activate the general input scaling, the interface type needs to be configured to "general input". This is done by writing the value 5 to the parameter <Type> (1#22).

This parameter setting is effective for the interface selected by <InterfaceNumber> (1#20).

➪ Chapter "6.3.9.3 Object 1#22: Type", page 71



$$y(x) = m \cdot x - m \cdot x_1 + y_1 + y_{off}$$

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

Figure 29: General input scaling

6.3.13.1 Object 1#211: General input minimum

General input means input to the controller = output of the scaling. This parameter defines the minimum output of the scaling (without offset). Example: GeneralInputMinimum (1#211) = 0 % = 0.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#211	GeneralInputMinimum	0x6228#1	INT32	rw	N	INT32	0

6.3.13.2 Object 1#212: General input maximum

General input means input to the controller = output of the scaling. This parameter defines the maximum output of the scaling (without offset). Example: GeneralInputMaximum (1#212) = 100 % = 16384.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#212	GeneralInputMaximum	0x6229#1	INT32	rw	N	INT32	16384

6.3.13.3 Object 1#214: General input signal minimum

This parameter defines the input signal of the scaling while the output of the scaling has its minimum (without offset). Example: GeneralInputSignalMinimum (1#214) = input voltage for 0 %.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#214	GeneralInputSignalMinimum	0x622C#1	INT32	rw	N	INT32	0

6.3.13.4 Object 1#215: General input signal maximum

This parameter defines the input signal of the scaling while the output of the scaling has its maximum (without offset). Example: GeneralInputSignalMaximum (1#215) = input voltage for 100 %.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#215	GeneralInputSignalMaximum	0x622D#1	INT32	rw	N	INT32	16384

6.3.13.5 Object 1#213: General input offset

This parameter defines the offset which is added to the two point scaling function defined by the four parameters before.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#213	GeneralInputOffset	0x622B#1	INT32	rw	N	INT32	0

6.3.14 Parameterization examples

This chapter gives an example how to configure one complete pressure transducer interface.

6.3.14.1 Get active transducer interface number and output value

The <InterfaceNumber> (1#20) shows the active transducer interface number. The output value of the active transducer interface can be read from the parameter <ActualValue> (1#83).

6.3.14.2 Example 1: Enable/disable transducer interface

1. Select the transducer interface which is intended to be enabled or disabled.
   Therefore set the <InterfaceNumber> (1#20).
2. Enable or disable the selected transducer interface and select the method of conditioning.
   Therefore set the interface type:
   <Type> (1#22) to 0 (no transducer function) or
   <Type> (1#22) to 2 (pressure transducer) or
   <Type> (1#22) to 5 (general input) or
   <Type> (1#22) to 64 (position incremental) or
   <Type> (1#22) to 65 (position SSI binary) or
   <Type> (1#22) to 66 (position SSI gray) or
   <Type> (1#22) to 67 (positon analog) or
   <Type> (1#22) to –1 (analog direct with 32 bit) or
   <Type> (1#22) to –2 (analog direct with 16 bit)

6.3.14.3 Example 2: Change sign of the transducer signal

1. Select the transducer interface which is intended to change the sign.
   Therefore set the <InterfaceNumber> (1#20).
2. Change the sign of the transducer interface.
   Therefore set the sign:
   <Sign> (1#86) to 1 (positive) or
   <Sign> (1#86) to –1 (negative)

6.3.14.4 Example 3: Adjust transducer interface without scaling

1. Select the transducer interface which is intended to be adjusted.
   Therefore set the <InterfaceNumber> (1#20).
2. Define the input signal for the transducer interface with the input parameter address.
   Get the CANopen index, sub-index and parameter length in bits from the object dictionary or from the parameter description.
   For example, the parameter <dums16> (71#112) should be used as input:

   Index: 0x0003
   Index MSB: 0x00
   Index LSB: 0x03
   Sub-index: 0x00
   Parameter bit length: 0x10
   Only parameters with a bit length of 0x10 are allowed to be mapped!

   Build the address value in the following manner:

Byte	3	2	1	0	Result
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter length in bit: 0x10	
Example	0x00	0x03	0x00	0x10	0x00030010

   Write the result 0x00030010 into the parameter <TransducerPort> (1#201).
3. Check the transducer interfaces <Sign> (1#86) and change the value (1 or –1) if needed.
4. Set the transducer interface <Type> (1#22) to –2 (analog direct).

6.3.14.5 Example 4: Adjust transducer interface with scaling

1. Select the transducer interface which is intended to be adjusted.
   Therefore set the <InterfaceNumber> (1#20).
2. Define the input signal for the transducer interface with the input parameter address.
   Get the CANopen index, sub-index and parameter length in bits from the object dictionary or from the parameter description.
   For example, the analog input 2 <ActualValue2> (1#88) should be used as input:

Index:	0x3214
Index MSB:	0x32
Index LSB:	0x14
Sub-index:	0x00
Parameter bit length:	0x10 Only parameters with a bit length of 0x10 are allowed to be mapped!

   Build the address value in the following manner:

Byte	3	2	1	0	Result
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length: 0x10	
Example	0x32	0x14	0x00	0x10	0x32140010

   Write the result 0x00030010 in the parameter <TransducerPort> (1#201).
3. Check transducer interfaces <Sign> (1#86) and change value (1 or –1) if needed.
4. Set transducer interface <Type> (1#22) to 2 (pressure transducer).
5. Set the <PressureOffset> (1#32) to 0.
6. Define the scaling of the linear function $y(x) = m \cdot x + b$ by using two points with their coordinates (x_1, x_2, y_1, y_2). The y values correspond to the output (normally the pressure) and the x values correspond to the mapped input signal.

   x_1: <MinimumTransducerSignal> (1#35)
   x_2: <MaximumTransducerSignal> (1#38)
   y_1: <MinimumPressure> (1#23)
   y_2: <MaximumPressure> (1#26)

6.4 Analog inputs

The following figure shows the available inputs and the physical connector names. All analog to digital converters have the same resolution of 12 bit.

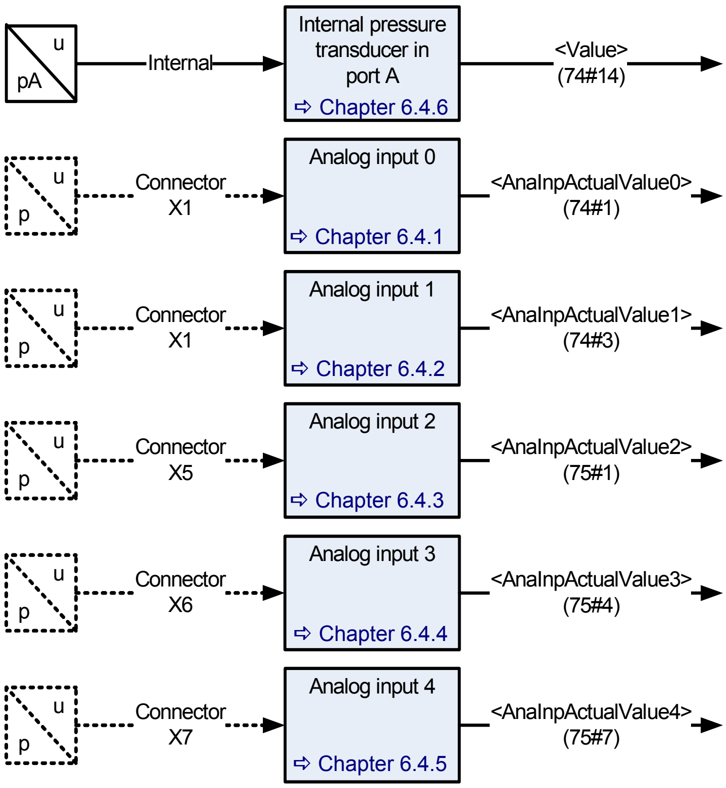


Figure 30: Analog inputs

⇨ Document CA63420-001 "User Manual Electrical Interfaces".

The analog input 0 is intended for the spool position setpoint value and the analog input 1 is intended for the pressure setpoint value. The analog inputs 0 and 1 will only be routed to the demand value generators, if the <DeviceMode> (0#39) is set to 2 (setpoint input locally). The setpoint values are provided by the field bus, if the <DeviceMode> (0#39) is set to 1 (setpoint input via bus). In this case the analog inputs 0 and 1 can be used as additional inputs for external transducers.

⇨ Chapter "6.2.3 Spool position setpoint value path", page 54

An external transducer on analog input 0, 1, 2, 3 or 4 can be scaled and mapped as input for the controller by using the transducer interface.

⇨ Chapter "6.3.9 Transducer interface definition", page 69

The analog inputs 0 and 1 are only effective as setpoint value inputs, if the <DeviceMode> (0#39) is set to 2 (setpoint input locally).

6.4.1 Analog input 0

6.4.1.1 Object 74#2: Input type

This input type describes the supported electrical signal for the analog input 0.

AnalogInput0							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#2	AnalnpType0	0x3200#0	INT8	rw	Y	INT8	1

Value description

<InputType>		**Value range of <ActualValue>**		
Potential-free	**Grounded**	**0 % or –100 %**	**+100 %**	**Value range of electrical signal**
1	9 not available	–16384	16384	±10 V (±100 %)
2	10 not available	0	16384	0…10 V (0…100 %)
3	6 not available	–16384	16384	±10 mA (±100 %)
4	7 not available	0	16384	0…10 mA (0…100 %)
5	8 not available	0	16384	4…20 mA (0…100 %)
11	12 not available	–16384	16384	4…20 mA (±100 %)

Table 26: Possible values of parameter <AnalnpType0> (74#2)

Not all possible input types may be calibrated on the servo valve! Only the ordered input types are calibrated.

6.4.1.2 Object 74#1: Actual value

Actual value of the analog input 0.

AnalogInput0							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#1	AnalnpActualValue0	0x3204#0	INT16	ro	-	INT16	None

6.4.2 Analog input 1

6.4.2.1 Object 74#4: Input type

This input type describes the supported electrical signal for the analog input 1.

AnalogInput1							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#4	AnalnpType1	0x3208#0	INT8	rw	Y	INT8	2

Value description

⇨ Table 26, page 82

6.4.2.2 Object 74#3: Actual value

Actual value of the analog input 1.

AnalogInput1							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#3	AnalnpActualValue1	0x320C#0	INT16	ro	-	INT16	None

6.4.3 Analog input 2

6.4.3.1 Object 75#2: Input type

This input type describes the supported electrical signal for the analog input 2.

AnalogInput2							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#2	AnaInpType2	0x3210#0	INT8	rw	Y	INT8	2

Value description

<InputType>		**Value range of <ActualValue>**		
Potential-free	**Grounded**	**0 % or –100 %**	**+100 %**	**Value range of electrical signal**
1 not available	9	–16384	16384	±10 V (±100 %)
2	10	0	16384	0…10 V (0…100 %)
3 not available	6 not available	–16384	16384	±10 mA (±100 %)
4	7	0	16384	0…10 mA (0…100 %)
5	8	0	16384	4…20 mA (0…100 %)
11	12	–16384	16384	4…20 mA (±100 %)

Table 27: Possible values of parameter <AnaInpType2> (75#2)

Not all possible input types may be calibrated on the servo valve! Only the ordered input types are calibrated.

6.4.3.2 Object 75#1: Actual value

Actual value of the analog input 2.

AnalogInput2							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#1	AnaInpActualValue2	0x3214#0	INT16	ro	-	INT16	None

Value description

➪ Table 27, page 83

6.4.4 Analog input 3

6.4.4.1 Object 75#5: Input type

This input type describes the supported electrical signal for the analog input 3.

AnalogInput3							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#5	AnaInpType3	0x3218#0	INT8	rw	Y	INT8	2

Value description

➪ Table 27, page 83

6.4.4.2 Object 75#4: Actual value

Actual value of the analog input 3.

AnalogInput3							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#4	AnaInpActualValue3	0x321C#0	INT16	ro	-	INT16	None

6.4.5 Analog input 4

6.4.5.1 Object 75#8: Input type

This input type describes the supported electrical signal for the analog input 4.

AnalogInput4							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#8	AnaInpType4	0x3220#0	INT8	rw	Y	INT8	2

Value description

⇨ Table 27, page 83

6.4.5.2 Object 75#7: Actual value

Actual value of the analog input 4.

AnalogInput4							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#7	AnaInpActualValue4	0x3224#0	INT16	ro	-	INT16	None

6.4.6 Internal pressure transducer input

The internal pressure transducer input is located in the servo valve port A. This transducer can also be used as input for the controller.

⇨ Chapter "6.3.9 Transducer interface definition", page 69

6.4.6.1 Object 74#14: Actual value

Actual value of the internal pressure transducer input.

PressureTransducer							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#14	AnaInpActualValue4	0x3404	INT16	ro	-	INT16	None

6.5 Analog outputs

The servo valve has two analog outputs that can have one of the following two types:

- 4…20 mA (referenced to supply ground)
- 2…10 V (referenced to supply ground)

Which of these two types is available depends on the type designation number 10 in the order code.

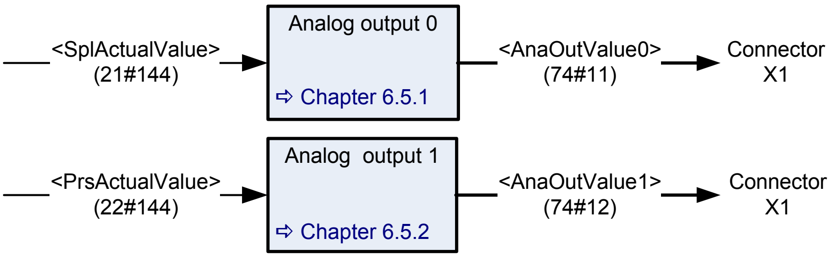
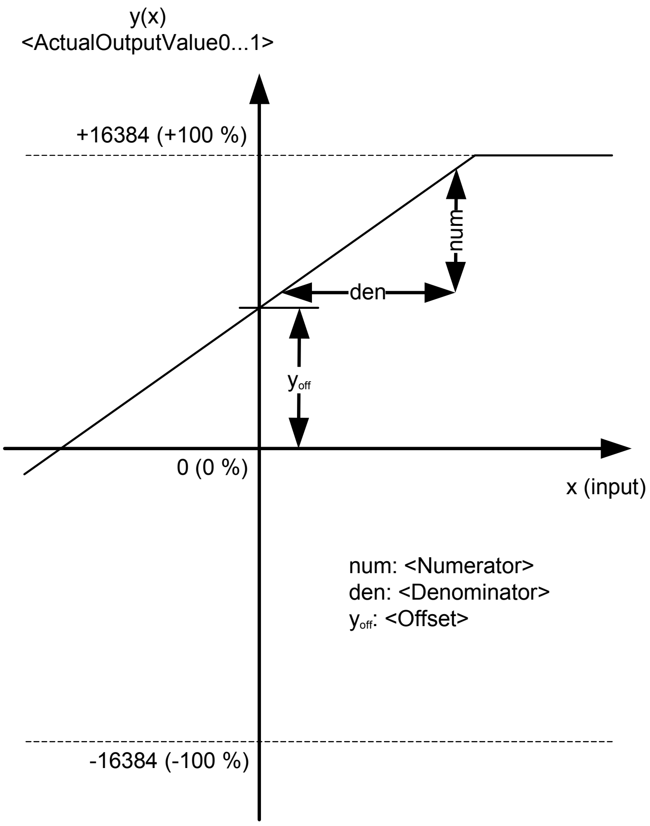


Figure 31: Analog outputs

⇨ Document CA63420-001 "User Manual Electrical Interfaces"

Each output can be scaled by the customer. The scaling is done according to the following formula:



$$y(x) = m \cdot x + y_{off}$$

$$m = \frac{num}{den}$$

Figure 32: Analog output scaling

6.5.1 Analog output 0

6.5.1.1 Object 74#5…7: Scaling

AnalogOutput0							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#5	AnaOutScaNumerator0	0x3244#1	INT16	rw	Y	INT16	16384
74#6	AnaOutScaDenominator0	0x3244#2	INT16	rw	Y	INT16	16384
74#7	AnaOutScaOffset0	0x3244#3	INT16	rw	Y	INT16	0

6.5.1.2 Object 74#11: Actual value

AnalogOutput0							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#11	AnaOutValue0	0x3245#0	INT16	ro	-	Depending on <AnaOutType0> (74#17)	None

6.5.1.3 Object 74#15: Mapping parameter

Every 16 bit parameter can be mapped to the analog output 0.

AnalogOutput0							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#15	AnaOutMappingParameter0	0x3240#0	UINT32	rw	Y	UINT32	0x63010110

Value description

<Parameter>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length: 0x10
Default	0x63	0x01	0x01	0x10

Table 28: Possible values of parameter <AnaOutMappingParameter0> (74#15)

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

The default value is 0x63010110, which refers to the <SplActualValue> (21#144…146), with the CANopen index 0x6301 and the CANopen sub-index 0x01 with a length of 16 bit (16=0x10).

6.5.1.4 Object 74#17: Type

AnalogOutput0							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#17	AnaOutType0	0x3243#0	UINT8	rw	-	0…1	0

Value description

<Type>	<AnaOutValue0> (74#11) range	Output signal range
0	–16384…16384	4…20 mA / 2…10 V (depending on hardware version)
1	0…16384	4…20 mA / 2…10 V (depending on hardware version)

Table 29: Possible values of parameter <AnaOutType0> (74#17)

6.5.2 Analog output 1

6.5.2.1 Object 74#8…10: Scaling

AnalogOutput1							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#8	AnaOutScaNumerator1	0x3265#1	INT16	rw	Y	INT16	16384
74#9	AnaOutScaDenominator1	0x3265#2	INT16	rw	Y	INT16	16384
74#10	AnaOutScaOffset1	0x3265#3	INT16	rw	Y	INT16	0

6.5.2.2 Object 74#12: Actual value

AnalogOutput1							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#12	AnaOutValue1	0x3266#0	INT16	ro	-	Depending on <AnaOutType0> (0x3263)	None

6.5.2.3 Object 74#16: Mapping parameter

Every 16 bit parameter can be mapped to the analog output 1.

AnalogOutput1							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#16	AnaOutMappingParameter1	0x3260#0	UINT32	rw	Y	UINT32	0x63810110

Value description

<Parameter>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length: 0x10
Default	0x63	0x81	0x01	0x10

Table 30: Possible values of parameter <AnaOutMappingParameter1> (74#16)

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

The default value is 0x63810110, which refers to the <PrsActualValue> (22#144…146), with the CANopen index 0x6381 and the CANopen sub-index 0x01 with a length of 16 bit (16=0x10).

6.5.2.4 Object 74#18: Type

AnalogOutput1							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#18	AnaOutType0	0x3263#0	UINT8	rw	-	0…1	0

Value description

<Type>	**<AnaOutValue1> (74#12) range**	**Output signal range**
0	–16384…16384	4…20 mA / 2…10 V (depending on hardware version)
1	0…16384	4…20 mA / 2…10 V (depending on hardware version)

Table 31: Possible values of parameter <AnaOutType0> (74#18)

6.6 Encoder input

The servo valve has one encoder input.

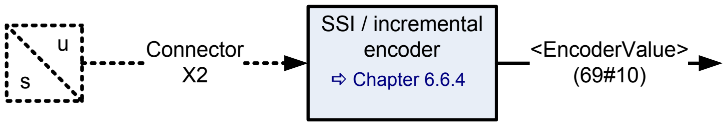


Figure 33: Encoder input

➪ See also document CA63420-001 "User Manual Electrical Interfaces"

The encoder input can be used to measure the position of a hydraulic axis. This position will be used by the axis position controller in order to close the control loop. The parameter <Type> (1#22) will select the type of the encoder (SSI binary, SSI gray or incremental)
It is possible to map the output of the encoder <EncoderValue> (69#10) to more than one <TransducerPort> (1#201) . In this case the first of this transducer ports is used to initialize the encoder hardware.

➪ Chapter "6.3.9.3 Object 1#22: Type", page 71
➪ Chapter "6.3.9.6 Object 1#201: Transducer port", page 72

6.6.1 Object 69#10: Encoder value

This raw value is the output from the encoder. It can be mapped to a transducer interface to be scaled to the needs of the application.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#10	EncoderValue	0x5613	UINT32	rw	N	UINT32	None

6.6.2 Object 69#15: Encoder set value

This value is the input of the encoder. With this parameter the encoder value can be set (for example, with z-pulse, during reference mode, setting an offset).

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#15	EncoderSetValue	0x5621	INT32	rw	Y	INT32	0

6.6.3 Object 69#1: Sensor supply enable

The valve can deliver the power supply for the encoder. It is possible to switch the supply voltage off, to 5 V or to 24 V to match the needs of the encoder hardware. The accuracy of the 24 V depends on the valve's power supply, the 5 V is regulated inside the servo valve.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#1	SensorSupplyEnable	0x5619	UINT8	rw	Y	0, 5, 24	0

Value description

<SensorSupplyEnable>	**Description**
0	0 V, supply disabled.
5	5 V
24	24 V

Table 32: Possible values of parameter <SensorSupplyEnable> (69#1)

6.6.4 Incremental Encoder

The incremental encoder only can measure a relative change in the position and the speed of the axis. To get the absolute position the encoder needs a reference run after power on.

The valve allows an automatically reference run. To start this reference run, switch the <DeviceMode> (0#39) to 4 (reference mode) and set bit 9 (axis install mode positive direction) or bit 10 (axis install mode negative direction) of the <ControlWord> (0#37).

In the first step of the reference run, axis will move with the <ReferencingVelocity> (69#2) to end of stroke. When the "end of stroke position" is reached, the pressure in the cylinder is rising. So the valve will wait until <PrsActualValue> (22#144) reaches the <ReferencingForce> (69#3) to finish the first step.

In the second step of the reference run, the axis is moving slowly back from the "end of stroke position". If <ZPulseTrigger> (69#5) = 1, the valve is waiting for the Z pulse. If the Z pulse appears or the axis moved the distance <ReferencingStop> (69#7), the reference run is finished and the <EncoderValue> (69#10) is set to 0.

A manual search for the reference position is also possible if the <DeviceMode> (0#39) is switched to 1 or 2 (setpoint input via bus or locally). If the flag <ZPulseEnable> (69#8) is set, the valve is waiting for the next Z pulse. When the Z pulse appears, the predefined position <ZPulseSet> (69#14) is copied to the <EncoderValue> (69#10) and <ZPulseEnable> (69#8) is reset to indicate the successful referencing.

⇨ Chapter "5.1.2 Object 0#37: Control word", page 40
⇨ Chapter "7.7 Axis position setpoint conditioning / demand value generator", page 181

6.6.4.1 Object 69#2: Referencing velocity

In the first step of the reference run, the axis will move with this velocity to end of stroke.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#2	ReferencingVelocity	0x5614	INT32	rw	Y	1000…INT32	1000

6.6.4.2 Object 69#3: Referencing force

In the first step of the reference run, axis will move to end of stroke. When the "end of stroke position" is reached, the pressure in the cylinder is rising. So the valve will wait until <PrsActualValue> (22#144) reaches the <ReferencingForce> (69#3) to finish the first step.

⇨ Chapter "7.5.1 Object 22#144…146: Actual value", page 158

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#3	ReferencingForce	0x561A	INT16	rw	Y	INT16	0

6.6.4.3 Object 69#7: Referencing stop

In the second step of the reference run, the axis is moving slowly back from the "end of stroke position". If <ZPulseTrigger> (69#5) = 1, the valve is waiting for the Z pulse. If the Z pulse appears or the axis has moved the distance <ReferencingStop> (69#7), the reference run is finished.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#7	ReferencingStop	0x561B	FLOAT32	rw	Y	FLOAT32	0.0

6.6.4.4 Object 69#5: Z pulse trigger

In the second step of the reference run, the parameter <ZPulseTrigger> (69#5) will enable the Z pulse to trigger the reference position of the encoder. If <ZPulseTrigger> (69#5) = 0, the Z pulse is ignored and will not be dedected.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#5	ZPulseTrigger	0x5617	UINT8	rw	N	0…1	0

6.6.4.5 Object 69#6: Z pulse detected

This flag will be set when the Z pulse will appear in the second step of the reference run. It will be reset on start of the reference run while the <DeviceMode> (0#39) is switched to 4 (reference mode).

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#6	ZPulseDetected	0x5611	UINT8	rw	N	0…1	0

6.6.4.6 Object 69#9: Z pulse clear

This parameter is implemented for compatibility reasons to old software. It will clear the internal Z pulse latch which is done automatically in the actual software.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#9	ZPulseClear	0x5618	UINT8	rw	N	0…1	0

6.6.4.7 Object 69#8: Z pulse enable

If this flag is set, the valve is waiting for the next Z pulse. When the Z pulse appears, the predefined position <ZPulseSet> (69#14) is copied to the <EncoderValue> (69#10) and <ZPulseEnable> (69#8) is reset to indicate the successful referencing. Only if the <DeviceMode> (0#39) is switched to 1 or 2 (setpoint input via bus or locally).

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#8	ZPulseEnable	0x5610#1	UINT8	rw	N	0…1	0

6.6.4.8 Object 69#14: Z pulse set

This is a predefined position. It will be copied to the <EncoderValue> (69#10), if a Z pulse appears while <ZPulseEnable> (69#8) is set. Only if the <DeviceMode> (0#39) is switched to 1 or 2 (setpoint input via bus or locally).

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#14	ZPulseSet	0x5612	INT32	rw	N	INT32	0

6.6.5 SSI encoder

The SSI encoder sends an absolute position to the valve. For communication the valve needs two parameters, the <BitSize> (1#69) and <BitRate> (1#209).

6.6.5.1 Object 1#210: Master slave

Only the SSI master interface is supported.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#210	MasterSlave	0x5620	UINT8	rw	N	0…1	1

Value description

<MasterSlave>	**Description**	**Clock signal**
0	Slave (not yet supported)	From the encoder.
1	Master	From the servo valve.

Table 33: Possible values of parameter <MasterSlave> (1#210)

6.6.5.2 Object 1#69: Bit size

This parameter sets the number of bits which have to be sent to the valve (the size of the SSI telegram). It depends on the type of the transducer.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#69	BitSize	0x6243	UINT8	rw	N	0…32	24

6.6.5.3 Object 1#209: Bit rate

The <BitRate> (1#209) is used to change the clock frequency for the transmission of the bits to the valve. It depends, i.e. on the length of the cable, the noise environment and the specification of the transducer.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#209	BitRate	0x561F	UINT8	rw	Y	0…7	2

Value description

<BitRate>	**SSI clock frequency**
0	5 MHz
1	2.5 MHz
2	1.25 MHz
3	625 kHz
4	312.5 kHz
5	156.25 kHz
6	78.125 kHz
7	39.0625 kHz

Table 34: Possible values of parameter <BitRate> (1#209)

6.6.5.4 Object 75#17: SSI Error Count

The <SSIErrorCount> (75#17) is used to delay the SSI Error for n SSI transmission cycles.
Default: after 3 incorrect SSI transmission cycles, the SSI Error (fault reaction 44) will be thrown.

Drive_ActualValueConditioning							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#17	SSIErrorCount	0x3252	UINT8	rw	Y	0…254	3

6.7 Digital inputs

The servo valve has one digital input.

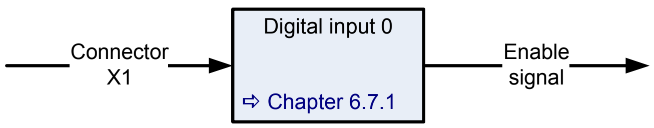


Figure 34: Digital inputs

⇨ Document CA63420-001 "User Manual Electrical Interfaces"

6.7.1 Digital input 0 (enable signal)

The digital enable signal incorporates the following functions:

- Control the device state machine (DSM).
  ⇨ Chapter "5.2.2.4.1 DSM state transitions depending on the enable signal", page 47
- Fault confirmation by toggling the digital enable signal.
  ⇨ Chapter "5.2.2.4.2 Fault confirmation with the enable signal", page 47

6.8 Digital outputs

The following digital outputs are available for the servo valves if a 11+PE connector for X1 is used. The digital output 0 is usually available on pin 8. The digital output 1 is usually available on pin 11.

Changes are possible. Please refer to the specific pin configuration of your valve.

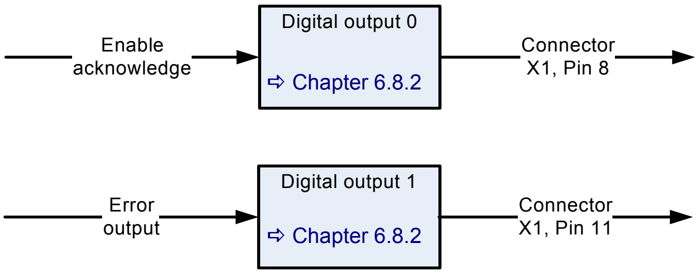


Figure 35: Digital outputs in the default configuration

⇨ Document CA63420-001 "User Manual Electrical Interfaces"

6.8.1 Object 0#218…219: Digital output setpoint

The <DigitalOutputSetpoint> (0#218…219) controls the state of the digital outputs in case the <DigitalOutputConfiguration> (0#220…221) is set to 0.

ValveDigitalOutputValue							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#218	DigitalOutputSetpoint0	0x5E42#1	INT8	rw	-	0…1	0
0#219	DigitalOutputSetpoint1	0x5E42#2	INT8	rw	-	0…1	0

6.8.2 Object 0#220…221: Digital output configuration

The behavior of the digital outputs can be configured by the parameter <DigitalOutputConfiguration> (0#220…221).

ValveDigitalOutputType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#220	DigitalOutputConfiguration0	0x5E41#1	INT8	ro	-	0…4	3
0#221	DigitalOutputConfiguration1	0x5E41#2	INT8	ro	-	0…4	0

Value description

<DigitalOutputType>	Description
0	The servo valve's digital outputs are controlled by the parameter <DigitalOutputSetpoint> (0#218…219). The pin can be used for special purposes.
1	Failsafe spool position monitoring on. The digital output is controlled by the failsafe monitoring. ⇨ Chapter "7.15.5 Failsafe monitoring", page 231
2	Error output pin. The servo valve's digital output acts according to the Device Profile Fluid Power. This means it is controlled by the device state machine (fault indication). In this case the digital output is used to indicate fault states (negative logic). ⇨ Chapter "5.2.2.5 Error output pin", page 48
3	Enable Acknowledge. The digital output gets high (24 V) if the servo valve device state machine (DSM) state is 'ACTIVE' (supply voltage is > 18 V, digital enable input is high, no fault will force the DSM to fault state, …).
4	Control Deviation Monitoring. The digital output gets high (24 V) if the 'control deviation bit 11' of the <StatusWord> (0#38) is low. That means there is no control error. ⇨ Chapter "7.15 Monitoring", page 222

Table 35: Possible values of parameter <DigitalOutputConfiguration> (0#220…221)

WARNING

Moving machine parts!

The word "failsafe" means not a personnel safety. Parts of the machine can move if the servo valve has a fault.

- If a personnel safety is needed, some additional electrical and hydraulic parts are necessary!

The availability of the digital output 1 (error output pin) depends on the servo valve model.

6.8.2.1 Object 0#209: Digital output 1 type

The <DigitalOutput1Type> (0#209) is a copy of the <DigitalOutputConfiguration1> (0#221) and is used to be compatible to old firmware versions.

ValveDigitalOutputType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#209	DigitalOutput1Type	0x2420#0	INT8	ro	-	0…4	0

6.8.3 Object 0#223…224: Digital output value

The <DigitalOutputValue> (0#223…224) shows the state of the digital outputs.

ValveDigitalOutputMonitor							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#223	DigitalOutputValue_0	0x5E44#1	UINT8	ro	-	0…1	None
0#224	DigitalOutputValue_1	0x5E44#2	UINT8	ro	-	0…1	None

6.9 Local CAN

The servo valve has a local CAN connector X2 or X10 which can be used to

- connect the Moog Valve and Pump Configuration Software to the servo valve.
- connect an external CAN device to the servo valve e.g. a position sensor or a pressure transducer.

The following CAN protocols are supported:

- Process data object (PDO) protocol
  ⇨ Chapter "6.9.2 Local CAN process data object (PDO)", page 97
- Service data object (SDO) communication
  The SDO communication will be handled by (vendor specific) parameters. With these parameters, a gateway between the field bus and the local CANs SDO channel is realized.
  ⇨ Chapter "6.9.6 Local CAN service data object (SDO) gateway", page 111
- Network management (NMT) protocol
  The NMT protocol command "start remote node" is supported.
  ⇨ Chapter "6.9.1.3 Object 73#3: Start remote node", page 96
- Synchronization (SYNC) producer protocol (implemented indirectly)
  If a synchronous transition is needed, a SYNC-telegram can be emulated using one of the transmit PDO channels.
  ⇨ Chapter "6.9.7 Local CAN Synchronization (SYNC) producer protocol emulation", page 113

If one application parameter is written by the local CAN interface and the field bus by the same cycle, the parameter last written will be processed.

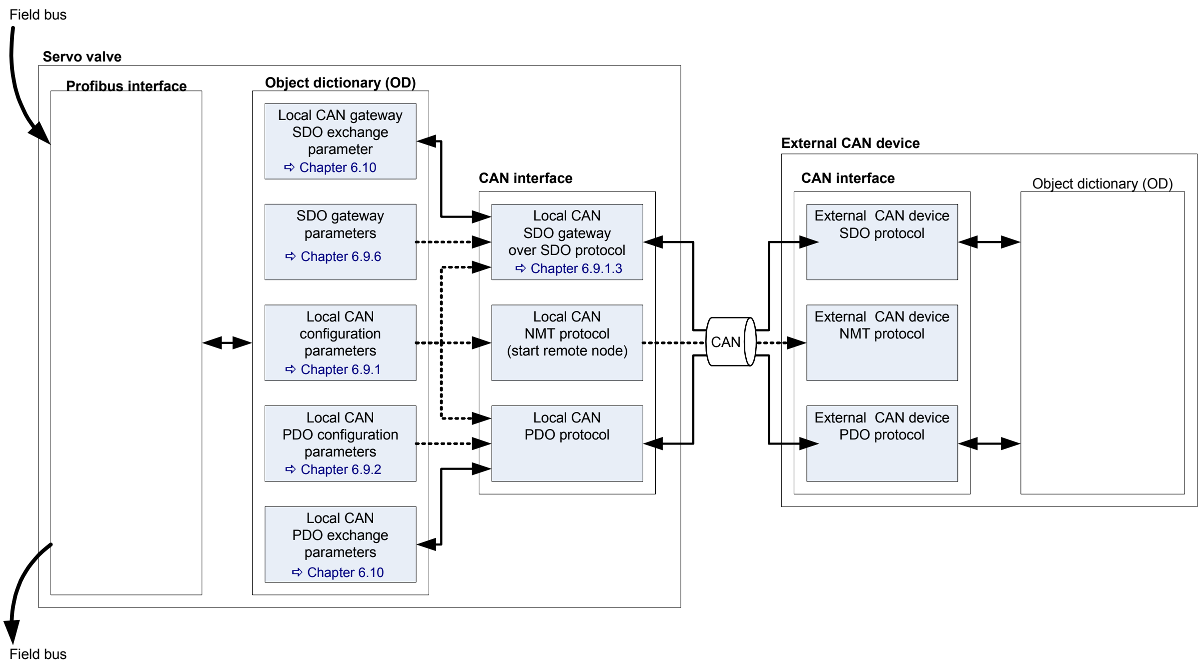


Figure 36: Structure of local CAN bus

6.9.1 Local CAN general configuration

6.9.1.1 Object 73#1: Module identifier

This parameter represents the local CAN identifier of the servo valve. A changed CAN identifier is effective after the next reset. Therefore the identifier needs to be stored.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
73#1	LocalCANModuleIdentifier	0x5B00#0	UINT8	rw	Y	1…127	127

6.9.1.2 Object 73#2: Bit rate

The local CAN bit rate in bits per second.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
73#2	LocalCANBitrate	0x5B01#0	UINT32	rw	Y	0…1000000	500000

6.9.1.3 Object 73#3: Start remote node

To start cyclic communication of the external CAN device (i.e., a pressure transducer with CAN interface), the network management (NMT) state machine of the external CAN device must be changed to the state 'Operational'. To do this, the external CAN device CAN node identifier must be written to the parameter <LocalCANStartRemoteNode> (73#3). To change all devices to the state 'Operational', write the node identifier 0 to this parameter.

⇨ Document CiA 301 "CANopen application layer and communication profile"

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
73#3	LocalCANStartRemoteNode	0x5B02#0	UINT8	rw	N	UINT8	None

ⓘ The PDO communication cannot be used before the external CAN device NMT machine is set to 'Operational'!

6.9.1.4 Object 73#110: TxPDO trigger

Writing this parameter triggers a single TxPDO. This can be used when a PDO should be transmitted on request only. To trigger one of the four TxPDOs, the parameter <LocalCANTPdoTrigger> (73#110) value must be set to the PDO number which is to be sent.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
73#110	LocalCANTPdoTrigger	0x5B03#0	UINT8	rw	N	0…4	None

6.9.1.5 Object 73#114: Termination resistor

If the servo valve is at the end of the local CAN bus a 120 Ω resistor can terminate the CAN bus. To switch on the resistor set <LocalCANTerminationResistor> (73#114) to 1.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
73#114	LocalCANTerminationResistor	0x5B14#0	UINT8	rw	Y	0…1	0

6.9.2 Local CAN process data object (PDO)

The process data object (PDO) communication allows sending and receiving parameters in real time. Different transmission modes are available: synchronous or event or timer driven transmission.

Four transmit and four receive PDOs are implemented:

- Receive process data object (RxPDO) protocol
  ⇨ Chapter "6.9.3 Receive process data object (RxPDO) configuration", page 98
- Transmit process data object (TxPDO) protocol
  ⇨ Chapter "6.9.4 Transmit process data object (TxPDO) configuration", page 104

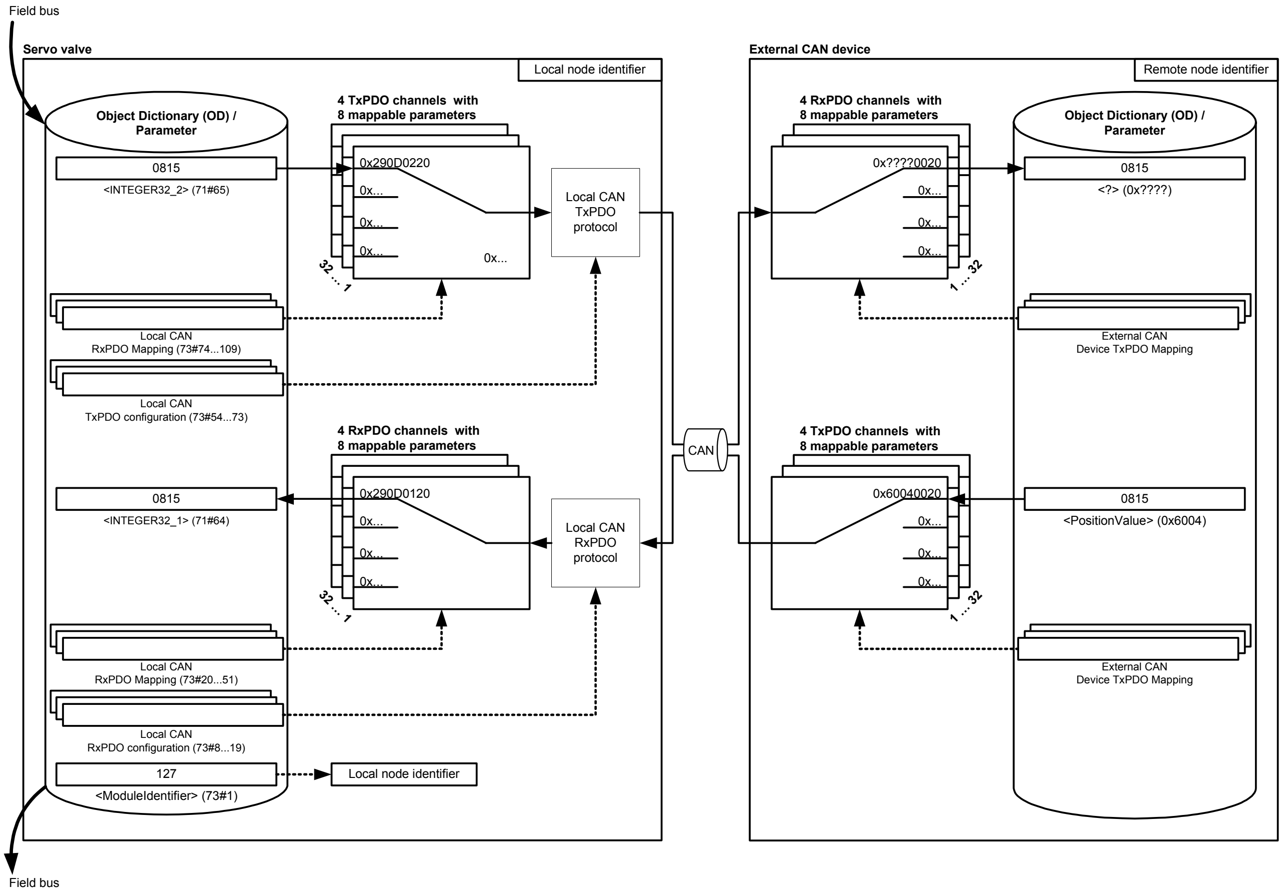


Figure 37: Local CAN process data object (PDO) configuration

6.9.3 Receive process data object (RxPDO) configuration

To enable PDO reception, the local application parameters must be mapped to the receive PDO and a transmission type must be selected for each PDO channel. Four receive PDO channels are available.

The parameters described in the ⇨ Chapter "6.10 Free to use parameters", page 114 can be used to transmit values to the external CAN device.

If the external CAN device uses the CAN-IDs according to the CiA 301, the CAN object identifier (COB-IDs) <LocalCANPRdo1…4_CobIdUsedByPdo> (73#8…11) parameters can be configured with the COB-IDs in the following table:

CAN-ID	COB-ID	Protocol	Reference
0x200	0x200 + Remote node identifier	First receive process data object (RxPDO)	CiA 301
0x300	0x300 + Remote node identifier	Second receive process data object (RxPDO)	CiA 301
0x400	0x400 + Remote node identifier	Third receive process data object (RxPDO)	CiA 301
0x500	0x500 + Remote node identifier	Fourth receive process data object (RxPDO)	CiA 301

Table 36: Receive PDOs and corresponding COB-IDs

Make sure that the receive identifiers <LocalRPdo1…4_CobIdUsedByPdo> are different to the transmit identifiers <LocalTPdo1…4_CobIdUsedByPdo>.
Otherwise you receive your own response and you will not get a timeout fault!

⇨ Chapter "6.9.6.3 Object 73#6: Remote node identifier", page 113

6.9.3.1 Object 73#8 / 73#12 / 73#16: 1st RxPDO configuration

First receive PDO channel configuration.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#8	LocalRPdo1_CobIdUsedByPdo	0x5400#1	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0200+127
73#12	LocalRPdo1_TransmissionType	0x5400#2	UINT8	rw	Y	UINT8	255
73#16	LocalRPdo1_EventTimer	0x5400#3	UINT16	rw	Y	UINT16	0

Value description

Parameter	**Description**
<LocalRPdo1_CobIdUsedByPdo>	COB-ID of the 1st receive PDO.
<LocalRPdo1_TransmissionType>	For local CAN this parameter must be set to 0xFF (asynchron).
<LocalRPdo1_EventTimer>	This parameter defines the timeout in milliseconds for the receive PDO timeout monitoring. If this parameter is set to 0 the receive timeout monitoring is turned off.

Table 37: Parameters of 1st RxPDO configuration object (73#8 / 73#12 / 73#16)

<LocalRPdo1_CobIdUsedByPdo>				
Bit	31	30	29…11	10…0
Description	0: PDO will be processed/received 1: PDO will not be processed/received	Reserved	Reserved	11 bit COB-ID

Table 38: Possible values of parameter <LocalRPdo1_CobIdUsedByPdo> (73#8)

<LocalRPdo1_TransmissionType>	
0	The last incoming PDO will be processed synchronously to the next incoming SYNC telegram.
1…240	The last incoming PDO will be processed synchronously to every <TransmissionType> SYNC telegram.
241…253	Reserved
254…255	PDO will be processed immediately after reception.

Table 39: Possible values of parameter <LocalRPdo1_TransmissionType> (73#12)

6.9.3.2 Object 73#9 / 73#13 / 73#17: 2nd RxPDO configuration

Second receive PDO channel configuration.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#9	LocalRPdo2_CobIdUsedByPdo	0x5401#1	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0300+127
73#13	LocalRPdo2_TransmissionType	0x5401#2	UINT8	rw	Y	UINT8	255
73#17	LocalRPdo2_EventTimer	0x5401#3	UINT16	rw	Y	UINT16	0

Value description

⇨ Chapter "6.9.3.1 Object 73#8 / 73#12 / 73#16: 1st RxPDO configuration", page 99

6.9.3.3 Object 73#10 / 73#14 / 73#18: 3rd RxPDO configuration

Third receive PDO channel configuration.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#10	LocalRPdo3_CobIdUsedByPdo	0x5402#1	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0400+127
73#14	LocalRPdo3_TransmissionType	0x5402#2	UINT8	rw	Y	UINT8	255
73#18	LocalRPdo3_EventTimer	0x5402#3	UINT16	rw	Y	UINT16	0

Value description

⇨ Chapter "6.9.3.1 Object 73#8 / 73#12 / 73#16: 1st RxPDO configuration", page 99

6.9.3.4 Object 73#11 / 73#15 / 73#19: 4th RxPDO configuration

Fourth receive PDO channel configuration.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
73#11	LocalRPdo4_CobIdUsedByPdo	0x5403#1	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0500+127
73#15	LocalRPdo4_TransmissionType	0x5403#2	UINT8	rw	Y	UINT8	255
73#19	LocalRPdo4_EventTimer	0x5403#3	UINT16	rw	Y	UINT16	0

Value description

⇨ Chapter "6.9.3.1 Object 73#8 / 73#12 / 73#16: 1st RxPDO configuration", page 99

6.9.3.5 Receive process data object (RxPDO) mapping

With the receive process data object (RxPDO) mapping most object dictionary entries can be mapped to a RxPDO. Each CANopen telegram can carry 8 bytes data. The smallest data types used in the digital servo valve are 8 bit integers. Therefore eight object dictionary entries with 8 bit data length or four object dictionary entries with 16 bits or two object dictionary entries with 32 bit can be mapped within one RxPDO. An arbitrary combination of different data types is possible if the sum of the mapped RxPDO data is less or equal 8 bytes.

Example:
The following values should be mapped to the fourth RxPDO (default mapping):

- Device state machine (DSM) Control Word <ControlWord> (0#37)
  ⇨ Chapter "5.2 Device state machine (DSM)", page 42
- Spool position setpoint value <SplSetpoint> (21#21…23)
  ⇨ Chapter "6.2.3 Spool position setpoint value path", page 54
- Pressure setpoint value <PrsSetpoint> (22#21…23)
  ⇨ Chapter "6.2.4 Pressure setpoint value path", page 56

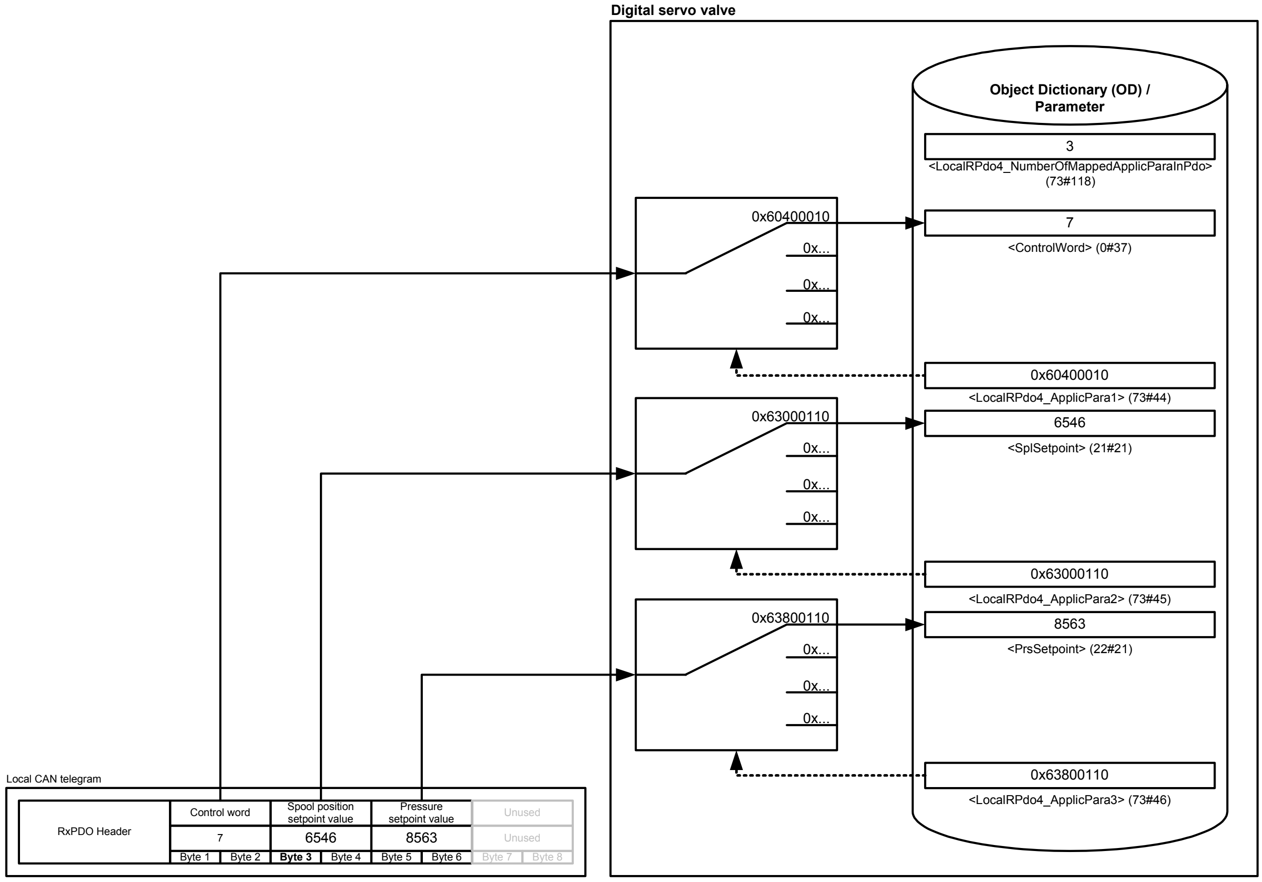


Figure 38: Receive process data object (RxPDO) mapping

The parameter <LocalRPdo4_NumberOfMappedApplicParaInPdo> (73#118) defines the number of mapped values for the fourth RxPDO. The parameter <LocalRPdo1_ApplicPara1> (173#44) references to the <ControlWord> (0#37). The references to the parameters <SplSetpoint> (21#21) and <PrsSetpoint> (22#21) are defined in the same manner.

6.9.3.6 Object 73#115 / 73#20…27: 1st RxPDO mapping

With the parameter <LocalRPdo1_NumberOfMappedApplicParaInPdo> (73#115) the number of real-time application parameters to be received can be set. To map the application parameter itself, its CANopen index, sub-index and length must be combined to a 32 bit number and written to one of the eight possible parameters <LocalRPdo1_ApplicPara1…8> (72#20…27) within the PDO object.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#115	LocalRPdo1_NumberOfMappedApplicParaInPdo	0x5600#0	UINT8	rw	Y	0…8	0
73#20	LocalRPdo1_ApplicPara1	0x5600#1	UINT32	rw	Y	UINT32	0
73#21	LocalRPdo1_ApplicPara2	0x5600#2	UINT32	rw	Y	UINT32	0
73#22	LocalRPdo1_ApplicPara3	0x5600#3	UINT32	rw	Y	UINT32	0
73#23	LocalRPdo1_ApplicPara4	0x5600#4	UINT32	rw	Y	UINT32	0
73#24	LocalRPdo1_ApplicPara5	0x5600#5	UINT32	rw	Y	UINT32	0
73#25	LocalRPdo1_ApplicPara6	0x5600#6	UINT32	rw	Y	UINT32	0
73#26	LocalRPdo1_ApplicPara7	0x5600#7	UINT32	rw	Y	UINT32	0
73#27	LocalRPdo1_ApplicPara8	0x5600#8	UINT32	rw	Y	UINT32	0

Value description

Parameter	**Description**
<LocalRPdo1_NumberOfMappedApplicParaInPdo>	Number of configured application objects
<LocalRPdo1_ApplicPara1>	Mapping of the 1^{st} application parameter
<LocalRPdo1_ApplicPara2>	Mapping of the 2^{nd} application parameter
<LocalRPdo1_ApplicPara3>	Mapping of the 3^{rd} application parameter
<LocalRPdo1_ApplicPara4>	Mapping of the 4^{th} application parameter
<LocalRPdo1_ApplicPara5>	Mapping of the 5^{th} application parameter
<LocalRPdo1_ApplicPara6>	Mapping of the 6^{th} application parameter
<LocalRPdo1_ApplicPara7>	Mapping of the 7^{th} application parameter
<LocalRPdo1_ApplicPara8>	Mapping of the 8^{th} application parameter

Table 40: Parameters of 1st RxPDO mapping object (0x560073#115 / 73#20…27)

<LocalRPdo1_ApplicPara1…8>				
Byte	**3**	**2**	**1**	**0**
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length 0x08, 0x10 or 0x20
Default	0x60	0x40	0x00	0x10

Table 41: Value description of mapping parameter <LocalRPdo1_ApplicPara1…8>

This pointer contains a combination of index, sub-index and length of the parameter to be used.

The example value is 0x60400010, which refers to the <ControlWord> (0#37), with the CANopen index 0x6040 and the CANopen sub-index 0x00 with a length of 16 bit (16=0x10).

6.9.3.7 Object 73#116 / 73#28…35: 2nd RxPDO mapping

Second receive PDO mapping.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#116	LocalRPdo2_NumberOfMappedApplicParaInPdo	0x5601#0	UINT8	rw	Y	0…8	0
73#28	LocalRPdo2_ApplicPara1	0x5601#1	UINT32	rw	Y	UINT32	0
73#29	LocalRPdo2_ApplicPara2	0x5601#2	UINT32	rw	Y	UINT32	0
73#30	LocalRPdo2_ApplicPara3	0x5601#3	UINT32	rw	Y	UINT32	0
73#31	LocalRPdo2_ApplicPara4	0x5601#4	UINT32	rw	Y	UINT32	0
73#32	LocalRPdo2_ApplicPara5	0x5601#5	UINT32	rw	Y	UINT32	0
73#33	LocalRPdo2_ApplicPara6	0x5601#6	UINT32	rw	Y	UINT32	0
73#34	LocalRPdo2_ApplicPara7	0x5601#7	UINT32	rw	Y	UINT32	0
73#35	LocalRPdo2_ApplicPara8	0x5601#8	UINT32	rw	Y	UINT32	0

Value description

⇨ Chapter "6.9.3.6 Object 73#115 / 73#20…27: 1st RxPDO mapping", page 102

6.9.3.8 Object 73#117 / 73#36…43: 3rd RxPDO mapping

Third receive PDO mapping.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#117	LocalRPdo3_NumberOfMappedApplicParaInPdo	0x5602#0	UINT8	rw	Y	0…8	0
73#36	LocalRPdo3_ApplicPara1	0x5602#1	UINT32	rw	Y	UINT32	0
73#37	LocalRPdo3_ApplicPara2	0x5602#2	UINT32	rw	Y	UINT32	0
73#38	LocalRPdo3_ApplicPara3	0x5602#3	UINT32	rw	Y	UINT32	0
73#39	LocalRPdo3_ApplicPara4	0x5602#4	UINT32	rw	Y	UINT32	0
73#40	LocalRPdo3_ApplicPara5	0x5602#5	UINT32	rw	Y	UINT32	0
73#41	LocalRPdo3_ApplicPara6	0x5602#6	UINT32	rw	Y	UINT32	0
73#42	LocalRPdo3_ApplicPara7	0x5602#7	UINT32	rw	Y	UINT32	0
73#43	LocalRPdo3_ApplicPara8	0x5602#8	UINT32	rw	Y	UINT32	0

Value description

⇨ Chapter "6.9.3.6 Object 73#115 / 73#20…27: 1st RxPDO mapping", page 102

6.9.3.9 Object 73#118 / 73#44…51: 4th RxPDO mapping

Fourth receive PDO mapping.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#118	LocalRPdo4_NumberOfMappedApplicParaInPdo	0x5603#0	UINT8	rw	Y	0…8	0
73#44	LocalRPdo4_ApplicPara1	0x5603#1	UINT32	rw	Y	UINT32	0
73#45	LocalRPdo4_ApplicPara2	0x5603#2	UINT32	rw	Y	UINT32	0
73#46	LocalRPdo4_ApplicPara3	0x5603#3	UINT32	rw	Y	UINT32	0
73#47	LocalRPdo4_ApplicPara4	0x5603#4	UINT32	rw	Y	UINT32	0
73#48	LocalRPdo4_ApplicPara5	0x5603#5	UINT32	rw	Y	UINT32	0
73#49	LocalRPdo4_ApplicPara6	0x5603#6	UINT32	rw	Y	UINT32	0
73#50	LocalRPdo4_ApplicPara7	0x5603#7	UINT32	rw	Y	UINT32	0
73#51	LocalRPdo4_ApplicPara8	0x5603#8	UINT32	rw	Y	UINT32	0

Value description

⇨ Chapter "6.9.3.6 Object 73#115 / 73#20…27: 1st RxPDO mapping", page 102

6.9.4 Transmit process data object (TxPDO) configuration

Transmit PDOs can be used for cyclic parameter transmission. Therefore the TxPDO communication must be configured and the parameters to be transmitted must be mapped to the local parameters. Four transmit TxPDO channels are available.

The parameters described in the ⇨ Chapter "6.10 Free to use parameters", page 114 can be used to transmit values to the external CAN device.

If the external CAN device uses the CAN-IDs according to the CiA 301, the CAN object identifier (COB-IDs) <LocalCANTPdo1…4_CobIdusedByPdo> (73#54…57) parameters can be configured with the COB-IDs in the following table:

CAN-ID	COB-ID	Protocol	Reference
0x180	0x180 + Remote node identifier	First transmit process data object (TxPDO)	CiA 301
0x280	0x280 + Remote node identifier	Second transmit process data object (TxPDO)	CiA 301
0x380	0x380 + Remote node identifier	Third transmit process data object (TxPDO)	CiA 301
0x480	0x480 + Remote node identifier	Fourth transmit process data object (TxPDO)	CiA 301

Table 42: Transmit PDOs and corresponding COB-IDs

Make sure that the receive identifiers <LocalRPdo1…4_CobIdUsedByPdo> are different to the transmit identifiers <LocalTPdo1…4_CobIdUsedByPdo>.
Otherwise you receive your own response and you will not get a timeout fault!

⇨ Chapter "6.9.6.3 Object 73#6: Remote node identifier", page 113

6.9.4.1 Object 73#54 / 73#58 / 73#70: 1st TxPDO configuration

First transmit PDO channel configuration.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#54	LocalTPdo1_CobIdUsedByPdo	0x5800#1	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0180+127
73#58	LocalTPdo1_TransmissionType	0x5800#2	UINT8	rw	Y	UINT8	255
73#70	LocalTPdo1_EventTimer	0x5800#5	UINT16	rw	Y	UINT16	0

Value description

Parameter	Description
<LocalTPdo1_CobIdUsedByPdo>	COB-ID of the 1st transmit PDO.
<LocalTPdo1_TransmissionType>	For local CAN this parameter must be set to 0xFF (asynchron).
<LocalTPdo1_EventTimer>	This parameter defines the send cycle time in milliseconds. If this parameter is set to 0 no PDOs are sent.

Table 43: Parameters of 1st TxPDO configuration object (73#54 / 73#58 / 73#70)

<LocalTPdo1_CobIdUsedByPdo>				
Bit	31	30	29…11	10…0
Description	0: PDO will be processed/received 1: PDO will not be processed/received	Reserved	Reserved	11 bit COB-ID

Table 44: Possible values of parameter <LocalTPdo1_CobIdUsedByPdo> (73#54 / 73#58 / 73#70)

<LocalTPdo1_TransmissionType>	
0	The process data for the transmit PDO will be updated and send immediately after the next incoming SYNC telegram.
1…240	The process data for the transmit PDO will be updated and send immediately every <TransmissionType> SYNC telegrams.
241…253	Reserved
254…255	The transmit PDO will be send after the event time is elapses (when the event time is nonzero).

Table 45: Possible values of parameter <LocalTPdo1_TrasnmissionType> (73#54 / 73#58 / 73#70)

To initiate only a single request, the <LocalTPdo1_EventTimer> (73#54 / 73#58 / 73#70) must be set to 0. The transmission then can be triggered by the parameter <LocalCANTPdoTrigger> (73#110).

6.9.4.2 Object 73#55 / 73#59 / 73#71: 2nd TxPDO configuration

Second transmit PDO channel configuration.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#55	LocalTPdo2_CobIdUsedByPdo	0x5801#1	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0280+127
73#59	LocalTPdo2_TransmissionType	0x5801#2	UINT8	rw	Y	UINT8	255
73#71	LocalTPdo2_EventTimer	0x5801#5	UINT16	rw	Y	UINT16	0

Value description

⇨ Chapter "6.9.4.1 Object 73#54 / 73#58 / 73#70: 1st TxPDO configuration", page 105

6.9.4.3 Object 73#56 / 73#60 / 73#72: 3rd TxPDO configuration

Third transmit PDO channel configuration.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#56	LocalTPdo3_CobIdUsedByPdo	0x5802#1	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0380+127
73#60	LocalTPdo3_TransmissionType	0x5802#2	UINT8	rw	Y	UINT8	255
73#72	LocalTPdo3_EventTimer	0x5802#5	UINT16	rw	Y	UINT16	0

Value description

⇨ Chapter "6.9.4.1 Object 73#54 / 73#58 / 73#70: 1st TxPDO configuration", page 105

6.9.4.4 Object 73#57 / 73#61 / 73#73: 4th TxPDO configuration

Fourth transmit PDO channel configuration.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#57	LocalTPdo4_CobIdUsedByPdo	0x5803#1	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0480+127
73#61	LocalTPdo4_TransmissionType	0x5803#2	UINT8	rw	Y	UINT8	255
73#73	LocalTPdo4_EventTimer	0x5803#5	UINT16	rw	Y	UINT16	0

Value description

⇨ Chapter "6.9.4.1 Object 73#54 / 73#58 / 73#70: 1st TxPDO configuration", page 105

6.9.4.5 Transmit process data object (TxPDO) mapping

With the transmit process data object (TxPDO) mapping the most object dictionary entries can be mapped to a TxPDO. A CANopen telegram can carry 8 bytes data in one package. The smallest data types used in the digital servo valve are 8 bit integers. Therefore eight object dictionary entries with 8 bit data length or four object dictionary entries with 16 bits or two object dictionary entries with 32 bit can be mapped within one TxPDO. An arbitrary combination of different data types is possible, if the sum of the mapped TxPDO data is less or equal 8 bytes.

Example:
The following values should be mapped to the fourth TxPDO (default mapping):

- Device state machine (DSM) Status Word <StatusWord> (0#38)
  ➪ Chapter "5.2 Device state machine (DSM)", page 42
- Spool position actual value <SplActualValue> (21#144)
  ➪ Chapter "7.3 Spool position controller", page 144
- Pressure actual value <PrsActualValue> (22#144)
  ➪ Chapter "7.5.11 Actual value filter", page 165

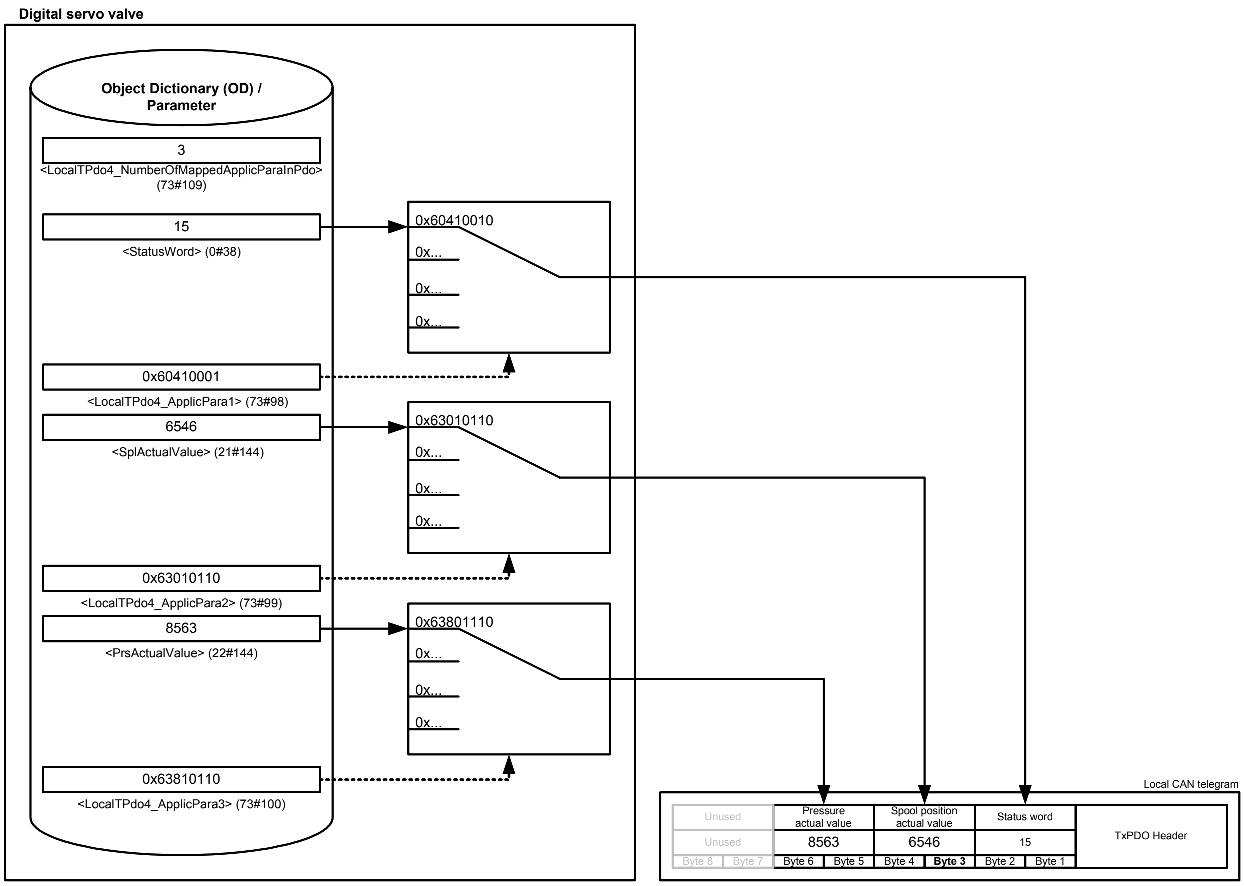


Figure 39: Transmit process data object (TxPDO) mapping

The parameter <LocalTPdo4_NumberOfMappedApplicParaInPdo> (73#109) defines the number of mapped values for the fourth TxPDO. The parameter <LocalTPdo4_ApplicPara1> (73#89) references to the <StatusWord> (0#38). The references to the parameters <SplActualValue> (21#144) and <PrsActualValue> (22#44) are defined in the same manner.

6.9.4.6 Object 73#106 / 73#74…81: 1st TxPDO mapping

First transmit PDO mapping.

With the parameter <LocalTPdo1_NumberOfMappedApplicParaInPdo> (73#106) the number of real-time application parameters to be transmitted can be set. To map the application parameter itself, its CANopen index, sub-index and length must be combined to a 32 bit number and written to one of the eight possible parameters <LocalTPdo1_ApplicPara1…8> (73#74…81) within the PDO object.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
73#106	LocalTPdo1_NumberOfMappedApplicParaInPdo	0x5A00#0	UINT8	rw	Y	0…8	0
73#74	LocalTPdo1_ApplicPara1	0x5A00#1	UINT32	rw	Y	UINT32	0
73#75	LocalTPdo1_ApplicPara2	0x5A00#2	UINT32	rw	Y	UINT32	0
73#76	LocalTPdo1_ApplicPara3	0x5A00#3	UINT32	rw	Y	UINT32	0
73#77	LocalTPdo1_ApplicPara4	0x5A00#4	UINT32	rw	Y	UINT32	0
73#78	LocalTPdo1_ApplicPara5	0x5A00#5	UINT32	rw	Y	UINT32	0
73#79	LocalTPdo1_ApplicPara6	0x5A00#6	UINT32	rw	Y	UINT32	0
73#80	LocalTPdo1_ApplicPara7	0x5A00#7	UINT32	rw	Y	UINT32	0
73#81	LocalTPdo1_ApplicPara8	0x5A00#8	UINT32	rw	Y	UINT32	0

Value description

Parameter	**Description**
<LocalTPdo1_NumberOfMappedApplicParaInPdo>	Number of configured application objects
<LocalTPdo1_ApplicPara1>	Mapping of the 1st application parameter
<LocalTPdo1_ApplicPara2>	Mapping of the 2nd application parameter
<LocalTPdo1_ApplicPara3>	Mapping of the 3rd application parameter
<LocalTPdo1_ApplicPara4>	Mapping of the 4th application parameter
<LocalTPdo1_ApplicPara5>	Mapping of the 5th application parameter
<LocalTPdo1_ApplicPara6>	Mapping of the 6th application parameter
<LocalTPdo1_ApplicPara7>	Mapping of the 7th application parameter
<LocalTPdo1_ApplicPara8>	Mapping of the 8th application parameter

Table 46: Parameters of 1st TxPDO mapping object (73#106 / 73#74...81)

<LocalTPdo1_ApplicPara1…8>				
Byte	**3**	**2**	**1**	**0**
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length 0x08, 0x10 or 0x20
Default	0x60	0x41	0x00	0x10

Table 47: Possible values of parameter <LocalTPdo1_ApplicPara1…8>

This pointer contains a combination of index, sub-index and length of the parameter to be used.

The example value is 0x63410010, which refers to the <StatusWord> (0#38), with the CANopen index 0x6341 and the CANopen sub-index 0x00 with a length of 16 bit (16=0x10).

6.9.4.7 Object 73#107 / 73#82…89: 2nd TxPDO mapping

Second transmit PDO mapping.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#107	LocalTPdo2_NumberOfMappedApplicParaInPdo	0x5A01#0	UINT8	rw	Y	0…8	0
73#82	LocalTPdo2_ApplicPara1	0x5A01#1	UINT32	rw	Y	UINT32	0
73#83	LocalTPdo2_ApplicPara2	0x5A01#2	UINT32	rw	Y	UINT32	0
73#84	LocalTPdo2_ApplicPara3	0x5A01#3	UINT32	rw	Y	UINT32	0
73#85	LocalTPdo2_ApplicPara4	0x5A01#4	UINT32	rw	Y	UINT32	0
73#86	LocalTPdo2_ApplicPara5	0x5A01#5	UINT32	rw	Y	UINT32	0
73#87	LocalTPdo2_ApplicPara6	0x5A01#6	UINT32	rw	Y	UINT32	0
73#88	LocalTPdo2_ApplicPara7	0x5A01#7	UINT32	rw	Y	UINT32	0
73#89	LocalTPdo2_ApplicPara8	0x5A01#8	UINT32	rw	Y	UINT32	0

Value description

⇨ Chapter "6.9.4.6 Object 73#106 / 73#74…81: 1st TxPDO mapping", page 108

6.9.4.8 Object 73#108 / 73#90…97: 3rd TxPDO mapping

Third transmit PDO mapping.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#108	LocalTPdo3_NumberOfMappedApplicParaInPdo	0x5A02#0	UINT8	rw	Y	0…8	0
73#90	LocalTPdo3_ApplicPara1	0x5A02#1	UINT32	rw	Y	UINT32	0
73#91	LocalTPdo3_ApplicPara2	0x5A02#2	UINT32	rw	Y	UINT32	0
73#92	LocalTPdo3_ApplicPara3	0x5A02#3	UINT32	rw	Y	UINT32	0
73#93	LocalTPdo3_ApplicPara4	0x5A02#4	UINT32	rw	Y	UINT32	0
73#94	LocalTPdo3_ApplicPara5	0x5A02#5	UINT32	rw	Y	UINT32	0
73#95	LocalTPdo3_ApplicPara6	0x5A02#6	UINT32	rw	Y	UINT32	0
73#96	LocalTPdo3_ApplicPara7	0x5A02#7	UINT32	rw	Y	UINT32	0
73#97	LocalTPdo3_ApplicPara8	0x5A02#8	UINT32	rw	Y	UINT32	0

Value description

⇨ Chapter "6.9.4.6 Object 73#106 / 73#74…81: 1st TxPDO mapping", page 108

6.9.4.9 Object 73#109 / 73#98…105: 4th TxPDO mapping

Fourth transmit PDO mapping.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#109	LocalTPdo4_NumberOfMappedApplicParaInPdo	0x5A03#0	UINT8	rw	Y	0…8	0
73#98	LocalTPdo4_ApplicPara1	0x5A03#1	UINT32	rw	Y	UINT32	0
73#99	LocalTPdo4_ApplicPara2	0x5A03#2	UINT32	rw	Y	UINT32	0
73#100	LocalTPdo4_ApplicPara3	0x5A03#3	UINT32	rw	Y	UINT32	0
73#101	LocalTPdo4_ApplicPara4	0x5A03#4	UINT32	rw	Y	UINT32	0
73#102	LocalTPdo4_ApplicPara5	0x5A03#5	UINT32	rw	Y	UINT32	0
73#103	LocalTPdo4_ApplicPara6	0x5A03#6	UINT32	rw	Y	UINT32	0
73#104	LocalTPdo4_ApplicPara7	0x5A03#7	UINT32	rw	Y	UINT32	0
73#105	LocalTPdo4_ApplicPara8	0x5A03#8	UINT32	rw	Y	UINT32	0

Value description

⇨ Chapter "6.9.4.6 Object 73#106 / 73#74…81: 1st TxPDO mapping", page 108

6.9.5 Local CAN service data object (SDO)

Service data objects are used to configure the cyclic communication parameters and the application parameters of the valve. The client, usually a CANopen master, is starting the service with an SDO(rx) request message. Here he selects the communication object (parameter) with index and subindex. In case of a write request also the new parameter value is in the message. The server (valve) will search in his object dictionary for this parameter using the index and subindex. Then the server will answer with a corresponding SDO(tx) acknowledge message which includes the actual parameter value in case of a read request.

The SDO abort codes are described in:
⇨ Chapter "8.3 Abort SDO Transfer Protocol", page 269

6.9.6 Local CAN service data object (SDO) gateway

If an external CAN device is connected to the local CAN interface the SDO parameters of the external CAN device can be read and written by the field bus master via the SDO gateway. The local CAN has a SDO client implemented. Every application parameter within the servo valve can be transmitted to, or used to receive a parameter from the external CAN device.

As CAN object identifier (COB-ID) for the SDO transmission the default SDO CAN object identifier according to the CiA 301 is used:

CAN-ID	COB-ID	Protocol	Reference
0x580	0x580 + <LocalCANRemoteNodeId> (73#6)	Transmit service data object (TxSDO)	CiA 301
0x600	0x600 + <LocalCANRemoteNodeId> (73#6)	Receive service data object (RxSDO)	CiA 301

For the SDO transmission a timeout of 500 ms is set.

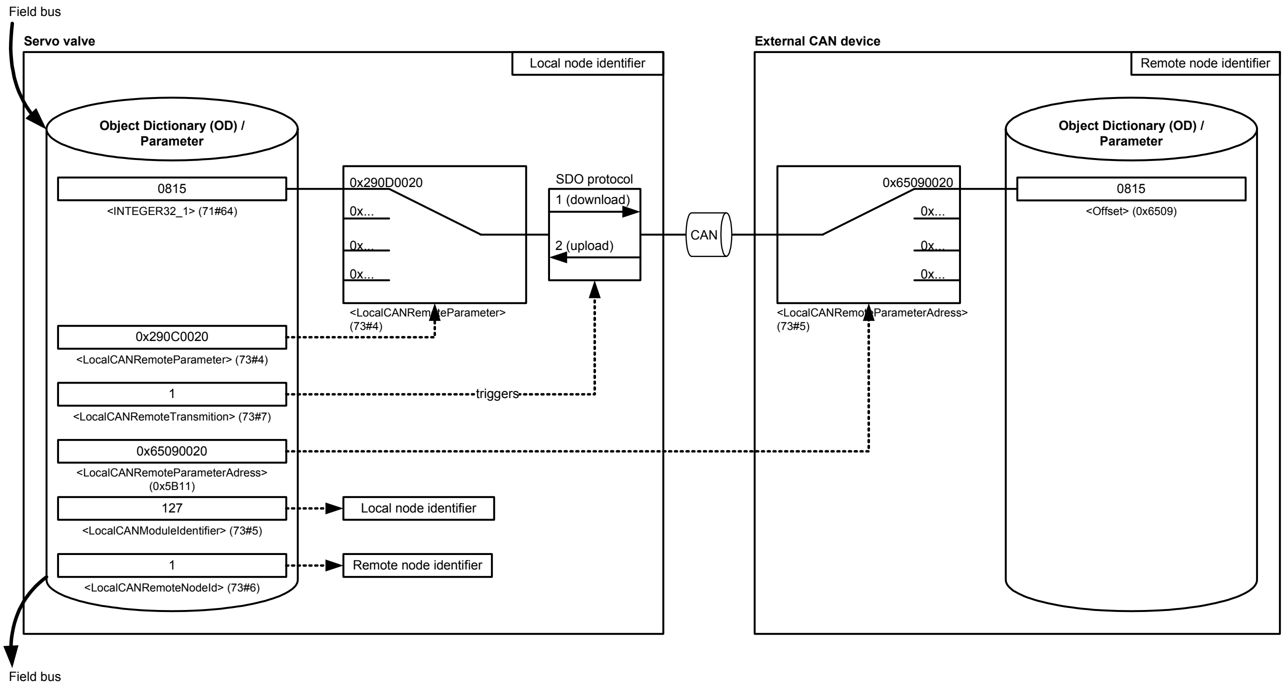


Figure 40: Local CAN service data object (SDO) gateway

6.9.6.1 Object 73#4: Remote parameter

This parameter defines the CANopen SDO index/sub-index in the servo valve to be read or written.

- In case of a SDO download it defines the local source to be transmitted.
- In case of a SDO upload it defines the local destination to be written to.

⇨ Chapter "6.10 Free to use parameters", page 114

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#4	LocalCANRemoteParameter	0x5B10#0	UINT32	rw	N	UINT32	None

Value description

<LocalCANRemoteParameter>				
Byte	**3**	**2**	**1**	**0**
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length: 0x10
Default	0x00	0x00	0x00	0x00

Table 48: Possible values of parameter <LocalCANRemoteParameter> (73#4)

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

6.9.6.2 Object 73#5: Remote parameter address

This parameter defines the CANopen SDO index/sub-index in the external CAN device to be read or written.

- In case of a CANopen SDO download it defines the destination parameter within the external CAN device to be written.
- In case of a CANopen SDO upload it defines the source parameter within the external CAN device to be transmitted.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#5	LocalCANRemoteParameterAdress	0x5B11#0	UINT32	rw	N	UINT32	None

Value description

<LocalCANRemoteParameterAdress>				
Byte	**3**	**2**	**1**	**0**
Description	CANopen indexMSB	CANopen index LSB	CANopen sub-index	Parameter bit length: 0x10
Default	0x00	0x00	0x00	0x00

Table 49: Possible values of parameter <LocalCANRemoteParameterAdress> (73#5)

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

6.9.6.3 Object 73#6: Remote node identifier

This parameter represents the node identifier of the external device.

This parameter must be different from the node identifier <LocalCANModuleIdentifier> (73#1) of the local CAN bus of the servo valve.

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#6	LocalCANRemoteNodeId	0x5B12#0	UINT8	rw	N	0…127	None

6.9.6.4 Object 73#7: Remote transmission

Writing to the <LocalCANRemoteTransmission> (73#7) parameter initiates a read or write SDO transmission. The address of the considered remote parameter will be defined with the parameter <LocalCANRemoteParameterAdress> (73#5). If the <LocalCANRemoteTransmission> is set to 1 (download) the value of the address <LocalCANRemoteParameterAdress> (73#5) will be read and saved in the local parameter defined by the <LocalCANRemoteParameter> (3#4). If <LocalCANRemoteTransmission> (73#7) is set to 2 (upload), the local value at the address <LocalCANRemoteParameter> (73#4) will be written to the remote parameter <LocalCANRemoteParameterAdress> (73#5).

LocalCAN							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persis-tence	Value range	Default
73#7	LocalCANRemoteTransmission	0x5B13#0	INT8	rw	N	–1…2	None

Value description

<LocalCANRemoteTransmission>	Description
–1	Operation was not successful.
0	Operation was successful.
1	Execute a SDO download operation from the servo valve to the external CAN device. If the download operation is successful the <LocalCANRemoteTransmission> (73#7) changes to 0. If the download operation is not successful the <LocalCANRemoteTransmission> (73#7) changes to –1.
2	Execute an SDO upload operation from the external CAN device to the servo valve. If the upload operation is successful the <LocalCANRemoteTransmission> (73#7) changes to 0. If the upload operation is not successful the <LocalCANRemoteTransmission> (73#7) changes to –1.

Table 50: Possible values of parameter <LocalCANRemoteTransmission> (73#7)

6.9.7 Local CAN Synchronization (SYNC) producer protocol emulation

The following configuration can be used to realize a synchronous PDO transmission between the servo valve and the external CAN device.

Therefore use one of the PDO transmit channels to emulate a sync message telegram.

For example the following settings set up the fourth transmit PDO channel to act as SYNC producer:

<LocalCANCobIdUsedByPdo> (73#57):	0x80 (SYNC Protocol COB-ID)
<LocalCANEventTimer> (73#61):	Sync cycle time in milliseconds
<LocalCANNumberOfMappedApplicationParametersInPdo> (73#109):	0 (no parameters mapped)

6.10 Free to use parameters

The following dummy parameters can be used as buffer parameter e.g.

- as placeholder in the PDO mapping or
- as temporary values in the event handler.

There are single parameters and arrays of parameters for most data types available.

6.10.1 Object 71#111: Signed one byte integer

DataType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#111	DummyDataS08	0x0002#0	INT8	rw	N	INT8	None

6.10.2 Object 71#112: Signed two byte integer

DataType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#112	DummyDataS16	0x0003#0	INT16	rw	N	INT16	None

6.10.3 Object 71#113: Signed four byte integer

DataType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#113	DummyDataS32	0x0004#0	INT32	rw	N	INT32	None

6.10.4 Object 71#114: Unsigned one byte integer

DataType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#114	DummyDataU08	0x0005#0	UINT8	rw	N	UINT8	None

6.10.5 Object 71#115: Unsigned two byte integer

DataType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#115	DummyDataU16	0x0006#0	UINT16	rw	N	UINT16	None

6.10.6 Object 71#116: Unsigned four byte integer

DataType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#116	DummyDataU32	0x0007#0	UINT32	rw	N	UINT32	None

6.10.7 Object 71#117: Float32

DataType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#117	DummyDataF32	0x0008#0	FLOAT32	rw	N	FLOAT32	None

6.10.8 Object 71#118: Visible string

DataType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#118	DummyDataVisibleString	0x0009#0	String	rw	N	64 byte	None

6.10.9 Object 71#48…55: Signed one byte integer array

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#48	Integer08_1	0x290B#1	INT8	rw	N	INT8	None
71#49	Integer08_2	0x290B#2	INT8	rw	N	INT8	None
71#50	Integer08_3	0x290B#3	INT8	rw	N	INT8	None
71#51	Integer08_4	0x290B#4	INT8	rw	N	INT8	None
71#52	Integer08_5	0x290B#5	INT8	rw	N	INT8	None
71#53	Integer08_6	0x290B#6	INT8	rw	N	INT8	None
71#54	Integer08_7	0x290B#7	INT8	rw	N	INT8	None
71#55	Integer08_8	0x290B#8	INT8	rw	N	INT8	None

6.10.10 Object 71#56…63: Signed two byte integer array

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#56	Integer16_1	0x290C#1	INT16	rw	N	INT16	None
71#57	Integer16_2	0x290C#2	INT16	rw	N	INT16	None
71#58	Integer16_3	0x290C#3	INT16	rw	N	INT16	None
71#59	Integer16_4	0x290C#4	INT16	rw	N	INT16	None
71#60	Integer16_5	0x290C#5	INT16	rw	N	INT16	None
71#61	Integer16_6	0x290C#6	INT16	rw	N	INT16	None
71#62	Integer16_7	0x290C#7	INT16	rw	N	INT16	None
71#63	Integer16_8	0x290C#8	INT16	rw	N	INT16	None

6.10.11 Object 71#64…71: Signed four byte integer array

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#64	Integer32_1	0x290D#1	INT32	rw	N	INT32	None
71#65	Integer32_2	0x290D#2	INT32	rw	N	INT32	None
71#66	Integer32_3	0x290D#3	INT32	rw	N	INT32	None

71#67	Integer32_4	0x290D#4	INT32	rw	N	INT32	None
71#68	Integer32_5	0x290D#5	INT32	rw	N	INT32	None
71#69	Integer32_6	0x290D#6	INT32	rw	N	INT32	None
71#70	Integer32_7	0x290D#7	INT32	rw	N	INT32	None
71#71	Integer32_8	0x290D#8	INT32	rw	N	INT32	None

6.10.12 Object 71#72…79: Unsigned one byte integer array

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#72	Unsigned08_1	0x290E#1	UINT8	rw	N	UINT8	None
71#73	Unsigned08_2	0x290E#2	UINT8	rw	N	UINT8	None
71#74	Unsigned08_3	0x290E#3	UINT8	rw	N	UINT8	None
71#75	Unsigned08_4	0x290E#4	UINT8	rw	N	UINT8	None
71#76	Unsigned08_5	0x290E#5	UINT8	rw	N	UINT8	None
71#77	Unsigned08_6	0x290E#6	UINT8	rw	N	UINT8	None
71#78	Unsigned08_7	0x290E#7	UINT8	rw	N	UINT8	None
71#79	Unsigned08_8	0x290E#8	UINT8	rw	N	UINT8	None

6.10.13 Object 71#80…87: Unsigned two byte integer array

DataType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#80	Unsigned16_1	0x290F#1	UINT16	rw	N	UINT16	None
71#81	Unsigned16_2	0x290F#2	UINT16	rw	N	UINT16	None
71#82	Unsigned16_3	0x290F#3	UINT16	rw	N	UINT16	None
71#83	Unsigned16_4	0x290F#4	UINT16	rw	N	UINT16	None
71#84	Unsigned16_5	0x290F#5	UINT16	rw	N	UINT16	None
71#85	Unsigned16_6	0x290F#6	UINT16	rw	N	UINT16	None
71#86	Unsigned16_7	0x290F#7	UINT16	rw	N	UINT16	None
71#87	Unsigned16_8	0x290F#8	UINT16	rw	N	UINT16	None

6.10.14 Object 71#88…95: Unsigned four byte integer array

DataType							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#88	Unsigned32_1	0x2910#1	UINT32	rw	N	UINT32	None
71#89	Unsigned32_2	0x2910#2	UINT32	rw	N	UINT32	None
71#90	Unsigned32_3	0x2910#3	UINT32	rw	N	UINT32	None
71#91	Unsigned32_4	0x2910#4	UINT32	rw	N	UINT32	None
71#92	Unsigned32_5	0x2910#5	UINT32	rw	N	UINT32	None
71#93	Unsigned32_6	0x2910#6	UINT32	rw	N	UINT32	None
71#94	Unsigned32_7	0x2910#7	UINT32	rw	N	UINT32	None
71#95	Unsigned32_8	0x2910#8	UINT32	rw	N	UINT32	None

7 Servo valve functions

This chapter describes how the servo valve operates depending on the <ControlMode> (0#40):

- Controller (spool position, pressure, axis position, axis velocity and flow controller)
- Monitoring functions
- Command signal conditioning (spool position, pressure and axis positon demand value generator)

The following structure shows the controllers and the signal conditioning blocks "demand value generator" and how they are embedded into the whole system.

⇨ Chapter "7.1 Control modes", page 118

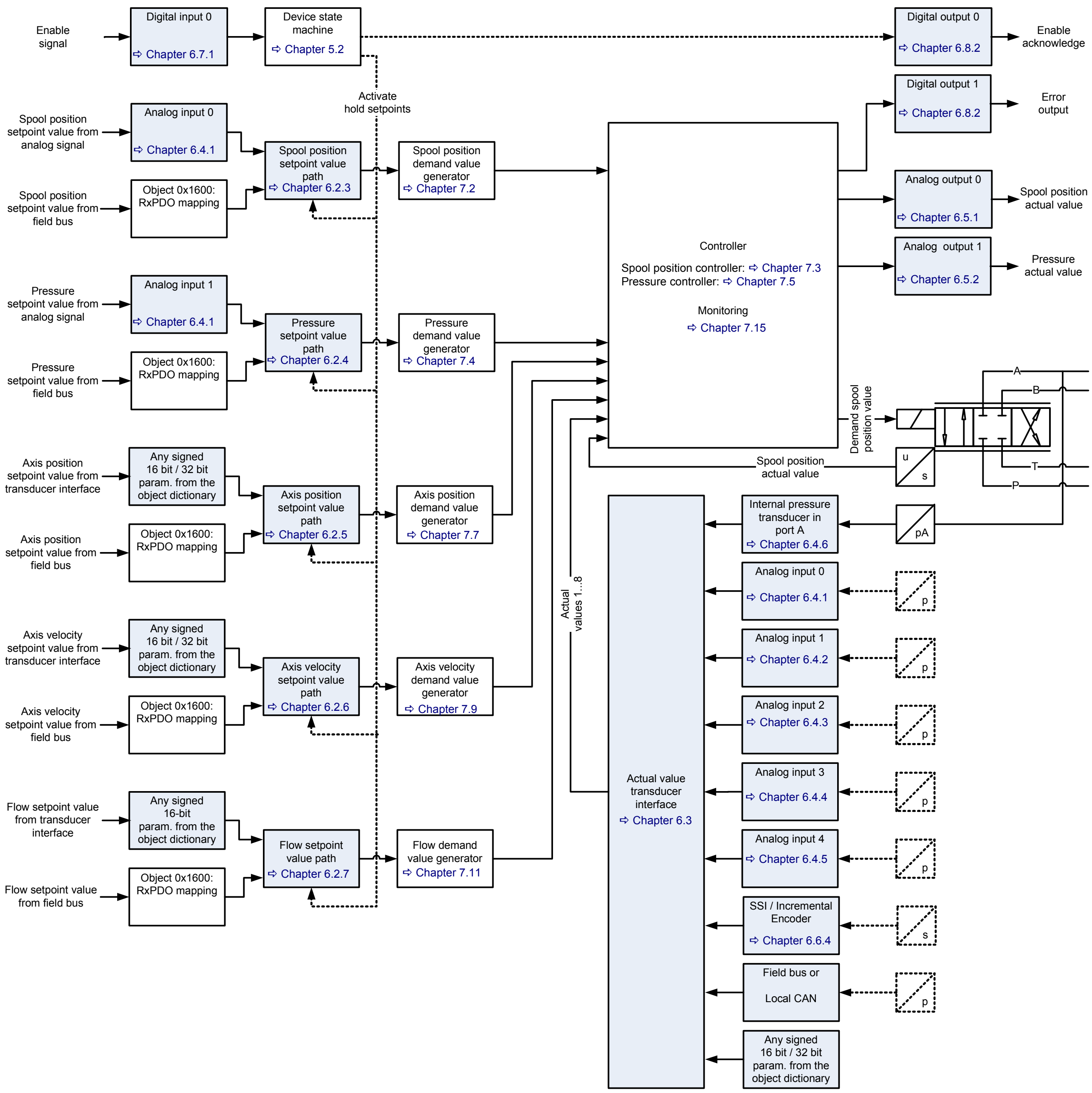


Figure 41: Servo valve controller and command signal conditioning

The servo valve can be used to control the spool position (≈ flow) and / or the pressure, axis position, axis velocity or the flow. For these tasks the following control structures are implemented:

- Spool position controller (configured by Moog)
  ➪ Chapter "7.3 Spool position controller", page 144
- Pressure controller (configured by user)
  ➪ Chapter "7.5 Pressure controller", page 158
- Pressure demand signal polarity
  ➪ Chapter "7.4.6 Pressure demand signal sign", page 157
- Spool position (Q) / pressure (P) switchover
  ➪ Chapter "7.6 Spool position (Q) / pressure (P) switchover", page 177
- Axis position controller
  ➪ Chapter "7.8 Axis position controller", page 184
- Axis velocity controller
  ➪ Chapter "7.10 Axis velocity controller", page 198
- Flow controller
  ➪ Chapter "7.12 Flow control", page 207
- Flow (Q) / pressure (P) switchover
  ➪ Chapter "7.13 Flow / pressure (P) switchover", page 218

With these control structures the user can build several controllers depending on the <ControlMode> (0#40).

7.1 Control modes

The servo valve can be run in the following control modes. The control mode of the servo valve is set with the parameter <ControlMode> (0#40).

<ControlMode>	Meaning
1	Spool position control open loop Used for tests ➪ Chapter "7.1.3 Spool position control open loop for single stage valves", page 120
2	Spool position control closed loop Spool position control ➪ Chapter "7.1.5 Spool position control closed loop for single stage and dual stage valves", page 121
3	Pressure control open loop Used for tests. Behaves like a closed loop Q control. ➪ Chapter "7.1.6 Pressure control open loop", page 121
4	Pressure control closed loop Pressure / force control ➪ Chapter "7.1.7 Pressure control closed loop", page 122
5	p/Q control In many applications the p/Q controller is used as Q controller with pressure/force limiting. ➪ Chapter "7.1.8 p/Q control closed loop", page 122
7	Axis velocity control ➪ Chapter "7.10 Axis velocity controller", page 198
8	Axis force control (equal to the pressure closed loop control) ➪ Chapter "7.1.7 Pressure control closed loop", page 122
9	Axis position control ➪ Chapter "7.8 Axis position controller", page 184
13	Flow control ➪ Chapter "7.12 Flow control", page 207
14	p/flow control In many applications the p/flow controller is used as flow controller with pressure/force limiting. ➪ Chapter "7.13 Flow / pressure (P) switchover", page 218

Table 51: Control mode values

7.1.1 Object 0#40: Control mode

This parameter selects the servo valve control mode.

The available control modes are defined by the <Capability> (0#50) parameter.
⇨ Chapter "4.1.9 Object 0#50: Capability", page 36

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#40	ControlMode	0x6043#0	INT8	rw	N	–1…14	<ControlModeDefault> (0x208)

Value description

<ControlMode>	Meaning
1	Spool position control open loop
2	Spool position control closed loop
3	Pressure control open loop
4	Pressure control closed loop
5	p/Q control
7	Axis velocity control closed loop
8	Axis force control closed loop
9	Axis position control closed loop
13	Flow control closed loop
14	p/flow control

Table 52: Possible values of parameter <ControlMode> (0#40)

7.1.2 Object 0#208: Control mode default

The <ControlModeDefault> (0#208) defines the control mode after power-up of the servo valve. During start-up of the servo valve, the parameter <ControlModeDefault> (0#208) is copied to the parameter <ControlMode> (0#40).

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
0#208	ControlModeDefault	0x4043#0	INT8	rw	Y	1…14	2

7.1.3 Spool position control open loop for single stage valves

The spool position open loop mode is selected by setting the parameter <ControlMode> (0#40) to 1 (spool position control open loop).

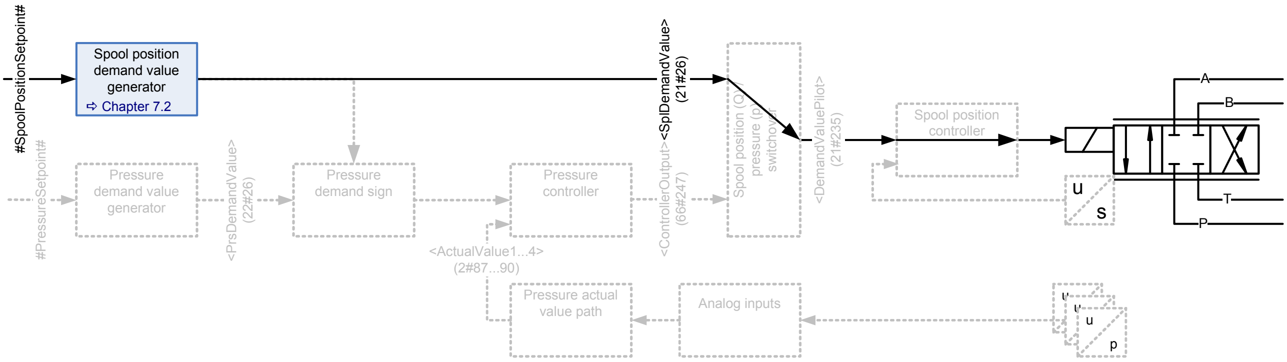


Figure 42: Spool position control open loop for single stage valves

➪ Chapter "7.1.1 Object 0#40: Control mode", page 119
➪ Chapter "7.2.1 Object 21#24…26: Demand value", page 127

This control mode is meant to be used only for commissioning or diagnostic purposes.

The signal #SpoolpositionSetpoint# is an internal signal only. It links the signal from the spool position setpoint value path to the spool position demand value generator.
➪ Chapter "6.2.3 Spool position setpoint value path", page 54

7.1.4 Spool position control open loop for dual stage valves

The spool position closed loop mode is selected by setting the parameter <ControlMode> (0#40) to 2 (spool position control closed loop). This open loop setting only affects the main stage. The pilot valve remains in spool position closed loop at all time.

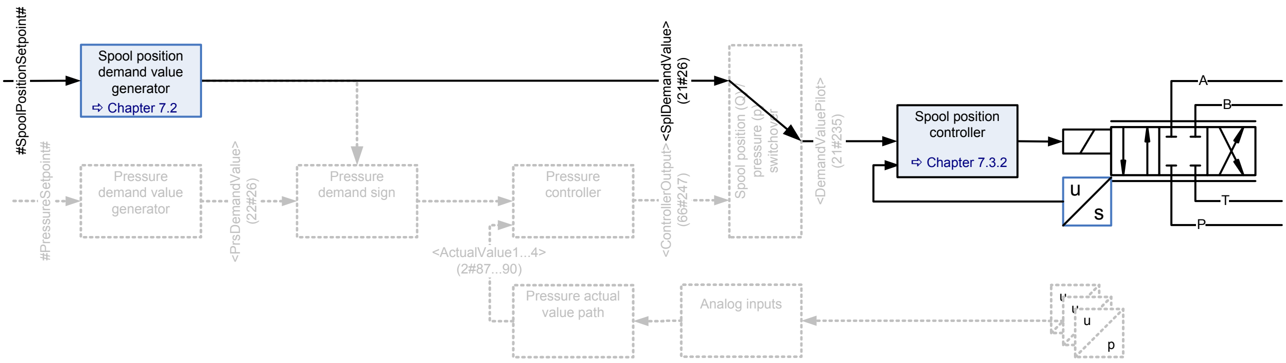


Figure 43: Spool position control open loop for dual stage valves

➪ Chapter "7.1.1 Object 0#40: Control mode", page 119
➪ Chapter "7.2.1 Object 21#24…26: Demand value", page 127

This control mode is meant to be used only for commissioning or diagnostic purposes.

The signal #SpoolpositionSetpoint# is an internal signal only. It links the signal from the spool position setpoint value path to the spool position demand value generator.
➪ Chapter "6.2.3 Spool position setpoint value path", page 54

7.1.5 Spool position control closed loop for single stage and dual stage valves

The spool position closed loop mode is selected by setting the parameter <ControlMode> (0#40) to 2 (spool position control closed loop).

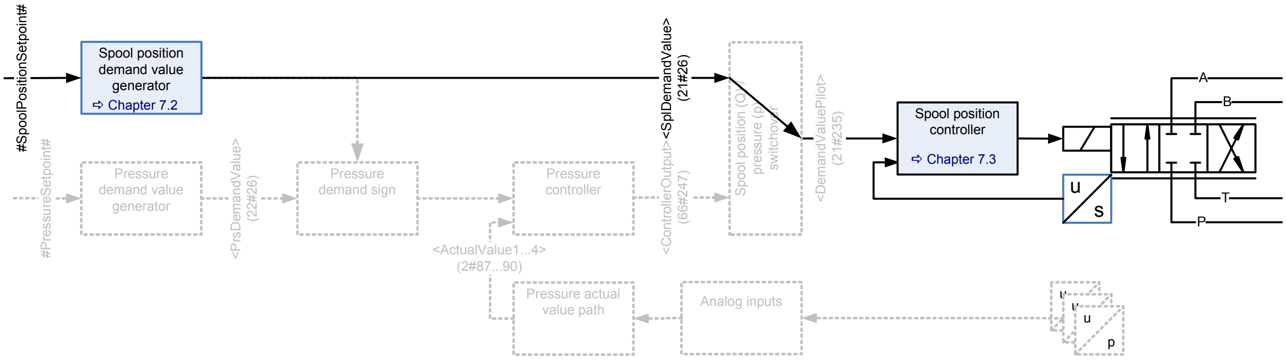


Figure 44: Spool position control closed loop for single stage and dual stage valves

⇨ Chapter "7.1.1 Object 0#40: Control mode", page 119
⇨ Chapter "7.2.1 Object 21#24…26: Demand value", page 127

The signal #SpoolpositionSetpoint# is an internal signal only. It links the signal from the spool position setpoint value path to the spool position demand value generator.
⇨ Chapter "6.2.3 Spool position setpoint value path", page 54

7.1.6 Pressure control open loop

The pressure control open loop mode is selected by setting the parameter <ControlMode> (0#40) to 3 (pressure control open loop). The behavior is the same as in the spool position closed loop control mode.

⇨ Chapter "7.1.5 Spool position control closed loop for single stage and dual stage valves", page 121

This mode is for commissioning of the servo valve only and should only be used by Moog staff..

7.1.7 Pressure control closed loop

The pressure control closed loop mode is selected by setting the parameter <ControlMode> (0#40) to 4 (pressure control closed loop).

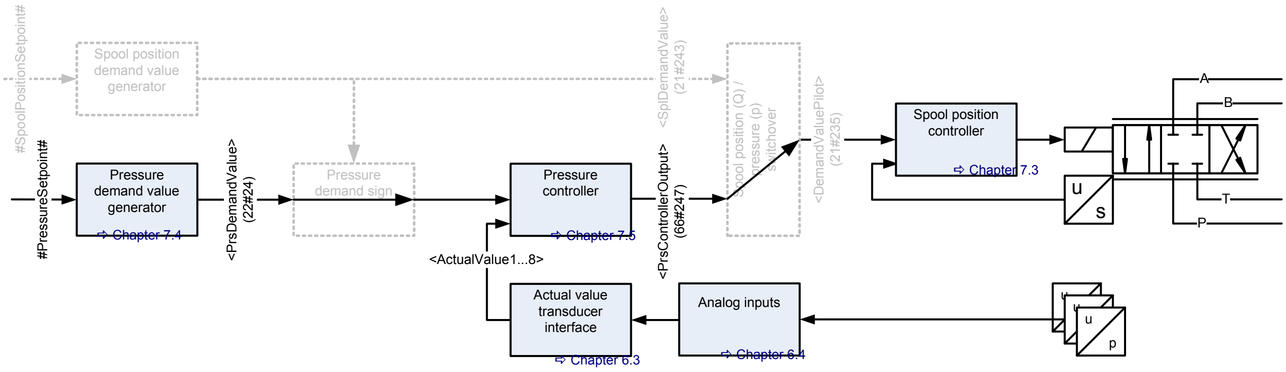


Figure 45: Pressure control closed loop

Description of the feedback signal <ActualValue1…8> (1#87…1#94):
⇨ Chapter "6.3 Drive transducer interface", page 67

⇨ Chapter "7.1.1 Object 0#40: Control mode", page 119
⇨ Chapter "7.4.1 Object 22#24…26: Demand value", page 149
⇨ Chapter "7.5.7 Object 66#247: Controller output", page 160

The signal #PressureSetpoint# is an internal signal only. It links the signal from the pressure setpoint value path to the pressure demand value generator.
⇨ Chapter "6.2.4 Pressure setpoint value path", page 56

7.1.8 p/Q control closed loop

The p/Q control mode is selected by setting the parameter <ControlMode> (0#40) to 5 (p/Q control).

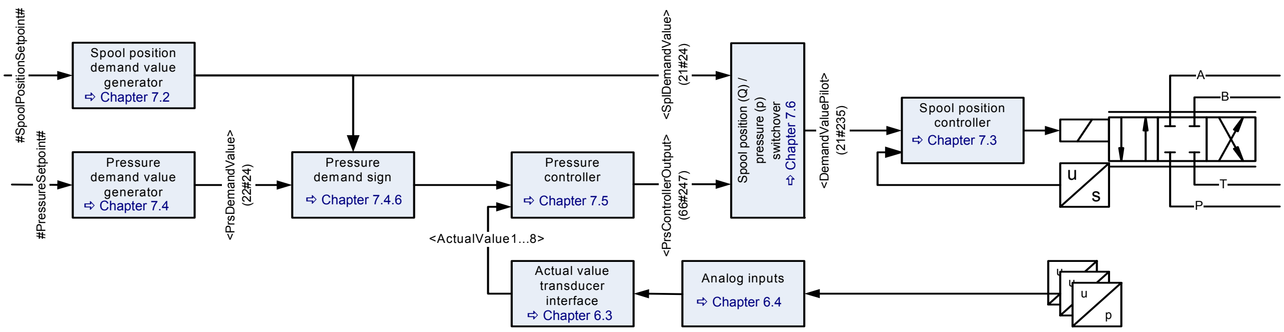


Figure 46: p/Q control closed loop

Description of the feedback signal <ActualValue1…8> (1#87…1#94):
⇨ Chapter "6.3 Drive transducer interface", page 67

⇨ Chapter "7.2.1 Object 21#24…26: Demand value", page 127
⇨ Chapter "7.4.1 Object 22#24…26: Demand value", page 149
⇨ Chapter "7.5.7 Object 66#247: Controller output", page 160
⇨ Chapter "7.6.1.1 Object 21#235: Demand value pilot", page 178

The signal #PressureSetpoint# is an internal signal only. It links the signal from the pressure setpoint value path to the pressure demand value generator.
⇨ Chapter "6.2.4 Pressure setpoint value path", page 56

The signal #SpoolPositionSetpoint# is an internal signal only. It links the signal from the spool position setpoint value path to the spool position demand value generator.
➪ Chapter "6.2.3 Spool position setpoint value path", page 54

7.1.9 Axis position control

The axis position control mode is selected by setting the parameter <ControlMode> (0#40) to 9 (axis position control closed loop).

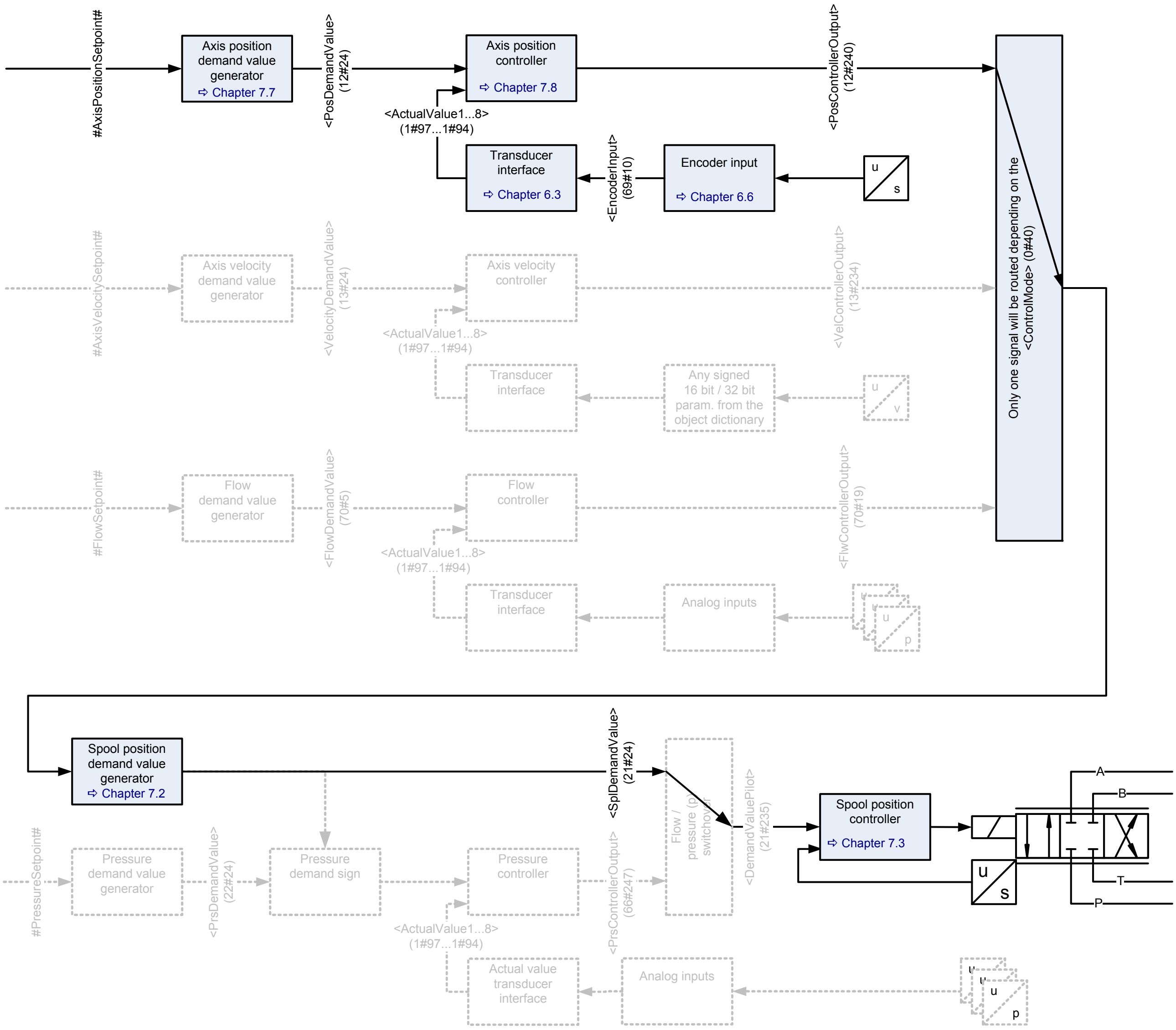


Figure 47: Axis position control

The signal #AxisPositionSetpoint# is an internal signal only. It links the signal from the axis position setpoint value path to the axis position demand value generator.
➪ Chapter "6.2.5 Axis position setpoint value path", page 58

7.1.10 Axis velocity control

The axis velocity control mode is selected by setting the parameter <ControlMode> (0#40) to 7 (axis velocity control closed loop).

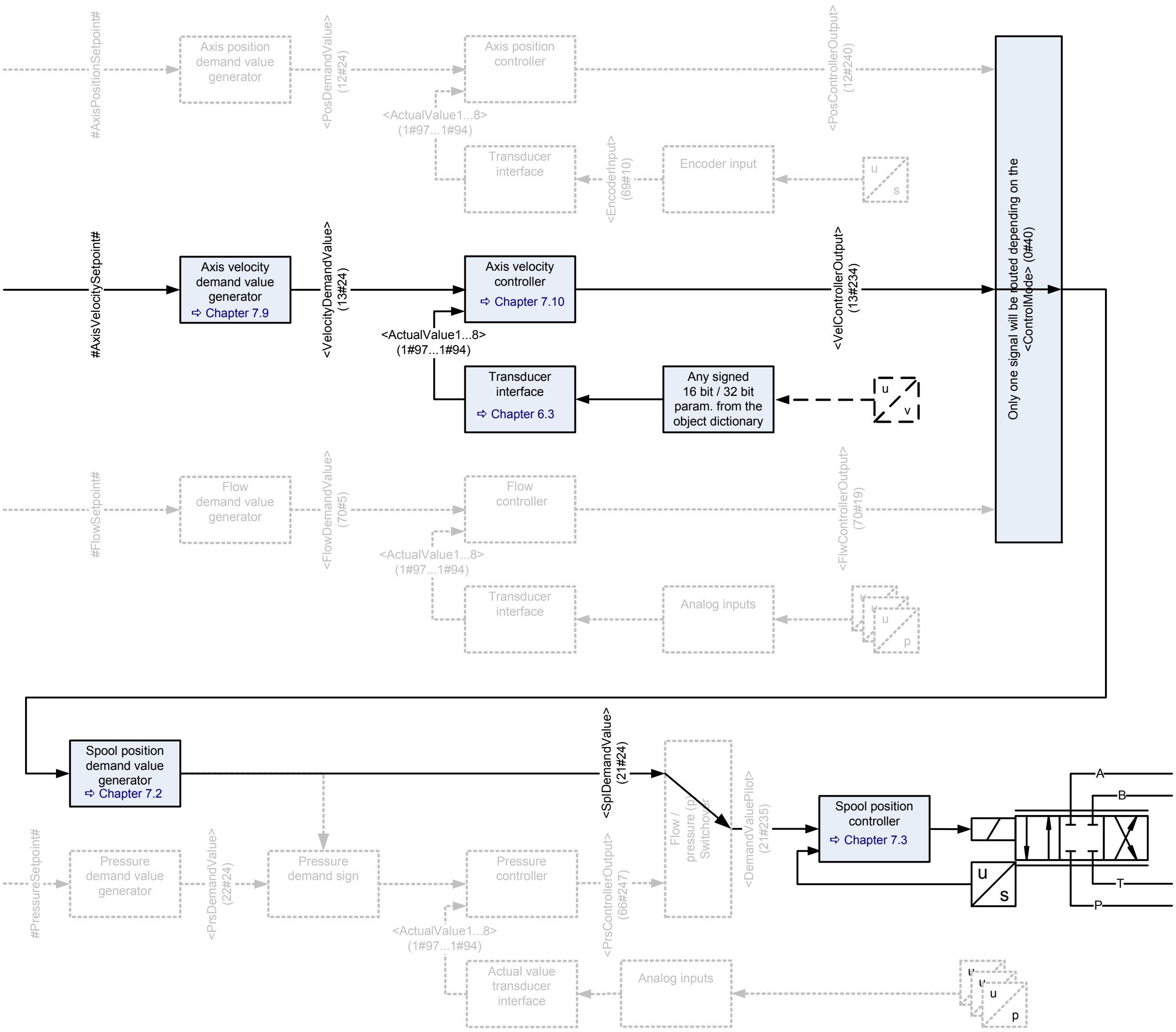


Figure 48: Axis velocity control

The signal #AxisVelocitySetpoint# is an internal signal only. It links the signal from the axis velocity setpoint value path to the axis velocity demand value generator.
⇨ Chapter "6.2.6 Axis velocity setpoint value path", page 62

7.1.11 Flow control

The flow control mode is selected by setting the parameter <ControlMode> (0#40) to 13 (flow control closed loop).

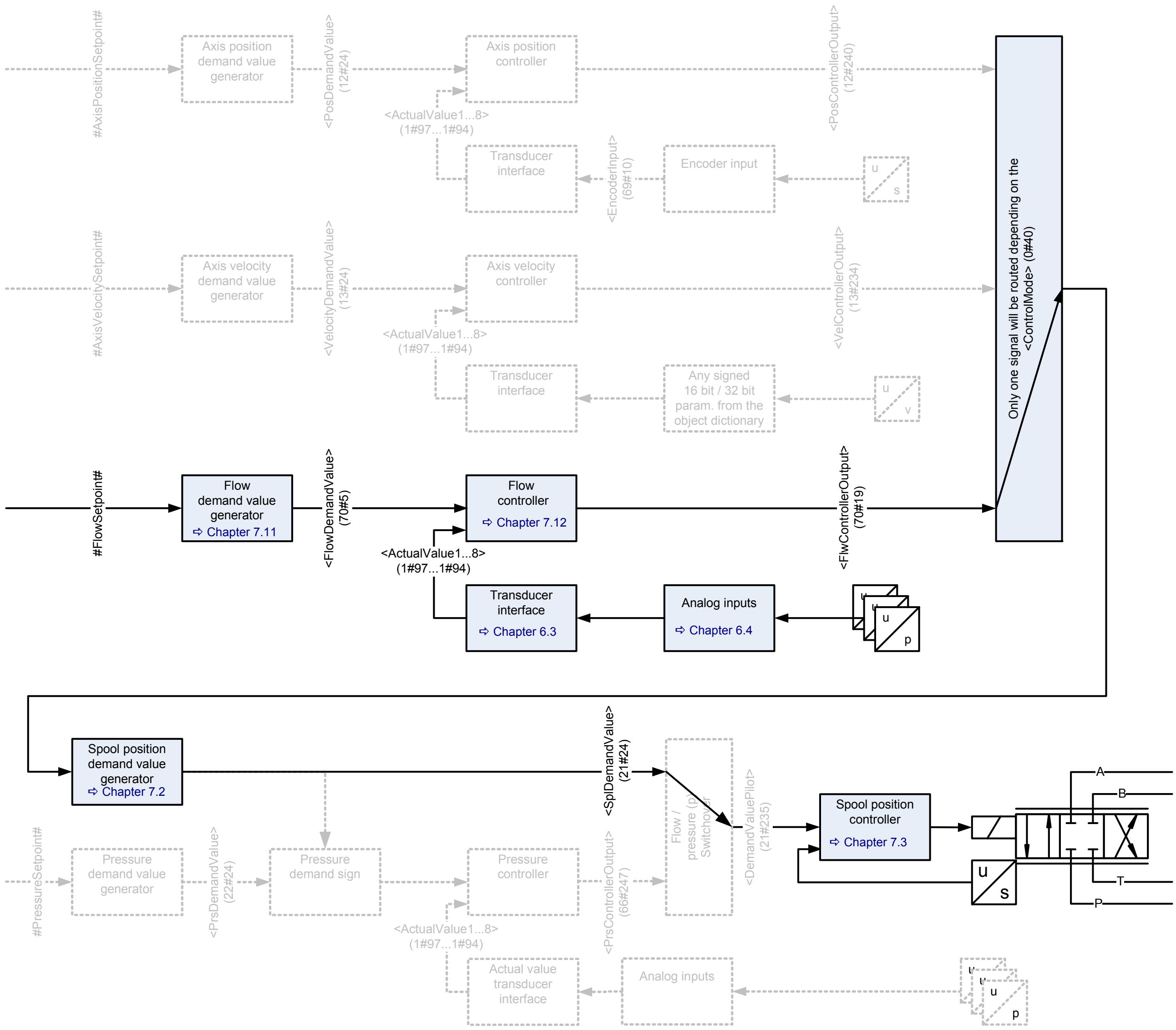


Figure 49: Flow control

The signal #FlowSetpoint# is an internal signal only. It links the signal from the flow setpoint value path to the flow demand value generator.
➩ Chapter "6.2.7 Flow setpoint value path", page 65

7.1.12 p/flow control

The p/flow control mode is selected by setting the parameter <ControlMode> (0#40) to 14 (p/flow control).

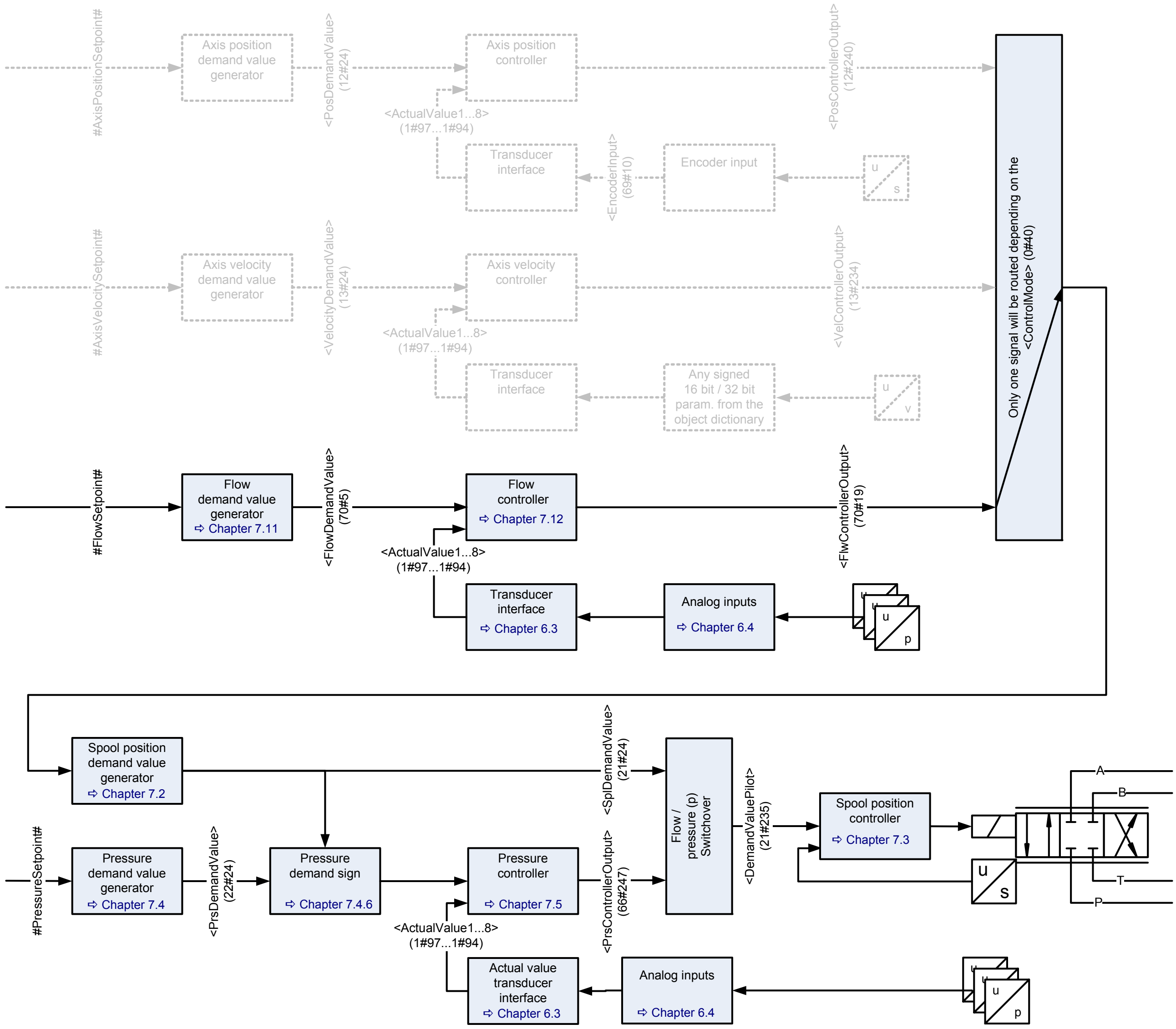


Figure 50: p/flow control

The signal #PressureSetpoint# is an internal signal only. It links the signal from the pressure setpoint value path to the pressure demand value generator.
➩ Chapter "6.2.4 Pressure setpoint value path", page 56

The signal #FlowSetpoint# is an internal signal only. It links the signal from the flow setpoint value path to the flow demand value generator.
➩ Chapter "6.2.7 Flow setpoint value path", page 65

7.1.13 Sample frequency

The <BasicSampleFrequency> (no access via Profibus) is the frequency of the real time control loop. With this frequency the actual spool position is calculated from the LVDT signal and, i.e. the current controller, pressure controller and spool position controller are calculated.

7.1.13.1 Object 0x3030: Basic sample frequency

This frequency is fixed to 10 kHz that corresponds to a sampling time of 0.1 ms.

Hardware							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3030	1	BasicSampleFrequency	UINT16	ro	-	10000	10000

7.2 Spool position setpoint conditioning / demand value generator

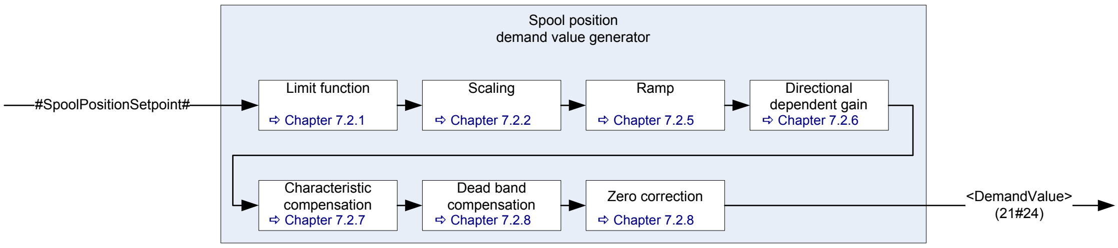


Figure 51: Spool position demand value generator

The internal signal #SpoolPositionSetpoint# is used to link the spool position setpoint value to the spool position demand value generator.
⇨ Chapter "6.2.3 Spool position setpoint value path", page 54

7.2.1 Object 21#24…26: Demand value

The demand value is generated from the #SpoolPositionSetpoint# by means of the functions in the demand value generator and forwarded to the spool position controller.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#24	SplDemandValue	0x6310#1	INT16	ro	-	INT16	None
21#25	Unit	0x6310#2	UINT8	ro	-	UINT8	0
21#26	Prefix	0x6310#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.2 Object 21#27…29: Reference value

The reference value is the value that corresponds to 100 % of the input signal. This means that a 100 % input signal is equal to 16384 increments and a –100 % input signal is equal to –16384 increments. This parameter depends on the controller hardware. It can be used by the field bus master to scale the setpoint values.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#27	SplReferenceValue	0x6311#1	INT16	ro	-	INT16	16384
21#28	Unit	0x6311#2	UINT8	ro	-	UINT8	0
21#29	Prefix	0x6311#3	INT8	ro	-	INT8	0

➩ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.3 Limit function

This function limits the value range of the spool position input signal. The limit is defined by setting the upper and lower limit.

Bit 10 of the status word indicates whether the input signal is being limited or not.
➩ Chapter "5.2.3 Object 0#38: Status word", page 48

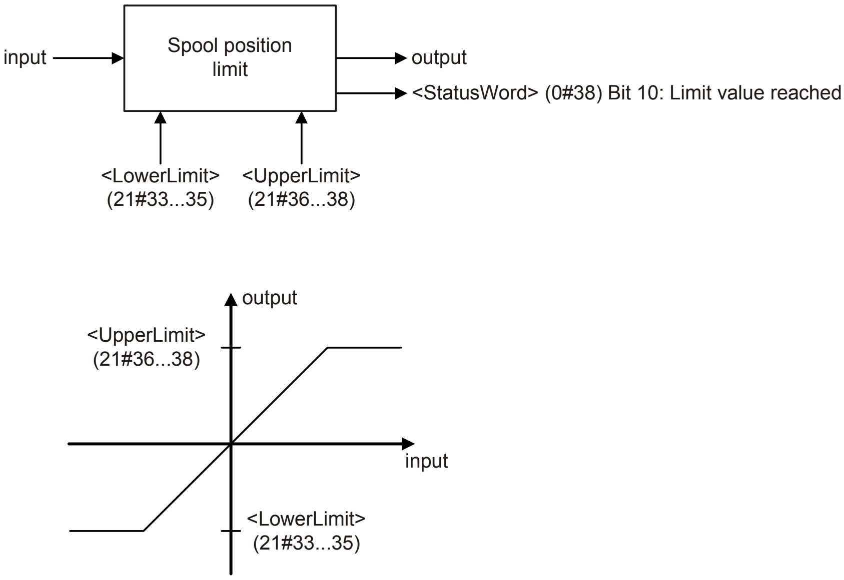


Figure 52: Limit function

The <UpperLimit> (21#33…35) must be greater than the <LowerLimit> (21#36…38).
If <LowerLimit> (21#36…38) will be set greater than the <UpperLimit> (21#33…35), the <UpperLimit> (21#33…35) will be set to the value of the <LowerLimit> (21#36…38).

7.2.3.1 Object 21#33…35: Upper Limit

ValvePositionControl_DemandValueGenerator_Limit							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#33	UpperLimit	0x6320#1	INT16	rw	Y	<LowerLimit> (21#36)…32767	16384
21#34	Unit	0x6320#2	UINT8	ro	-	UINT8	0
21#35	Prefix	0x6320#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

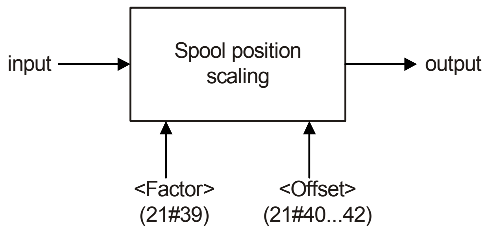
7.2.3.2 Object 21#36…38: Lower Limit

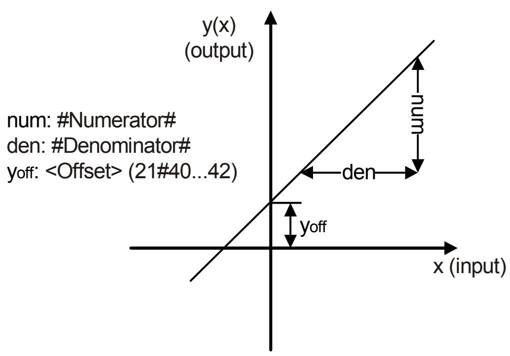
ValvePositionControl_DemandValueGenerator_Limit							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#36	LowerLimit	0x6321#1	INT16	rw	Y	–32768… <UpperLimit> (21#33)	–16384
21#37	Unit	0x6321#2	UINT8	ro	-	UINT8	0
21#38	Prefix	0x6321#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.4 Scaling

This function is used to scale the spool position setpoint value, e.g. to influence the input signal's range. The output signal is calculated by multiplication of the input signal with a scaling factor and a subsequent addition of an offset according to the following figure.





$$y(x) = m \cdot x + y_{off}$$

$$m = \frac{num}{den}$$

Figure 53: Scaling function

7.2.4.1 Object 21#39: Factor

This parameter is a slope factor by which the input is multiplied. It is defined by two signed integer values, the numerator (upper 16 bits of the parameter) and the denominator (lower 16 bits of the parameter).

ValvePositionControl_DemandValueGenerator_Scaling							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#39	Factor	0x6322#0	UINT32	rw	Y	UINT32	0x00010001

Value description

<Factor>				
Bit	31	16	15	0
Description	#Numerator#		#Denominator#	

Table 53: Data structure of the slope factor

7.2.4.2 Object 21#40…42: Offset

This parameter is the offset of the linear output function.

ValvePositionControl_DemandValueGenerator_Scaling							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#40	Offset	0x6323#1	INT16	rw	Y	INT16	0
21#41	Unit	0x6323#2	UINT8	ro	-	UINT8	0
21#42	Prefix	0x6323#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.5 Ramp

The ramp function limits the slew rate of the input signal. The <Type> (21#43) parameter is used to activate a one-quadrant, two-quadrant or four-quadrant ramp or to deactivate the ramp function.

If the ramp function is running or the ramp function is stopped by the bit 15 (ramp stop) of the #ControlWord# signal the following <StatusWord> (0#38) bits are set:

<StatusWord> (0#38) bit	**Description**
9	This bit is set if the following conditions are true: Spool position and/or pressure ramp function is active and spool position and/or pressure function is running and #ControlWord# bit 15 is set to false.
15	This bit is set if the following conditions are true: Spool position and/or pressure ramp function is active and #ControlWord# bit 15 is set to true.

⇨ Chapter "5.2.3 Object 0#38: Status word", page 48
⇨ Chapter "5.1.1 Object 0#41: Local", page 39
⇨ Chapter "5.1.2 Object 0#37: Control word", page 40
⇨ Chapter "5.1.3 Object 0#206: Local control word", page 41

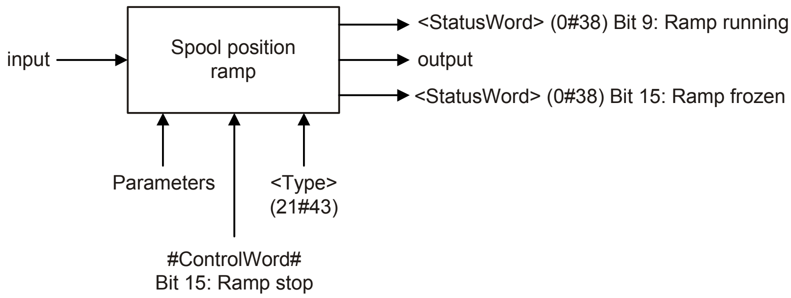


Figure 54: Ramp function

7.2.5.1 Object 21#43: Type

This parameter defines the slope rate of the ramp dependent on the direction of movement.

ValvePositionControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#43	Type	0x6330#0	INT8	rw	Y	0…3	0

Value description

<Type>	**Description**
0	No ramp
1	One-quadrant ramp
2	Two-quadrant ramp
3	Four-quadrant ramp

Table 54: Possible values of parameter <Type> (21#43)

7.2.5.2 One-quadrant ramp (ramp type 1)

This function limits the input signal's rate of change to the defined <AccelerationTime> (21#44…46).

This ramp type is active, if the parameter <Type> (21#43) is set to 1.

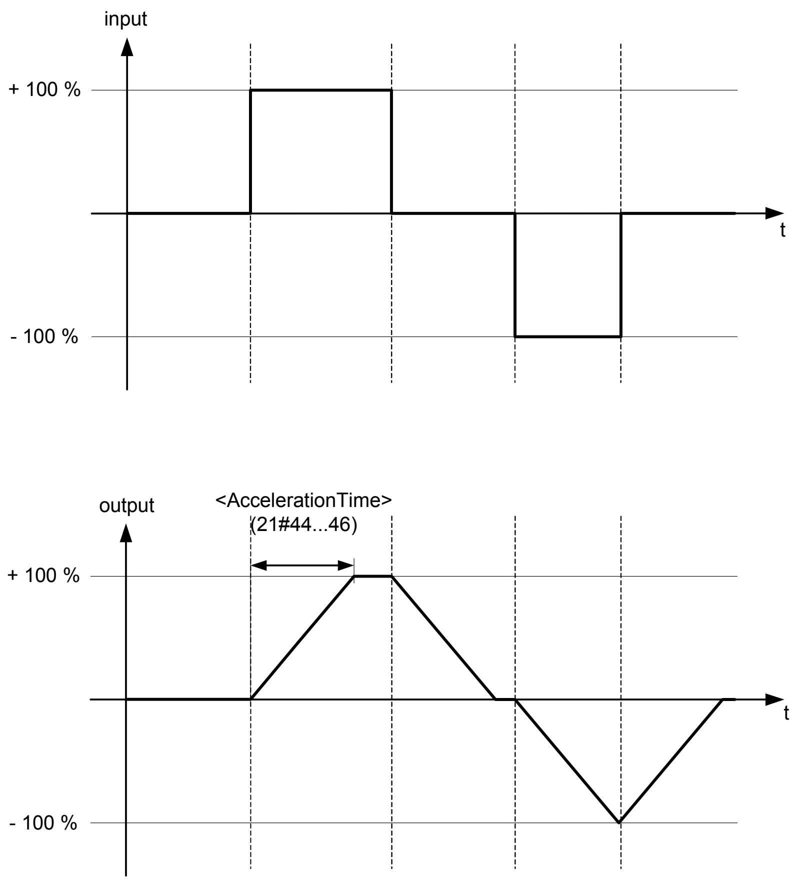


Figure 55: Ramp type 1

7.2.5.2.1 Object 21#44…46: Acceleration time

This parameter defines the output signal's maximum rate of change. The acceleration time corresponds to the time the signal needs for a change from 0 to 100 % as shown in ➪ Figure 55, page 132. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#44	AccelerationTime	0x6331#1	UINT16	rw	Y	UINT16	0
21#45	Unit	0x6331#2	UINT8	ro	-	UINT8	3
21#46	AccelerationTime_Prefix	0x6331#3	INT8	rw	Y	–4…0	–3

➪ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.5.3 Two-quadrant ramp (ramp type 2)

This function limits the input signal's rate of change to the defined <AccelerationTime> (21#44…46) and <DeccelerationTime> (21#50…52).

This ramp type is active, if the parameter <Type> (21#43) is set to 2.

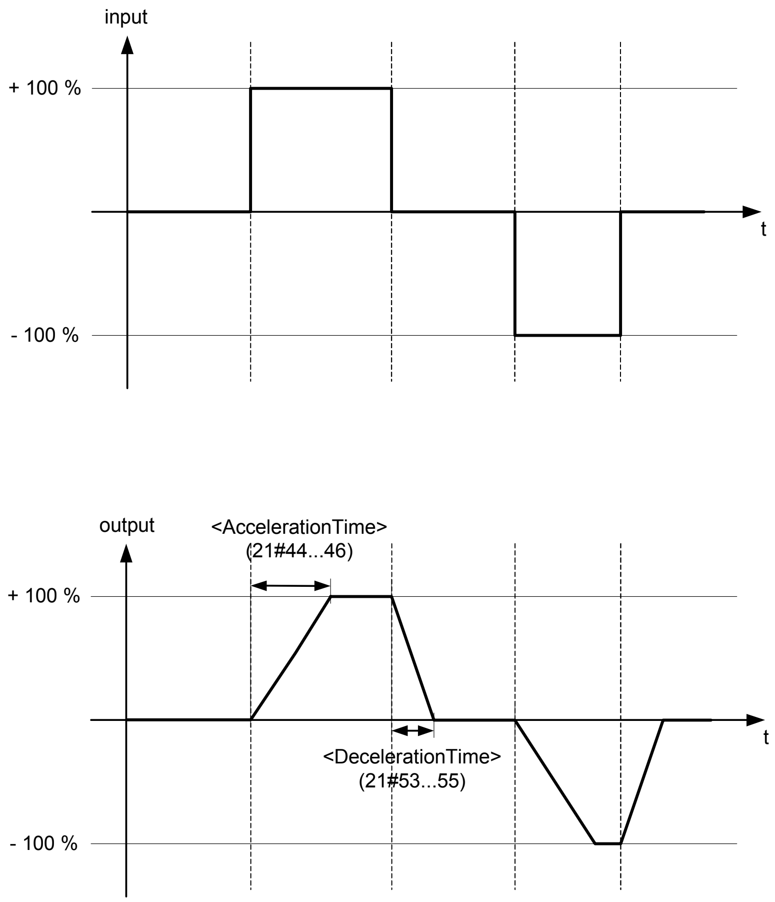


Figure 56: Ramp type 2

7.2.5.3.1 Object 21#44…46: Acceleration time

This parameter is the same as the acceleration time for ramp type 1.

➪ Chapter "7.2.5.2.1 Object 21#44…46: Acceleration time", page 132

7.2.5.3.2 Object 21#53…55: Deceleration time

This parameter defines the output signal's maximum rate of change. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#53	DecelerationTime	0x6334#1	UINT16	rw	Y	UINT16	0
21#54	Unit	0x6334#2	UINT8	ro	-	UINT8	3
21#55	DecelerationTime_Prefix	0x6334#3	INT8	rw	Y	–4…0	–3

➪ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.5.4 Four-quadrant ramp (ramp type 3)

This function limits the input signal's rate of change to an acceleration time and a deceleration time, each separated for the positive and negative sides.

This ramp type is active, if the parameter <Type> (21#43) is set to 3.

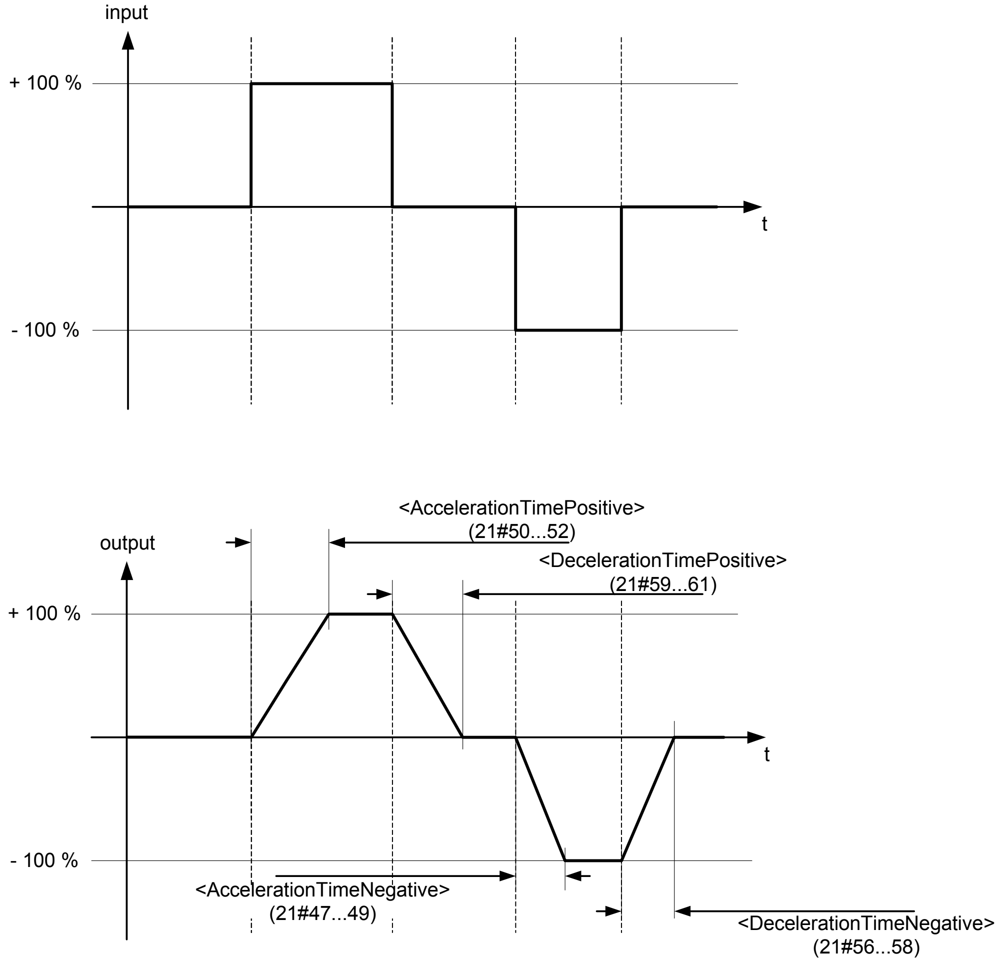


Figure 57: Ramp type 3

7.2.5.4.1 Object 21#50…52: Acceleration time positive

This parameter defines the output signal's maximum rate of change on the positive side. The acceleration time corresponds to the time that the signal needs for a change from 0 to 100 %. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#50	AccelerationTimePositive	0x6332#1	UINT16	rw	Y	UINT16	0
21#51	Unit	0x6332#2	UINT8	ro	-	UINT8	3
21#52	AccelerationTimePositive_Prefix	0x6332#3	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.5.4.2 Object 21#47…49: Acceleration time negative

This parameter defines the output signal's maximum rate of change on the negative side. The acceleration time corresponds to the time that the signal needs for a change from 0 to 100 %. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#47	AccelerationTimeNegative	0x6333#1	UINT16	rw	Y	UINT16	0
21#48	Unit	0x6333#2	UINT8	ro	-	UINT8	3
21#49	AccelerationTimeNegative_Prefix	0x6333#3	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.5.4.3 Object 21#59…61: Deceleration time positive

This parameter defines the output signal's maximum rate of change on the positive side. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#59	DecelerationTimePositive	0x6335#1	UINT16	rw	Y	UINT16	0
21#60	Unit	0x6335#2	UINT8	ro	-	UINT8	3
21#61	DecelerationTimePositive_Prefix	0x6335#3	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.5.4.4 Object 21#56…58: Deceleration time negative

This parameter defines the output signal's maximum rate of change on the negative side. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#56	DecelerationTimeNegative	0x6336#1	UINT16	rw	Y	UINT16	0
21#57	Unit	0x6336#2	UINT8	ro	-	UINT8	3
21#58	DecelerationTimeNegative_Prefix	0x6336#3	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.6 Directional dependent gain

This function reduces the input signal's gain by a <DirectionalDependentGain_Factor> (21#87) depending on the sign of the signal. The function has either an effect on the positive or the negative side of the input signal. This feature provides compensation for the different extend and retract velocities that result from cylinders with unequal area and from some load conditions. For a double rod cylinder this compensation factor is typically 1.0 but can be modified if the load favours one direction. This compensation need only be approximate but it helps ensure that the closed loop response in both extend and retract directions is symmetrical.

The actual flow depends not only on servo valve opening but on pressure loss in the servo valve and the system load.

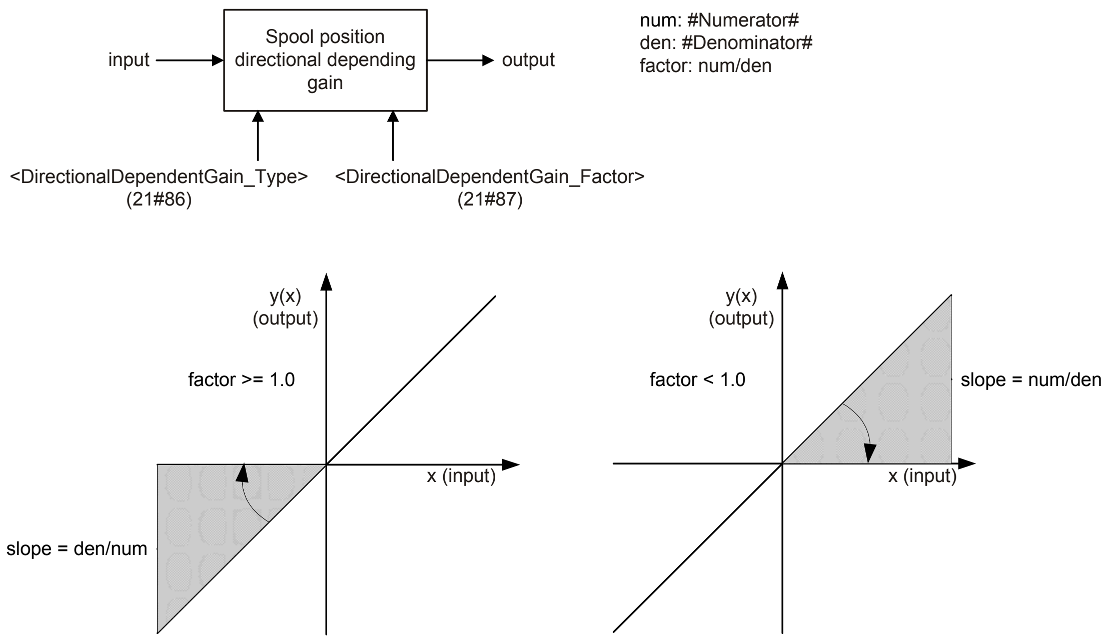


Figure 58: Directional depending gain

<Factor>	Input	Resulting output
$\lvert Factor \rvert < 1$	Input positive	$Output = Input \times \lvert Factor \rvert$
	Input negative	$Output = Input$
$\lvert Factor \rvert \geq 1$	Input positive	$Output = Input$
	Input negative	$Output = \frac{Input}{\lvert Factor \rvert}$

Table 55: Definition of the directional dependent gain factor values

7.2.6.1 Object 21#86: Type

This parameter switches the directional dependent gain function on or off.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#86	DirectionalDependentGain_Type	0x6340#0	INT8	rw	Y	0…1	0

Value description

<DirectionalDependentGain_Type>	**Description**
0	Directional dependent gain function switched off.
1	Directional dependent gain activated.

Table 56: Possible values of parameter <DirectionalDependentGain_Type> (21#86)

7.2.6.2 Object 21#87: Factor

The factor is calculated from a numerator (upper 16 bits of the parameter) and a denominator (lower 16 bits of the parameter). The value of the parameter <DirectionalDependentGain_Factor> is only effective, if the parameter <DirectionalDependentGain_Type> (21#86) is set to 1 (directional dependent gain function activated). The default value 0x00010001 corresponds to the factor 1.0.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
21#87	DirectionalDependentGain_Factor	0x6341#0	UINT32	rw	Y	UINT32	0x00010001

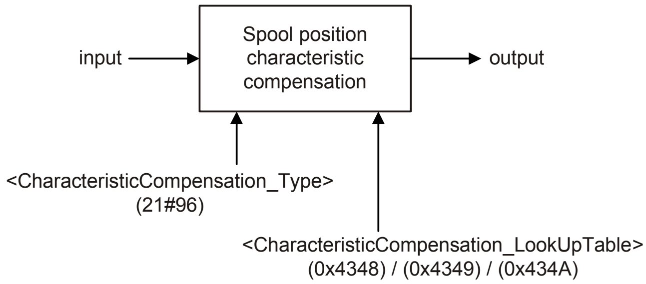
Value description

<DirectionalDependentGain_Factor>				
Bit	31	16	15	0
Description	#Numerator#		#Denominator#	

Table 57: Data structure of the directional dependent gain factor

7.2.7 Characteristic compensation

The characteristic compensation function can be used to modify the input signal in order to compensate the nonlinearity of the flow as a function of the spool position. A look-up table is used to define the spool position characteristic. With the parameter <CharacteristicCompensation_Type> (21#96), the characteristic compensation function is switched on or off. This table is predefined by Moog during production to linearize the flow. The look-up table can be modified for special applications.



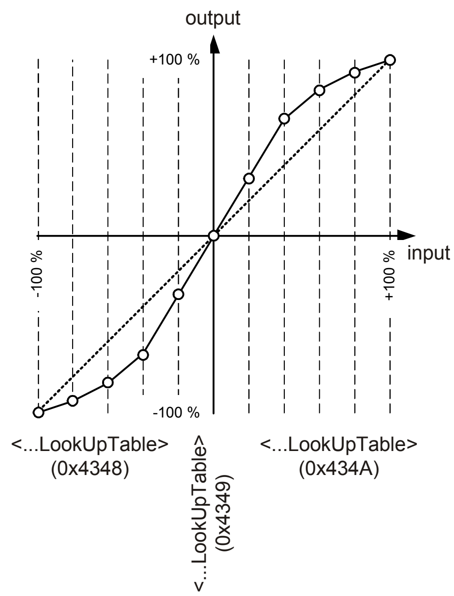


Figure 59: Characteristic compensation

7.2.7.1 Object 21#96: Type

This parameter switches the characteristic compensation function on or off.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#96	CharacteristicCompensation_Type	0x6346#0	INT8	rw	Y	–1…0	0

Value description

<CharacteristicCompensation_Type>	**Description**
0	Characteristic compensation switched off.
–1	Characteristic compensation activated.

Table 58: Possible values of parameter <CharacteristicCompensation_Type> (21#96)

7.2.7.2 Look-up table

The look-up table contains 257 sampling points and defines the characteristic curve. Intermediate values are linearly interpolated. The characteristic compensation function is activated with the parameter <CharacteristicCompensation_Type> (21#96) set to –1.

The <CharacteristicCompensation_LookUpTable> is only available via CAN SDO. There is no access via Profibus for this data type domain.

Index	Sub-index	Input values (fixed values)	E.g. linear output values (parameter values)
<CharacteristicCompensation_LookUpTable> (0x4348)	1	–16384	–16384
<CharacteristicCompensation_LookUpTable> (0x4348)	2	–16256	–16256
<CharacteristicCompensation_LookUpTable> (0x4348)	3…127	…	…
<CharacteristicCompensation_LookUpTable> (0x4348)	128	–128	–128
<CharacteristicCompensation_LookUpTable> (0x4349)	1	0	0
<CharacteristicCompensation_LookUpTable> (0x434A)	1	128	128
<CharacteristicCompensation_LookUpTable> (0x434A)	2	256	256
<CharacteristicCompensation_LookUpTable> (0x434A)	3…127	…	…
<CharacteristicCompensation_LookUpTable> (0x434A)	128	16384	16384

The distances between the input values are fixed to 128 increments per step. The possible input value range is –16384…16384 increments.

When an input value is less than –16384, the output value equals the value of sampling point <CharacteristicCompensation_LookUpTable> (0x4348), sub-index 1.
When a value is greater than 16384, the output value equals the value of sampling point <CharacteristicCompensation_LookUpTable> (0x434A), sub-index 128.

7.2.7.2.1 Object 0x4347: Look-up table

This parameter is for Moog internal use only.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
None	CharacteristicCompensation_LookUpTable	0x4347#0	DOMAIN	rw	Y	None	

7.2.7.2.2 Object 0x4348: Look-up table

This object contains the output values corresponding to the negative input values.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
None	CharacteristicCompensation_LookUpTable	0x4348#1…128	INT16	rw	Y	INT16	

7.2.7.2.3 Object 0x4349: Look-up table

This parameter contains the output values corresponding to the input value 0.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
None	CharacteristicCompensation_LookUpTable	0x4349#0	INT16	rw	Y	INT16	0

7.2.7.2.4 Object 0x434A: Look-up table

This object contains the output values corresponding to the positive input values.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
None	CharacteristicCompensation_LookUpTable	0x434A#1…128	INT16	rw	Y	INT16	0

7.2.8 Dead band compensation

Servo valves with overlap have virtually no flow when the spool is near the center or null position, i.e., inside the overlap. The overlap region is designed to have small leakage and to achieve a reliable fail safe position.

The dead band compensation function compensates for the spool overlap of the servo valve so that the flow dead band is reduced or effectively removed. With the parameter <DeadbandCompensation_Type> (21#106), the type of the dead band compensation function is selected or the function is switched off. The step height for the positive and negative direction of movement is set with the parameters <DeadbandCompensation_ASide> (21#107…109) and <DeadbandCompensation_BSide> (21#110…112).

The parameter <DeadbandCompensation_ASide> (21#107…109) specifies the step height on the positive side and the parameter <DeadbandCompensation_BSide> (21#110…112) the step height on the negative side. The border where the dead band compensation is effective is set by the parameter <DeadbandCompensation_Threshold> (21#113…115).

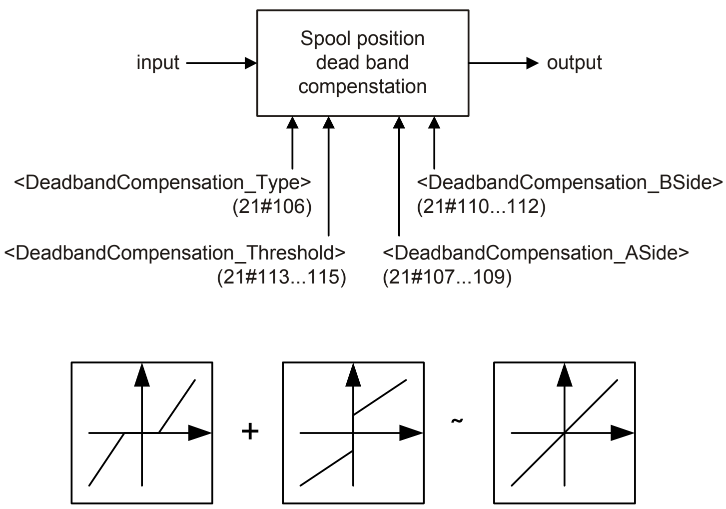


Figure 60: Dead band compensation

7.2.8.1 Object 21#106: Type

This parameter is used to select the compensation type or to switch off the dead band compensation function.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#106	DeadbandCompensation_Type	0x6342#0	INT8	rw	Y	0…2	0

Value description

<DeadbandCompensation_Type>	**Description**
0	Dead band compensation switched off.
1	Dead band jump function.
2	Dead band continuous function (recommended).

Table 59: Possible values of parameter <DeadbandCompensation_Type> (21#106)

7.2.8.2 Object 21#107…109: A side

This parameter defines the step height of the dead band on the positive side.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#107	DeadbandCompensation_ASide	0x6343#1	INT16	rw	Y	0…16384	0
21#108	Unit	0x6343#2	UINT8	ro	-	UINT8	0
21#109	Prefix	0x6343#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.8.3 Object 21#110…112: B side

This parameter defines the step height of the dead band on the negative side.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#110	DeadbandCompensation_BSide	21#112#1	INT16	rw	Y	0…16384	0
21#111	Unit	21#112#2	UINT8	ro	-	UINT8	0
21#112	Prefix	21#112#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.8.4 Object 21#113…115: Threshold

This parameter defines the positions of the steps (<DeadbandCompensation_ASide> (21#107) and <DeadbandCompensation_BSide> (21#110)). This determines the starting point of the dead band compensation step.

ValvePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#113	DeadbandCompensation_Threshold	0x6345#1	INT16	rw	Y	0…16384	0
21#114	Unit	0x6345#2	UINT8	ro	-	UINT8	0
21#115	Prefix	0x6345#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.2.8.5 Jump function (dead band compensation type 1)

If the input signals keep within the limits of the threshold value (set with parameter <DeadbandCompensation_Threshold>, 21#113…115), the output signal will be zero. At the threshold positions the output signal is increased or decreased by the step height defined by the parameter <DeadbandCompensation_ASide> (21#107…109) or <DeadbandCompensation_BSide> (21#110…112). The output signal is interpolated in the area between the threshold positions and 100 % of the input signal.

This dead band jump function is activated with the parameter <DeadbandCompensation_Type> (21#106) set to 1 (dead band jump function).

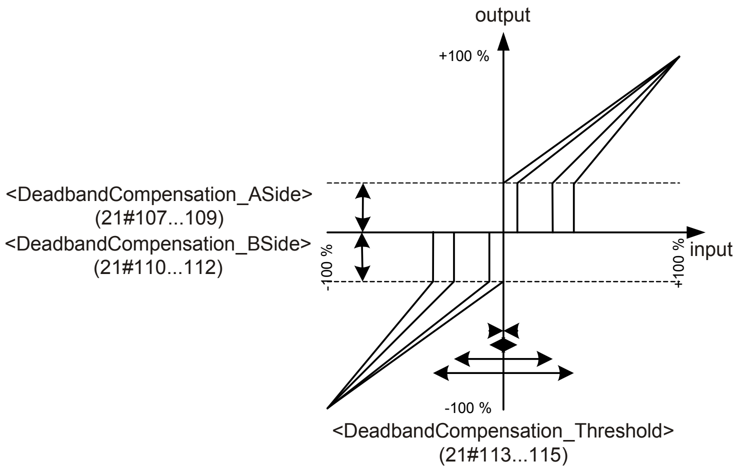


Figure 61: Dead band compensation type 1

7.2.8.6 Continuous function (dead band compensation type 2)

If the input signals keep within the range of the threshold value (set with parameter <DeadbandCompensation_Threshold>, 21#113…115), the output values will be interpolated between zero and the values defined by the parameters <DeadbandCompensation_ASide> (21#107…109) or <DeadbandCompensation_BSide> (21#110…112), respectively. At the threshold position the output signal is increased or decreased by the values set for the <DeadbandCompensation_ASide> (21#107…109) or <DeadbandCompensation_BSide> (21#110…112). The output signal is interpolated in the area between the threshold limit and 100 % of the input signal.

ⓘ This function serves to achieve a continuous transition between the areas that are within and outside of the threshold value.

This dead band continuous function is activated with the parameter <DeadbandCompensation_Type> (21#106) set to 2 (dead band continuous function).

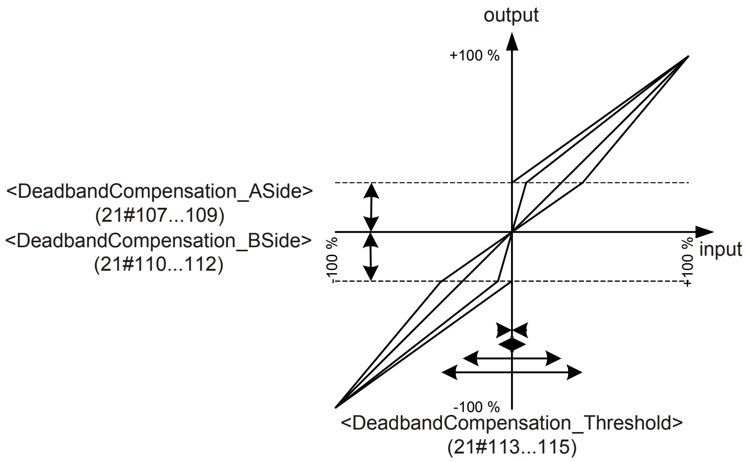
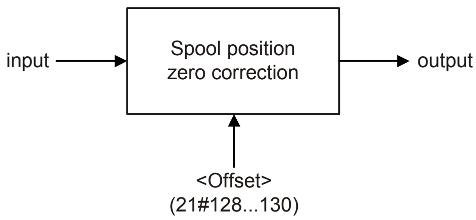


Figure 62: Dead band compensation type 2

7.2.9 Zero correction

The zero correction enables shifting of the input signal up and down by any desired offset. The <Offset> (21#128…130) is added to the input signal according to the following figure.



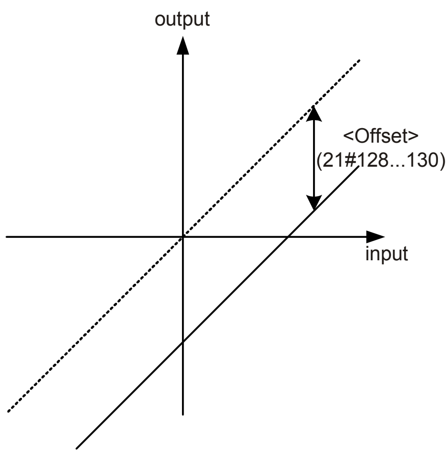


Figure 63: Zero correction

7.2.9.1 Object 21#128…130: Offset

ValvePositionControl_DemandValueGenerator_ZeroCorrection							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#128	Offset	0x6324#1	INT16	rw	Y	INT16	0
21#129	Unit	0x6324#2	UINT8	ro	-	UINT8	0
21#130	Prefix	0x6324#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.3 Spool position controller

The spool position controller controls the spool position. The parameters are set up by Moog during production. Only for the sake of completeness will some details be explained in this section. The user should understand the difference between a single stage and a dual stage servo valve.

Single stage means one spool position control loop. Dual stage means two (nested) spool position control loops.

The spool position controller is configured in the factory and cannot be changed by the user.

7.3.1 Single stage servo valve

Control structure of a single stage servo valve. For a single stage servo valve, there is only one spool position controller which is the pilot stage controller.

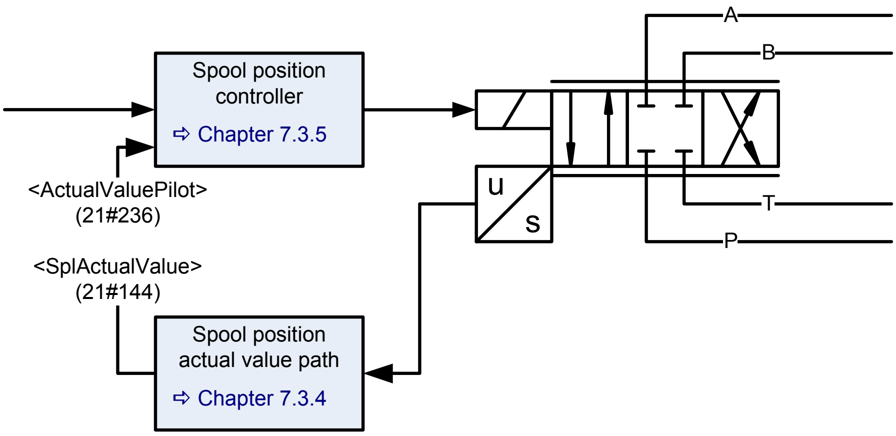


Figure 64: Single stage servo valve

7.3.2 Dual stage servo valve for open loop control

Control structure of a dual stage servo valve open loop control.

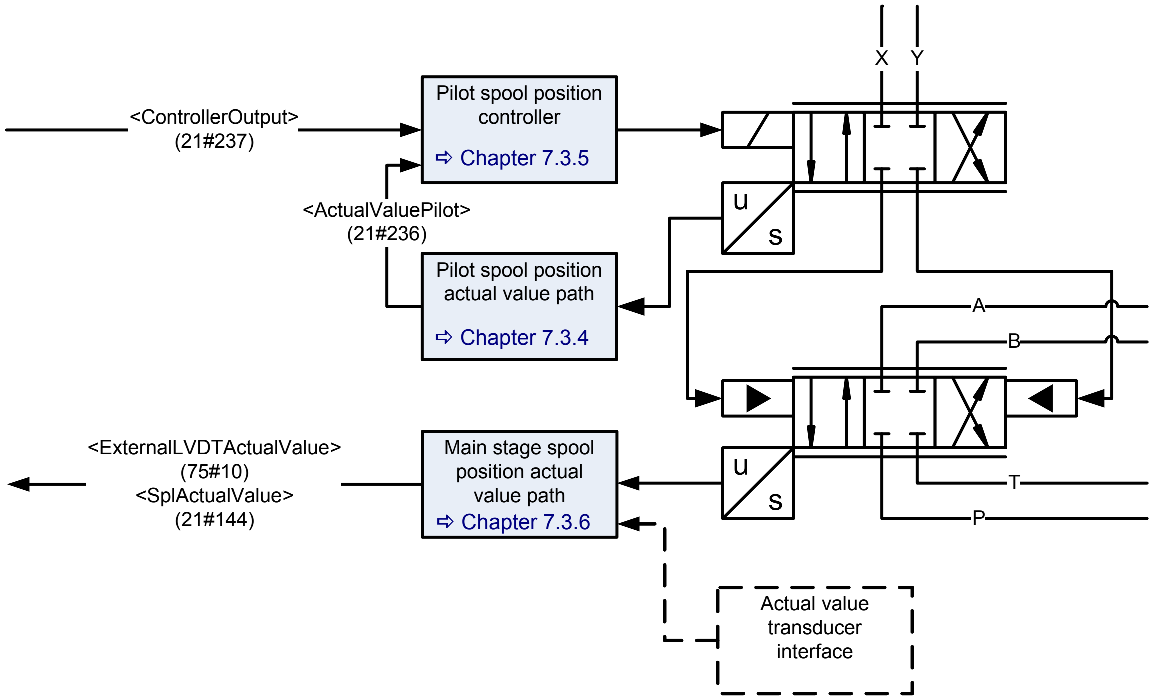


Figure 65: Dual stage servo valve for open loop control

7.3.3 Dual stage servo valve for closed loop control

Control structure of a dual stage servo valve closed loop control.

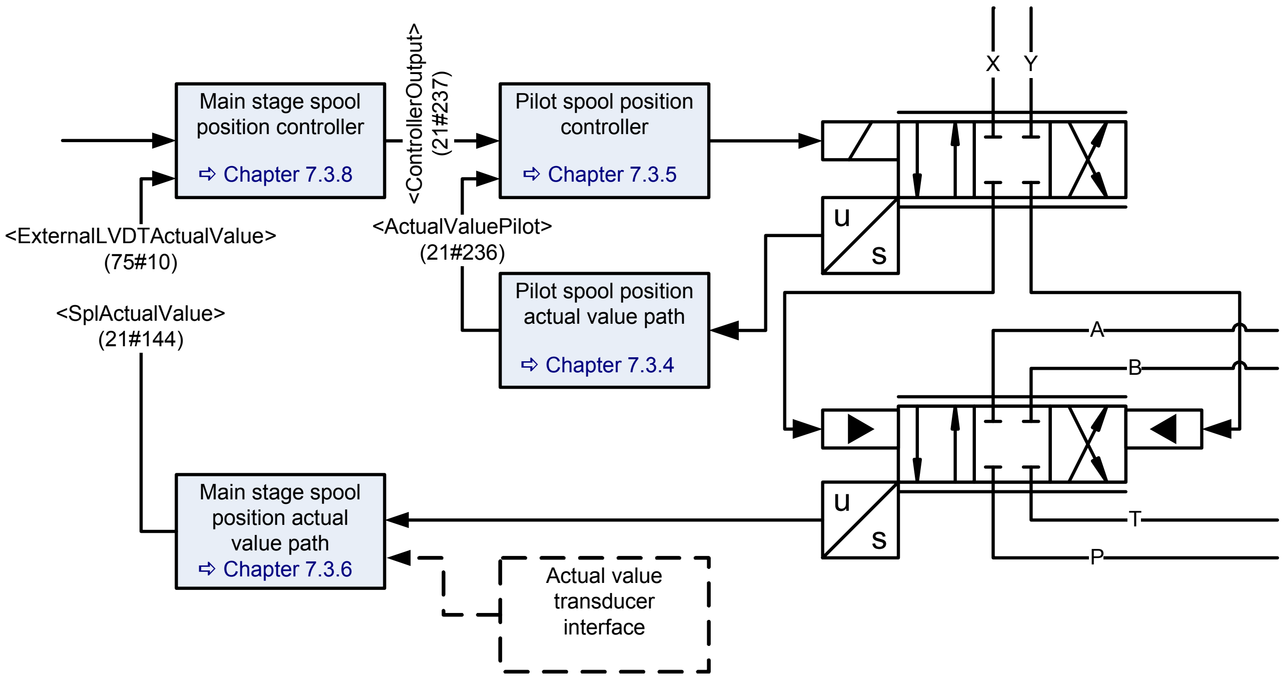


Figure 66: Dual stage servo valve for closed loop control

7.3.4 Spool position / pilot spool position actual value path

The following parameters contain the spool position depending on the servo valve hardware (single or dual stage).

7.3.4.1 Object 21#144…146: Actual value

This parameter holds the actual value of the spool position.

For a single stage servo valve, the spool position value is scaled and mapped to the signal <SplActualValue> (21#144…146).

For a dual stage servo valve, the spool position value of the main stage is scaled and mapped to the same signal <SplActualValue> (21#144…146).

ValvePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#144	SplActualValue	0x6301#1	INT16	ro	-	INT16	None
21#145	Unit	0x6301#2	UINT8	ro	-	UINT8	0
21#146	Prefix	0x6301#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.3.4.2 Object 21#236: Actual value pilot

For a single stage servo valve, the spool position value is scaled and mapped to the signal <ActualValuePilot> (21#236).

For a dual stage servo valve, the spool position value of the pilot stage is scaled and mapped to the same signal <ActualValuePilot> (21#236).

ValvePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#236	ActualValuePilot	0x3301#0	INT16	ro	-	INT16	None

7.3.4.3 Object 74#19: Customer Scaling Offset

This parameter allows an offset of ± 5 % to the pilot valve spool position <ActualValuePilot> (21#236). This offset may help for example in case of a temperature drift or another drift.

Lvdt							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
74#19	CustomerScalingOffset	0x3506#0	INT16	rw	Y	–819…819	0

7.3.5 Spool position / pilot spool position controller

For a single stage servo valve the spool position controller controls the spool position. The parameter <SplControlDeviation> (21#147…149) holds the spool position control deviation.

For a dual stage servo valve, this spool controller is used to control the pilots spool position. The main stage is controlled by the main stage spool position controller. The parameter <SplControlDeviation> (21#147…149) holds the main stage spool position control deviation.

7.3.5.1 Object 21#147…149: Control deviation

The control deviation is the difference between the setpoint value and the actual value. In case of a dual stage servo valve this parameter holds the control deviation of the main stage spool position.

ValvePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#147	SplControlDeviation	0x6350#1	INT16	ro	-	INT16	None
21#148	Unit	0x6350#2	UINT8	ro	-	UINT8	0
21#149	Prefix	0x6350#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.3.5.2 Object 21#160: Customer Overall Gain

This gain is accessible for the customer. It will be multiplied to the output of the pilot spool position controller. (Gain over all of this controller.)

ValvePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#160	CustomerOverallGain	0x241F#0	FLOAT32	rw	Y	0.0…2.0	1.0

7.3.6 Main stage spool position actual value path

If a dual stage servo valve is used, the main stage spool position value is scaled and mapped to the signal <ExternalLVDTActualValue> (75#10) and the signal <SplActualValue> (21#144…146).

⇨ Chapter "7.3.4.1 Object 21#144…146: Actual value", page 146

7.3.6.1 Object 75#11…13: Customer scaling external LVDT

This parameter is used to scale the input from the external LVDT to the actual main stage spool position.

$$\text{ExternalLVDTActualValue (75\#10)} = \frac{(\text{Input} + \text{ExternalLvdtOffset (75\#13)}) \times \text{ExternalLvdtScaNumerator (75\#11)}}{\text{ExternalLvdtScaDenominator (75\#12)}}$$

ExternalLVDT							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#11	ExternalLvdtScaNumerator	0x3237#1	INT16	rw	Y	INT16	16386
75#12	ExternalLvdtScaDenominator	0x3237#2	INT16	rw	Y	INT16	16386
75#13	ExternalLvdtOffset	0x3237#3	INT16	rw	Y	INT16	0

7.3.6.2 Object 75#10: External LVDT Actual Value

This parameter holds the main stage spool position actual value.

ExternalLVDT							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
75#10	ExternalLVDTActualValue	0x3235#0	INT16	ro	-	INT16	0

7.3.7 Main stage transducer selection

For applications with an external main stage position signal you can use a transducer interface to root any actual main stage spool position signal to the main stage spool position controller.

7.3.7.1 Object 21#238: Active transducer interface main stage

This parameter holds the interface number for the main stage spool position actual value signal rooted to the controller. If <ActiveTransducerInterfaceMainStage> (21#238) = 0 then <SplActualValue> (21#146) is used, otherwise the output of the interface number <ActiveTransducerInterfaceMainStage> (21#238).

ValveMainStageControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#238	ActiveTransducerInterfaceMainStage	0x2149#0	UINT8	rw	Y	0…8	None

7.3.8 Main stage spool position controller

The main stage spool position controller will be only used for dual stage servo valves. The main stage spool position control deviation and the main stage spool position controller output can be accessed by corresponding output parameters.

⇨ Chapter "7.3.5.1 Object 21#147…149: Control deviation", page 147

7.3.8.1 Object 21#240: Main stage customer overall gain

The internal controller output will be multiplied with this gain to get the <ControllerOutput> (21#237).

ValveMainStageControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#240	MainStageCustomerOverallGain	0x215C#0	FLOAT32	rw	Y	0.0…2.0	1.0

7.3.8.2 Object 21#237: Controller output

This parameter holds the main spool position controller output.

ValveMainStageControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#237	ControllerOutput	0x2158#0	INT16	ro	-	INT16	None

7.4 Pressure setpoint conditioning / demand value generator

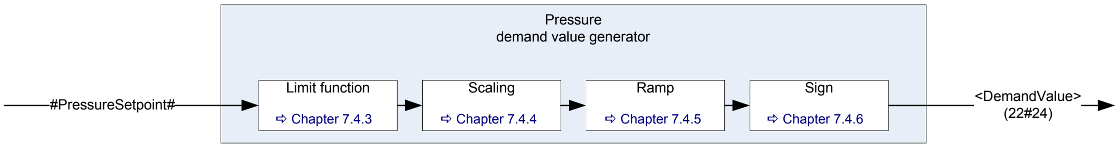


Figure 67: Pressure demand value generator

The internal signal #PressureSetpoint# is used to link the pressure setpoint value to the pressure demand value generator.
⇨ Chapter "6.2.4 Pressure setpoint value path", page 56

7.4.1 Object 22#24…26: Demand value

The demand value indicated by this parameter is generated from the #PressureSetpoint# by means of the functions in the demand value generator and forwarded to the pressure controller.

ValvePressureControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#24	PrsDemandValue	0x6390#1	INT16	ro	N	INT16	None
22#25	Unit	0x6390#2	UINT8	ro	-	UINT8	0
22#26	Prefix	0x6390#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.4.2 Object 22#27…29: Reference value

The reference value is the value that corresponds to 100 % of the input signal. This means that a 100 % pressure input signal is equal to 16384 increments and a –100 % input signal is equal to –16384 increments. This parameter depends on the controller hardware. It can be used by the field bus master to scale the setpoint values.

ValvePressureControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#27	PrsReferenceValue	0x6391#1	INT16	ro	-	INT16	16384
22#28	Unit	0x6391#2	UINT8	ro	-	UINT8	0
22#29	Prefix	0x6391#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.4.3 Limit function

This function limits the value range of the input signal. The limit is defined by setting the upper limit and lower limit parameters.

Bit 10 of the status word indicates whether the input signal is being limited by this function or not.
⇨ Chapter "5.2.3 Object 0#38: Status word", page 48

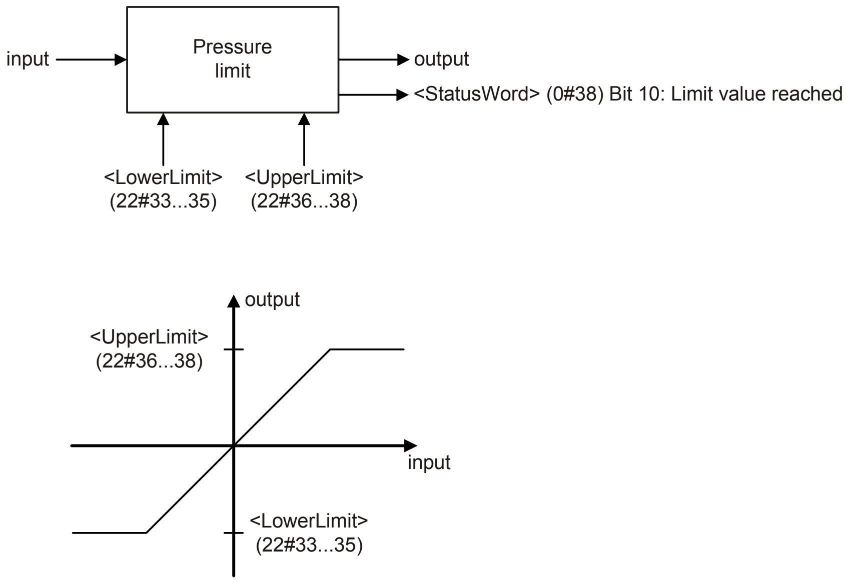


Figure 68: Limit function

The <UpperLimit> (22#33…35) must be greater than the <LowerLimit> (22#36…38).
If the <LowerLimit> (22#36…38) is greater than the <UpperLimit> (0x63A0), the <UpperLimit> (22#33…35) will be set to the value of the <LowerLimit> (22#36…38).

7.4.3.1 Object 22#33…35: Upper Limit

ValvePositionControl_DemandValueGenerator_Limit							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#33	UpperLimit	0x63A0#1	INT16	rw	Y	<LowerLimit> (0x63A1)…32767	32760
22#34	Unit	0x63A0#2	UINT8	ro	-	UINT8	0
22#35	Prefix	0x63A0#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

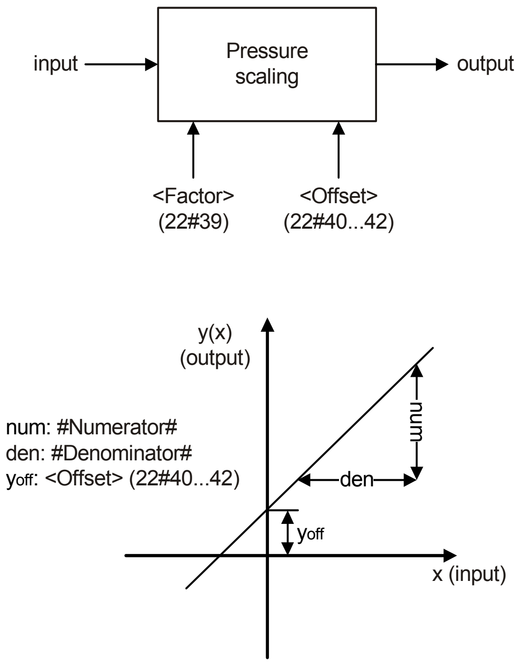
7.4.3.2 Object 22#36…38: Lower Limit

ValvePositionControl_DemandValueGenerator_Limit							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#36	LowerLimit	0x63A1#1	INT16	rw	Y	–32767…<UpperLimit> (0x63A0)	–32760
22#37	Unit	0x63A1#2	UINT8	ro	-	UINT8	0
22#38	Prefix	0x63A1#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.4.4 Scaling

This function is used to scale the pressure setpoint, e.g. to influence the input signal's value range. The output signal is calculated by multiplication of the input signal with a scaling factor and a subsequent addition of an offset according to the following figure.



$$y(x) = m \cdot x + y_{off}$$

$$m = \frac{num}{den}$$

Figure 69: Scaling function

7.4.4.1 Object 22#39: Factor

This parameter is a slope factor by which the input is multiplied. It is defined by two signed integer values, the numerator (upper 16 bits of the parameter) and the denominator (lower 16 bits of the parameter).

ValvePressureControl_DemandValueGenerator_Scaling							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
22#39	Factor	0x63A2#0	UINT32	rw	Y	UINT32	0x00010001

Value description

<Factor>				
Bit	31	16	15	0
Description	#Numerator#		#Denominator#	

Table 60: Data structure of the slope factor

7.4.4.2 Object 22#40…42: Offset

This parameter is the offset of the linear output function.

ValvePressureControl_DemandValueGenerator_Scaling							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
22#40	Offset	0x63A3#1	INT16	rw	Y	INT16	0
22#41	Unit	0x63A3#2	UINT8	ro	-	UINT8	0
22#42	Prefix	0x63A3#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.4.5 Ramp

The ramp function limits the slew rate of the input signal. The <Type> (22#43) parameter is used to activate a one-quadrant, two-quadrant or four-quadrant ramp or to deactivate the ramp function.

If the ramp function is running or the ramp function is stopped by the bit 15 (ramp stop) of the #ControlWord# signal the following <StatusWord> (0#38) bits are set:

<StatusWord> (0#38) bit	**Description**
9	This bit is set if the following conditions are true: Spool position and/or pressure ramp function is active and spool position and/or pressure function is running and #ControlWord# bit 15 is set to false.
15	This bit is set if the following conditions are true: Spool position and/or pressure ramp function is active and #ControlWord# bit 15 is set to true

⇨ Chapter "5.2.3 Object 0#38: Status word", page 48
⇨ Chapter "5.1.1 Object 0#41: Local", page 39
⇨ Chapter "5.1.2 Object 0#37: Control word", page 40
⇨ Chapter "5.1.3 Object 0#206: Local control word", page 41

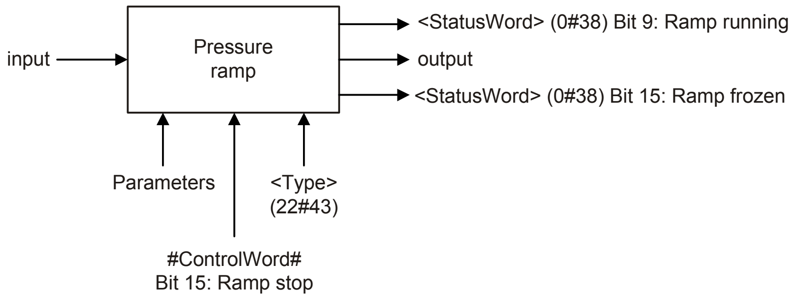


Figure 70: Ramp function

7.4.5.1 Object 22#43: Type

This parameter defines the progression of the ramp.

ValvePressureControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
22#43	Type	0x63B0#0	INT8	rw	Y	0…3	0

Value description

<Type>	Description
0	No ramp
1	One-quadrant ramp
2	Two-quadrant ramp
3	Four-quadrant ramp

Table 61: Possible values of parameter <Type> (22#43)

7.4.5.2 One-quadrant ramp (ramp type 1)

This function limits the input signal's rate of change to the defined <AccelerationTime> (22#44…46).

This ramp type is active, if the parameter <Type> (22#43) is set to 1.

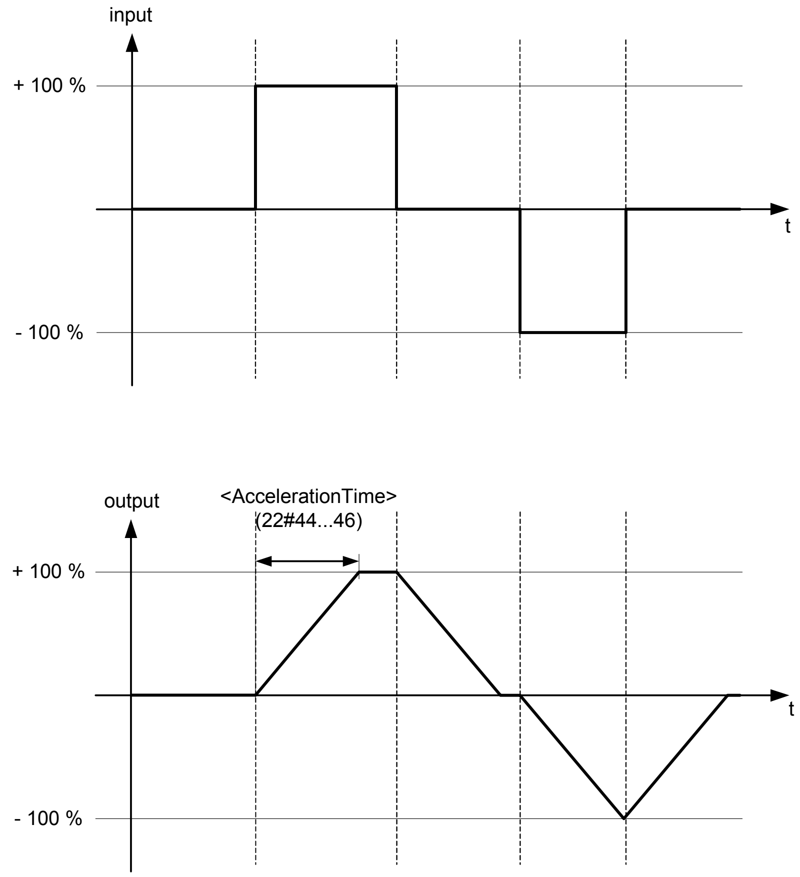


Figure 71: Ramp type 1

7.4.5.2.1 Object 22#44…46: Acceleration time

This parameter defines the output signal's maximum rate of change. The acceleration time corresponds to the time the signal needs for a change from 0 to 100 % as shown in the figure above. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#44	AccelerationTime	0x63B1#1	rw	Y	UINT16	0	rw
22#45	Unit	0x63B1#2	ro	-	UINT8	3	ro
22#46	AccelerationTime_Prefix	0x63B1#3	rw	Y	–4…0	–3	rw

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.4.5.3 Two-quadrant ramp (ramp type 2)

This function limits the input signal's rate of change to the defined <AccelerationTime> (22#44…46) and <DecclerationTime> (22#53…55).

This ramp type is active, if the parameter <Type> (22#43) is set to 2.

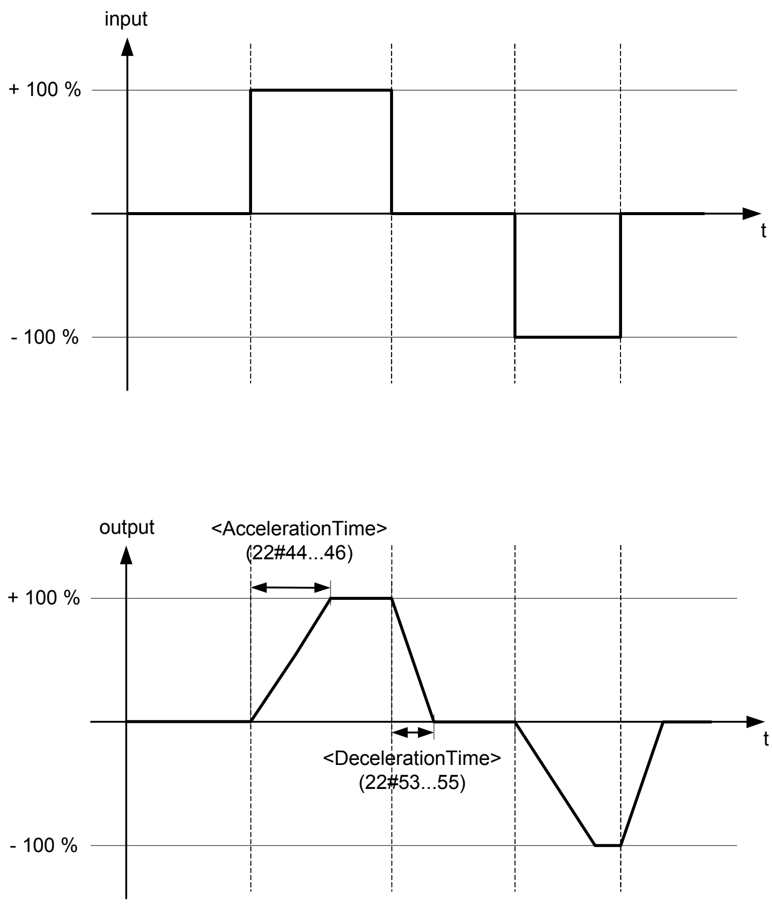


Figure 72: Ramp type 2

7.4.5.3.1 Object 22#44…46: Acceleration time

This parameter is the same as the acceleration time for ramp type 1.

⇨ Chapter "7.4.5.2.1 Object 22#44…46: Acceleration time", page 154

7.4.5.3.2 Object 22#53…55: Deceleration time

This parameter defines the output signal's maximum rate of change. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#53	DecelerationTime	0x63B4#1	UINT16	rw	Y	UINT16	0
22#54	Unit	0x63B4#2	UINT8	ro	-	UINT8	3
22#55	DecelerationTime_Prefix	0x63B4#3	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.4.5.4 Four-quadrant ramp (ramp type 3)

This function limits the input signal's rate of change to an acceleration time and a deceleration time, each separated for the positive and negative sides.

This ramp type is active, if the parameter <Type> (22#43) is set to 3.

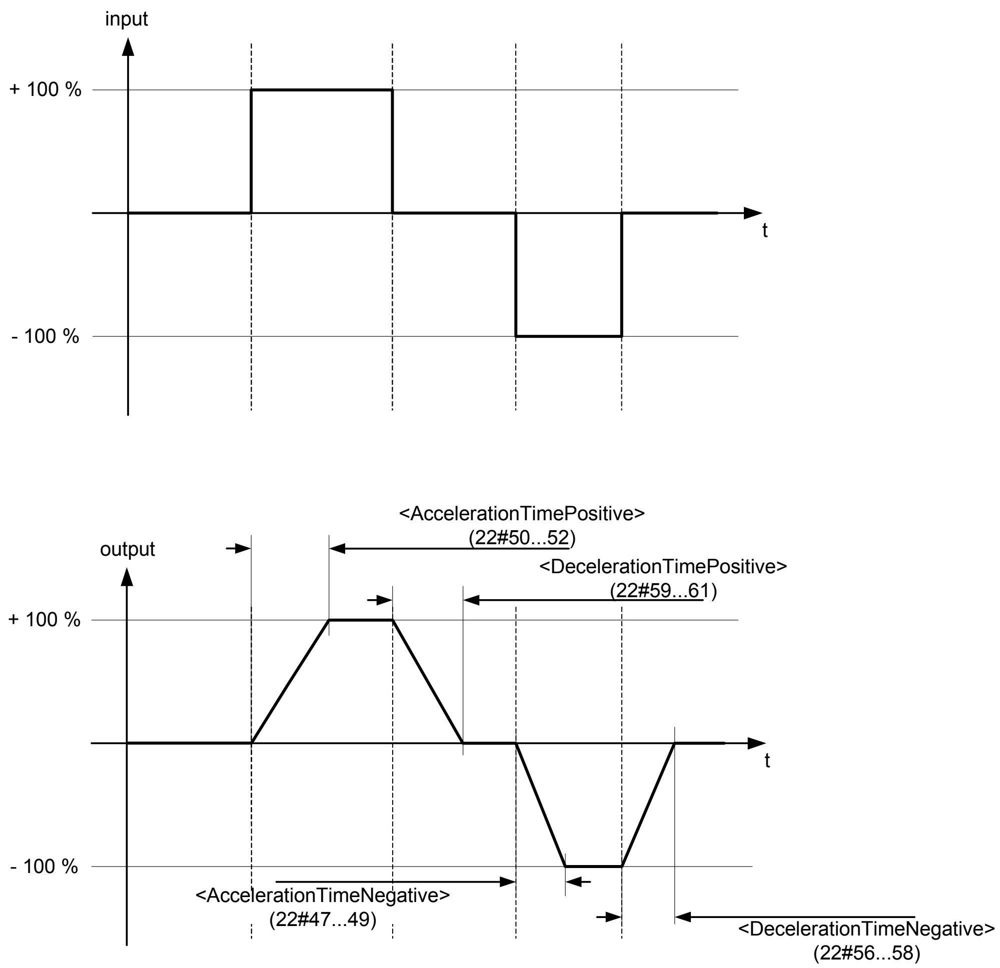


Figure 73: Ramp type 3

7.4.5.4.1 Object 22#50…52: Acceleration time positive

This parameter defines the output signal's maximum rate of change on the positive side. The acceleration time corresponds to the time that the signal needs for a change from 0 to 100 %. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#50	AccelerationTimePositive	0x63B2#1	UINT16	rw	Y	UINT16	0
22#51	Unit	0x63B2#2	UINT8	ro	-	UINT8	3
22#52	AccelerationTimePositive_Prefix	0x63B2#3	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.4.5.4.2 Object 22#47…49: Acceleration time negative

This parameter defines the output signal's maximum rate of change on the negative side. The acceleration time corresponds to the time that the signal needs for a change from 0 to 100 %. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#47	AccelerationTimeNegative	0x63B3#1	UINT16	rw	Y	UINT16	0
22#48	Unit	0x63B3#2	UINT8	ro	-	UINT8	3
22#49	AccelerationTimeNegative_Prefix	0x63B3#3	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.4.5.4.3 Object 22#59…61: Deceleration time positive

This parameter defines the output signal's maximum rate of change on the positive side. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#59	DecelerationTimePositive	0x63B5#1	UINT16	rw	Y	UINT16	0
22#60	Unit	0x63B5#2	UINT8	ro	-	UINT8	3
22#61	DecelerationTimePositive_Prefix	0x63B5#3	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.4.5.4.4 Object 22#56…58: Deceleration time negative

This parameter defines the output signal's maximum rate of change on the negative side. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#56	DecelerationTimeNegative	0x63B6#1	UINT16	rw	Y	UINT16	0
22#57	Unit	0x63B6#2	UINT8	ro	-	UINT8	3
22#58	DecelerationTimeNegative_Prefix	0x63B6#3	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.4.6 Pressure demand signal sign

In the pQ control mode, the <ControlMode> (0#40) is set to 5 (p/Q control), the following structure calculates the polarity of the pressure demand value. Negative pressure setpoint means that the pressure setpoint is effective on port B.

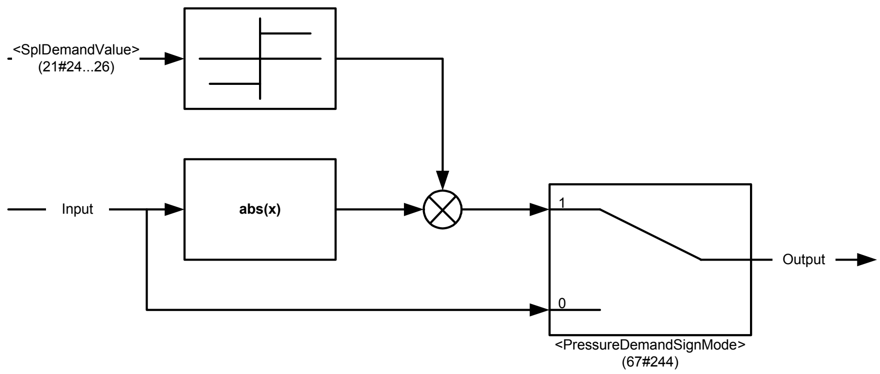


Figure 74: Pressure demand signal sign

7.4.6.1 Object 67#244: Pressure demand sign mode

With this parameter the pressure demand value sign can be parameterized to be dependent on the sign of the spool position setpoint value.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#244	PressureDemandSignMode	0x586D#0	UINT8	rw	Y	0…1	0

Value description

<PressureDemandSignMode>	**Description**
0	Sign of the pressure demand value will not be influenced.
1	Sign of the pressure demand value is the same as the sign of the spool position demand value.

Table 62: Possible values of parameter <PressureDemandSignMode> (67#244)

7.5 Pressure controller

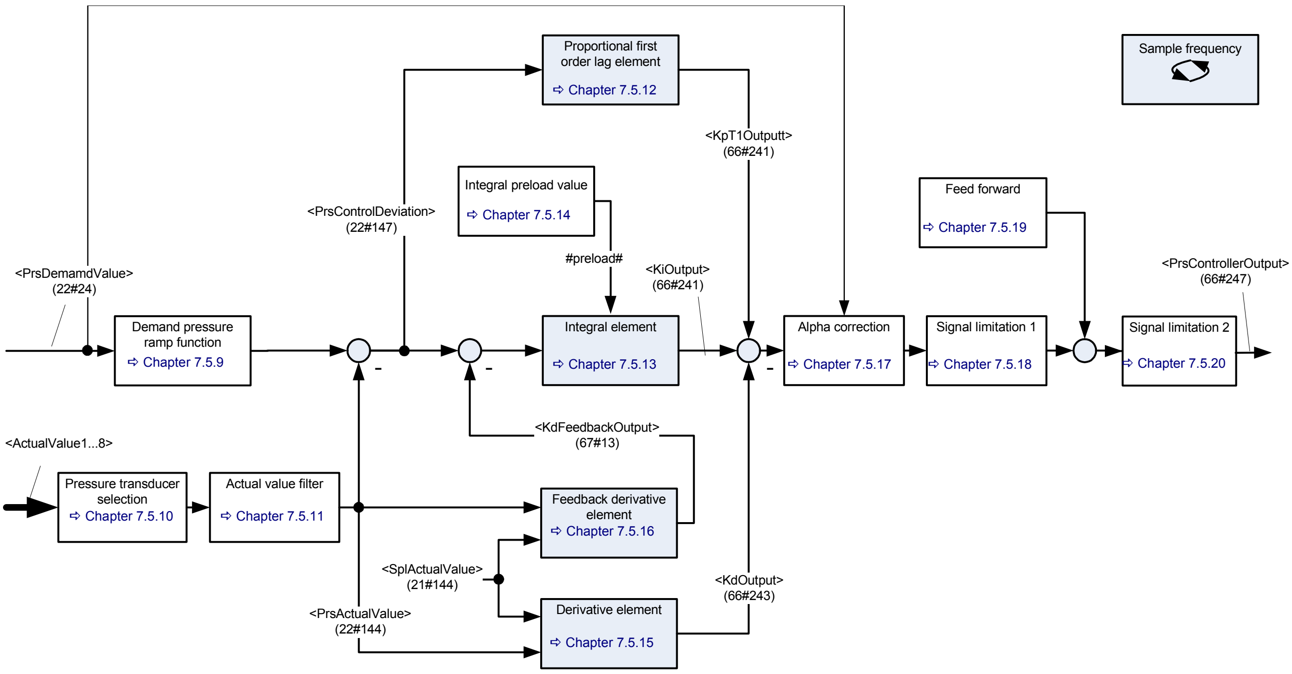


Figure 75: Pressure controller

➪ Chapter "7.4.1 Object 22#24…26: Demand value", page 149
➪ Chapter "7.5.2 Object 22#147…149: Control deviation", page 159
➪ Chapter "7.5.5 Object 66#243: Kd output", page 159
➪ Chapter "7.5.6 Object 67#13: Kd feedback output", page 159
➪ Chapter "7.5.4 Object 66#241: Ki output", page 159
➪ Chapter "7.5.3 Object 66#242: Kp T1 output", page 159
➪ Chapter "7.5.7 Object 66#247: Controller output", page 160

Description of the feedback signal <ActualValue1…8> (1#87…1#94):
➪ Chapter "6.3 Drive transducer interface", page 67

7.5.1 Object 22#144…146: Actual value

The filter output <PrsActualValue> (22#144…146) is the input of the pressure controller.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#144	PrsActualValue	0x6381#1	INT16	ro	-	INT16	None
22#145	Unit	0x6381#2	UINT8	ro	-	UINT8	0
22#146	Prefix	0x6381#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

7.5.2 Object 22#147…149: Control deviation

This parameter holds the deviation between the ramped pressure setpoint value and the filtered pressure actual value.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
22#147	PrsControlDeviation	0x63D0#1	INT16	ro	-	INT16	None
22#148	Unit	0x63D0#2	UINT8	ro	-	UINT8	0
22#149	Prefix	0x63D0#3	INT8	ro	-	INT8	0

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.5.3 Object 66#242: Kp T1 output

This parameter holds the output of the proportional element of the pressure controller.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
66#242	KpT1Output	0x2311#0	FLOAT32	ro	-	FLOAT32	None

7.5.4 Object 66#241: Ki output

This parameter holds the output of the integrator element of the pressure controller.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
66#241	KiOutput	0x2310#0	FLOAT32	ro	-	FLOAT32	None

7.5.5 Object 66#243: Kd output

This parameter holds the output of the first differential element of the pressure controller.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
66#243	KdOutput	0x2312#0	FLOAT32	ro	-	FLOAT32	None

7.5.6 Object 67#13: Kd feedback output

This parameter holds the output of the second differential element of the pressure controller.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
67#13	KdFeedbackOutput	0x5862#0	FLOAT32	ro	-	FLOAT32	None

7.5.7 Object 66#247: Controller output

This parameter holds the controller output of the pressure controller.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
66#247	ControllerOutput	0x2418#0	INT16	ro	-	INT16	None

7.5.8 Active parameter set number

The pressure controller contains objects to influence the control behavior.
To adapt the control behavior to different control conditions within the machine cycle, 16 parameter sets are provided. Within these sets most of the controller parameter can be configured.

Only one of the 16 parameter sets is used at the same time. The parameter set to be used is selected by <ActiveParameterSetNumber> (66#246). This parameter defines the parameter set number of the objects which is used for the pressure controller. The following table shows all pressure controller objects that are part of one parameter set.

Slot # Index	**CANopen SDO**	**Object name**
66#97…112	0x2303#1…16	Ramp slope
66#17…32	0x230D#1…16	Active transducer interface
66#33…48	0x230F#1…16	Transducer interface area B
66#113…128	0x2304#1…16	Proportional gain
66#129…144	0x230E#1…16	Proportional gain time constant
66#145…160	0x2305#1…16	Integrator gain
66#161…176	0x2306#1…16	Integrator factor
66#167…193	0x2307#1…16	Integrator control range
67#17…32	0x231A#1…16	Integrator upper output limit
67#33…48	0x231B#1…16	Integrator lower output limit
67#81…96	0x5861#1…16	Integrator proportional part P gain
66#193…208	0x2308#1…16	Differentiator gain
66#225…240	0x2309#1…16	Differentiator T1
67#97…112	0x5863#1…16	Differentiator gain 2
67#113…128	0x5864#1…16	Differentiator T1 2
67#49…64	0x230A#1…16	Upper output limit
67#65…80	0x230B#1…16	Lower output limit
67#161…176	0x5867#1…16	Feed forward gain
67#177…192	0x5868#1…16	Feed forward parameter
67#193…208	0x5870#1…16	Feed forward offset
67#129…144	0x5865#1…16	Upper controller output limit
67#145…160	0x5866#1…16	Lower controller output limit
67#209…224	0x586C#1…16	pQ switching mode
66#81…96	0x230C#1…16	Hydraulic capacity

Table 63: Pressure controller objects contained in a parameter set

The objects which are part of the parameter set are signed with an offset N (0...15) added to the index number. E.g. (66#113+N) according to the (<ActiveParameterSetNumber> (66#246) – 1).

7.5.8.1 Object 66#246: Active parameter set number

The servo valve provides parameter set 1 to parameter set 16. The <ActiveParameterSetNumber> (66#246) selects one of these 16 parameter sets. To calculate the correct Profibus address (slot#index) add N = (<ActiveParameterSetNumber> (66#246) – 1) to the index to the parameter of the first parameter set. (For the second parameter set you have to add 2–1 = 1)

Example: Object 66#97…112: Ramp slope. <ActiveParameterSetNumber> = 3.
The address of the ramp slope of parameter set 1 = 66#97
The address of the ramp slope of parameter set 3 = 66#97 + 3 – 1 = 66#99

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
66#246	ActiveParameterSetNumber	0x2350#0	INT8	rw	Y	1…16	1

7.5.9 Demand pressure ramp function

The one-quadrant ramp function limits the rate the demand pressure value rises or falls. The slope will be defined by the rising time <RampSlope> (66#97+N). The ramp is only active, if the parameter is greater zero.

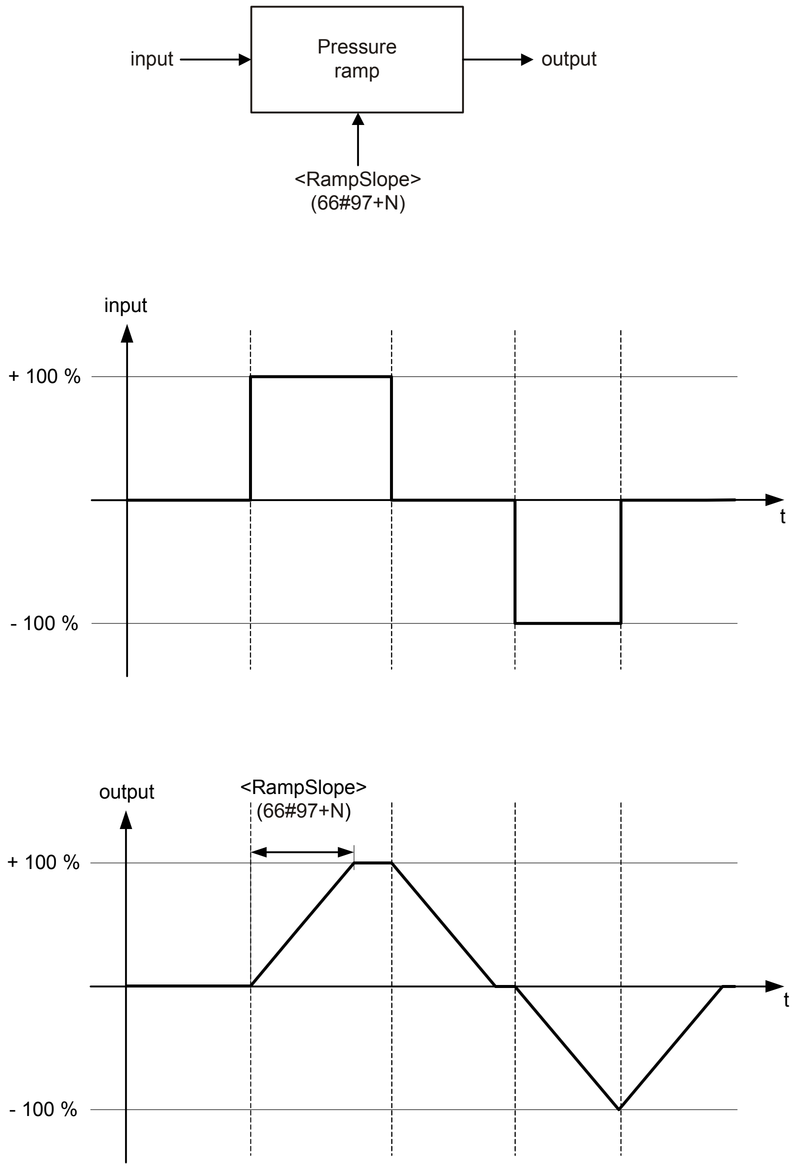


Figure 76: Demand pressure ramp function

7.5.9.1 Object 66#97…112: Ramp slope

If a 100 % step is set as input, the ramp output needs <RampSlope> (66#97+N) milliseconds to reach the 100 % ramp output.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
66#97…112	RampSlope	0x2303#1…16	UINT16	rw	Y	UINT16	0

7.5.10 Pressure transducer selection

The actual value for the pressure controller will be routed through the four possible transducer interfaces. Two different pressure control types depending on the pressure transducer selection are possible:

- Normal pressure control
  Only one pressure signal from the transducer interface <ActiveTransducerInterfaceAreaA> (66#17+N) is fed to the pressure controller. The <ActiveTransducerInterfaceAreaB> (66#33+N) is set to zero.
- Differential pressure control
  Two pressure signals from the transducer interfaces are fed to the pressure controller.
  One from <ActiveTransducerInterfaceAreaA> (66#17+N) for the pressure in the servo valve port A and one from the transducer interface <ActiveTransducerInterfaceAreaB> (66#33+N) for the pressure in the servo valve port B.

  For a differential cylinder, the resultant force can be calculated with the port pressures and the bore (A) and annulus (B) areas. For this the parameters <CylinderPistonDiameter> (66#248), <CylinderRodDiameterA> (66#249) and <CylinderRodDiameterB> (66#250) are used.

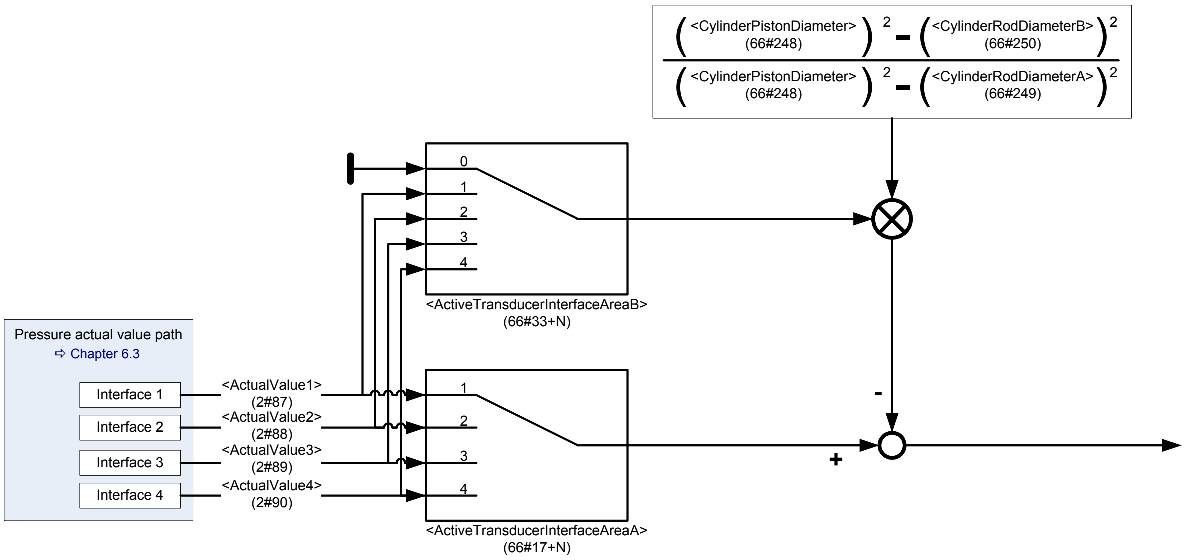


Figure 77: Pressure transducer selection

Set the parameter <ActiveTransducerInterfaceAreaB> (66#33+N) to 0 to switch off the differential pressure control.

7.5.10.1 Object 66#17…32: Active transducer interface area A

This parameter selects the transducer interface used as actual pressure value for the pressure controller. By default the pressure in the servo valve port A is linked to this actual pressure input.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
66#17…32	ActiveTransducerInterfaceAreaA	0x230D#1…16	INT8	rw	Y	1…4	1

7.5.10.2 Object 66#33…48: Active transducer interface area B

This parameter selects the transducer interface used for the second actual pressure value.

If a differential pressure control between the servo valve port A and B is required, this parameter selects the transducer interface used as the second pressure input. Normally the pressure in the servo valve port B is linked to this actual pressure input.
If only the pressure in port A is used, the <ActiveTransducerInterfaceAreaB> (66#33+N) must be set to zero.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
66#33…48	ActiveTransducerInterfaceAreaB	0x230F#1…16	INT8	rw	Y	0…4	None

7.5.10.3 Object 66#248: Cylinder piston diameter

The piston diameter is necessary to calculate the force of the cylinder.

The units used must be consistent with the units of the parameters <CylinderRodDiameterA> (66#249) and <CylinderRodDiameterB> (66#250).

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
66#248	CylinderPistonDiameter	0x585F#0	FLOAT32	rw	Y	0.0…+inf	1000000.0

7.5.10.4 Object 66#249: Cylinder rod diameter A

The rod diameter A is necessary to calculate the force of the cylinder.

The units used must be consistent with the units of the parameters <CylinderPistonDiameter> (66#248) and <CylinderRodDiameterB> (66#250).

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
66#249	CylinderRodDiameterA	0x585D#0	FLOAT32	rw	Y	0.0…<CylinderPistonDiameter> (66#248)	0.0

7.5.10.5 Object 66#250: Cylinder rod diameter B

The rod diameter is necessary to calculate the force of the cylinder.

The units used must be consistent with the parameter units of the parameters <CylinderPistonDiameter> (66#248) and <CylinderRodDiameterA> (66#249).

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
66#250	CylinderRodDiameterB	0x585E#0	FLOAT32	rw	Y	0.0…<CylinderPistonDiameter> (66#248)	0.0

7.5.11 Actual value filter

The parameters <ActualPressureFilterCutoffFrequency> (67#1) and <ActualPressureFilterOrder> (67#2) are used to set the behavior of the Butterworth filter. <ActualPressureFilterCutoffFrequency> (67#1) specifies the cutoff frequency of the filter in Hz. The order of the filter is set with the parameter <ActualPressureFilterOrder> (67#2) (possible values: 1…3).

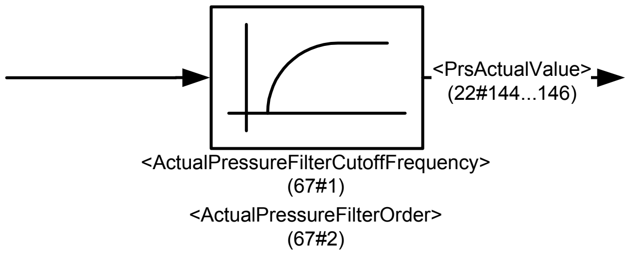


Figure 78: Actual value filter

7.5.11.1 Object 67#1: Actual pressure filter cutoff frequency

This parameter specifies the cutoff frequency of the Butterworth filter in Hz.

The value 0.0 disables the filter.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#1	ActualPressureFilterCutoffFrequency	0x23F2#0	FLOAT32	rw	Y	0.0 | 10.0…10000.0/3.0	None

7.5.11.2 Object 67#2: Actual pressure filter order

This parameter sets the order of the Butterworth filter.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#2	ActualPressureFilterOrder	0x23F3#0	UINT8	rw	Y	1…3	1

7.5.12 Proportional first order lag element (PT1)

The proportional part of the output is generated by a proportional gain (P-element) and first order lag element (PT1-element). The input signal is the pressure control deviation.

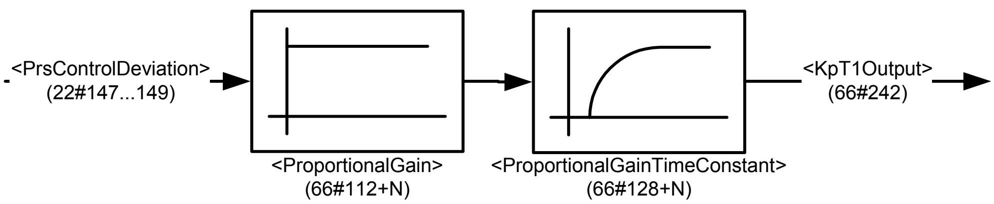


Figure 79: Proportional first order lag element (PPT1)

7.5.12.1 Object 66#113…128: Proportional Gain

This parameter sets the proportional gain.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
66#113…128	ProportionalGain	0x2304#1…16	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.12.2 Object 66#129…144: Proportional gain time constant

This parameter sets the time constant in seconds of the first order lag element (PT1).

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
66#129…144	ProportionalGainTimeConstant	0x230E#1…16	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.13 Integrator element (I)

To ensure a bumpless transfer between spool position and pressure control, the pressure integrator can be set to a defined preload (#Preload#) value before switching to the pressure control.

⇨ Chapter "7.5.14 Integrator preload value", page 168

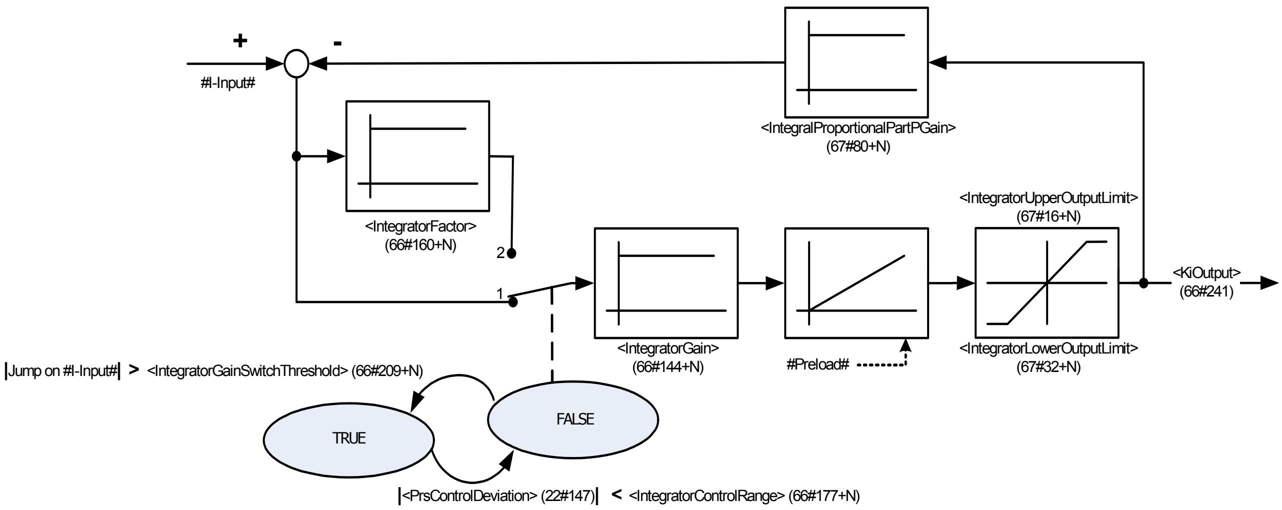


Figure 80: Integrator element (I)

7.5.13.1 Object 66#145…160: Integrator gain

This parameter contains the integrator gain.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
66#145…160	IntegratorGain	0x2305#1…16	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.13.2 Object 66#161…176: Integrator factor

This parameter contains an additional factor which is multiplied to the integrator gain if the control error is larger than the <IntegratorControlRange>.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
66#161…176	IntegratorFactor	0x2306#1…16	FLOAT32	rw	Y	0.0…+inf	0.1

7.5.13.3 Object 66#177…192: Integrator control range

This parameter contains the range of the control deviation for the integrator part. If the control deviation is outside the range, the <IntegratorGain> is multiplied with the <IntegratorFactor> (which is zero by default).

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
66#177…192	IntegratorControlRange	0x2307#1…16	INT16	rw	Y	0…16384	163

7.5.13.4 Object 66#209…224: Integrator gain switch threshold

This parameter contains the value of the height of the step / jump of the I-Input dx/dt projected of one second. If a step greater than <IntegratorGainSwitchThreshold> is detected on the I-Input, the I-Input will be multiplied with the smaller <IntegratorFactor> as long as |<PrsControlDeviation>| is less than <IntegratorControlRange>.

This means if a step of 1 bit is detected (if the valve is used with analog inputs we always detect a noise of min. 1 bit), the default value doesn't make really sense, but the default value has to be kept to be compatible to further versions.

The following example shows the calculation:

1 Bit / IRQ -> 1 / 100 µs -> this means 10,000 / second

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
66#209…224	IntegratorGainSwitchThreshold	0x5857#1…16	UINT32	rw	Y	UINT32	5000

7.5.13.5 Object 67#17…32: Integrator upper output limit

This parameter contains the upper limit of the integrator output.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#17…32	IntegratorUpperOutputLimit	0x231A#1…16	INT16	rw	Y	<IntegratorLowerOutputLimit> (67#33+N)…32767	16384

7.5.13.6 Object 67#33…48: Integrator lower output limit

This parameter contains the lower limit of the integrator output.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#33…48	IntegratorLowerOutputLimit	0x231B#1…16	INT16	rw	Y	–32768… <IntegratorUpperOutputLimit> (67#17+N)	–16384

7.5.13.7 Object 67#81…96: Integrator proportional part P gain

Using this feedback proportional gain, the integrator can be modified to a first order lag element.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#81…96	IntegralProportionalPartPGain	0x5861#1…16	FLOAT32	rw	Y	FLOAT32	0.0

7.5.14 Integrator preload value

To ensure a bumpless transfer between spool position control and pressure control, the pressure integrator can be set to a defined preload value (#Preload#).

The source of the integrator preload value is configurable with the parameter <IntegratorPreloadParameter> (67#242).

If the <IntegratorPreloadMode> (67#243) is 1, the preload value will become effective every time when switching from spool position control to pressure control.

To turn off the integrator preload function set <IntegratorPreloadMode> (67#243) to 0 (off).

Setting the <IntegratorPreloadMode> to 2, the preload value will become effective only once.
Attention: The <IntegratorPreloadMode> will return to either 1 or 0, depending on the former value [1]. This function is comparable to a push-button.

⇨ Chapter "7.5.13 Integrator element (I)", page 166

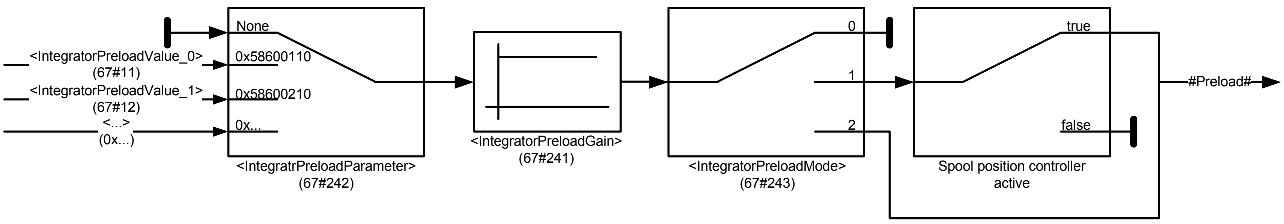


Figure 81: Integrator preload value

1 Former value means the value of the <IntegratorPreloadMode> before setting the value to 2 (usually to default value 1).

7.5.14.1 Object 67#243: Integrator preload mode

This parameter is to select the integrator preload mode.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
67#243	IntegratorPreloadMode	0x586B#0	UINT8	rw	Y	0…2	0

Value description

<IntegratorPreloadMode>	Behavior of preload output
0	The Preload function is off. The pressure integrator element is only set to zero during initialization of the controller.
1	**Spool position controller is active:** The pressure integrator element is set to the #Preload# value. **Pressure controller is active:** The preload function does not influence the pressure integrator element.
2	The pressure integrator element is set to the #Preload# value.The #Preload# value will become effective only once (one time overwrite).

Table 64: Possible values of parameter <IntegratorPreloadMode> (67#243)

7.5.14.2 Object 67#241: Integrator preload gain

This parameter contains the integrator preload gain.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
67#241	IntegratorPreloadGain	0x5869#0	FLOAT32	rw	Y	FLOAT32	0.0

7.5.14.3 Object 67#242: Integrator preload parameter

With this parameter every INT16 application parameter can be mapped as preload input. Per default the pressure setpoint <PrsSetpoint> (22#21) is mapped.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
67#242	IntegratorPreloadParameter	0x586A#0	UINT32	rw	Y	UINT32	0x63800110

Value description

<Parameter>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length: 0x10
Default	0x63	0x80	0x01	0x10

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

The default value is 0x63800110, which refers to the <PrsSetpoint> (22#21…23), with the CANopen index 0x6380 and the CANopen sub-index 0x01 with a length of 16 bit (16=0x10).

7.5.14.4 Object 67#11…12: Integrator preload values

This object contains two pre-calculated preload values.
These values can be mapped using the parameter <IntegratorPreloadParameter> (67#242) to the integrator preload input.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#11	IntegratorPreloadValue_0	0x5860#1	INT16	ro	-	INT16	None
67#12	IntegratorPreloadValue_1	0x5860#2	INT16	ro	-	INT16	None

Value description

<Parameter>	**Behavior of preload output**
<IntegratorPreloadValue1>	<SplDemandValue> (21#24) minus <ProportionalPart> (66#242)
<IntegratorPreloadValue2>	<SplDemandValue> (21#24) minus <ProportionalPart> (66#242) minus <FeedForwardOffset> (67#193…208)

Table 65: Behavior of preload output

7.5.15 Derivative element (PD)

This element differentiates the pressure actual value <PrsActualValue> (22#144…146) with a differentiator element including a first order filter with a time constant T1. Also a proportional gain element is implemented.

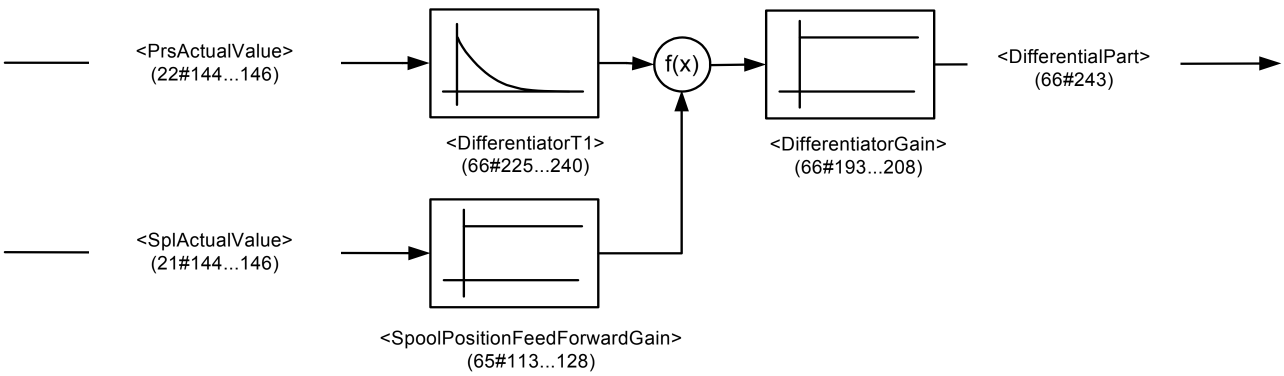


Figure 82: Derivative element (PD)

7.5.15.1 Object 66#193…208: Differentiator gain

This parameter contains the gain of the first differentiator.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
66#193…208	DifferentiatorGain	0x2308#1…16	FLOAT32	rw	Y	–inf…+inf	0.0

7.5.15.2 Object 66#225…240: Differentiator T1

This parameter contains the time constant of the first differentiator in seconds.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
66#225…240	DifferentiatorT1	0x2309#1…16	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.15.3 Object 65#113…128: Spool Position Feed Forward Gain

The gain controls the influence of the spool position to the derivative element (PD).

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
65#113…128	SpoolPositionFeedForwardGain	0x2324#1…16	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.16 Feedback derivative element (PD)

This element differentiates the pressure actual value <PrsActualValue> (22#144…146) with a real differentiator element including a first order filter with a time constant T1. A gain element is below this differentiator.

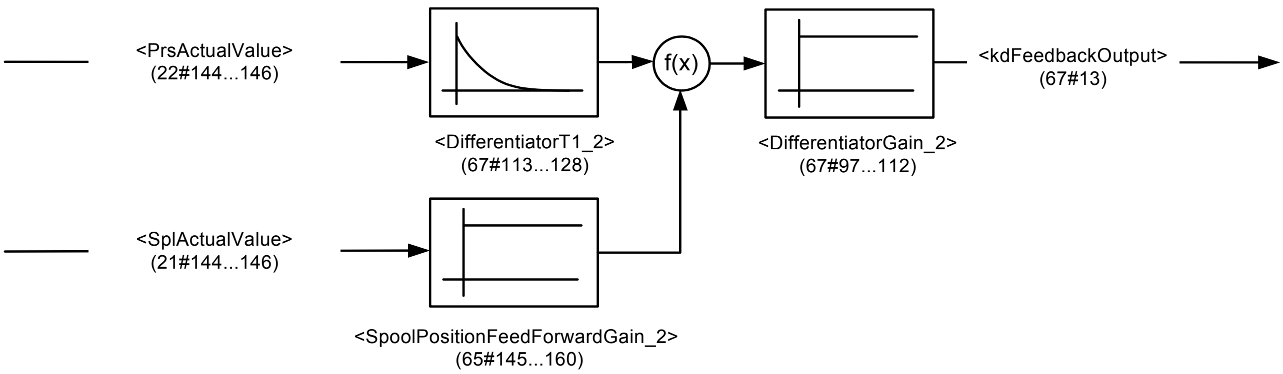


Figure 83: Feedback derivative element (PD)

7.5.16.1 Object 67#97…112: Differentiator gain 2

This parameter contains the gain of the second differentiator.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#97…112	DifferentiatorGain_2	0x5863#1…16	FLOAT32	rw	Y	–inf…+inf	0.0

7.5.16.2 Object 67#113…128: Differentiator T1 2

This parameter contains the time constant of the first differentiator in seconds.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#113…128	DifferentiatorT1_2	0x5864#1…16	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.16.3 Object 65#145…160: Spool Position Feed Forward Gain_2

This gain controls the influence of the spool position to the feedback derivative element (PD).

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
65#145…160	SpoolPositionFeedForwardGain_2	0x5858#1…16	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.17 Alpha correction

The alpha correction can be used to compensate the area differences between side A and side B of a differential cylinder. The alpha factor is only effective, if the pressure demand value <DemandValue> (22#24…26) is greater than zero.

⇨ Chapter "7.5.10 Pressure transducer selection", page 163
⇨ Chapter "7.5.10.3 Object 66#248: Cylinder piston diameter", page 164
⇨ Chapter "7.5.10.4 Object 66#249: Cylinder rod diameter A", page 164
⇨ Chapter "7.5.10.5 Object 66#250: Cylinder rod diameter B", page 164

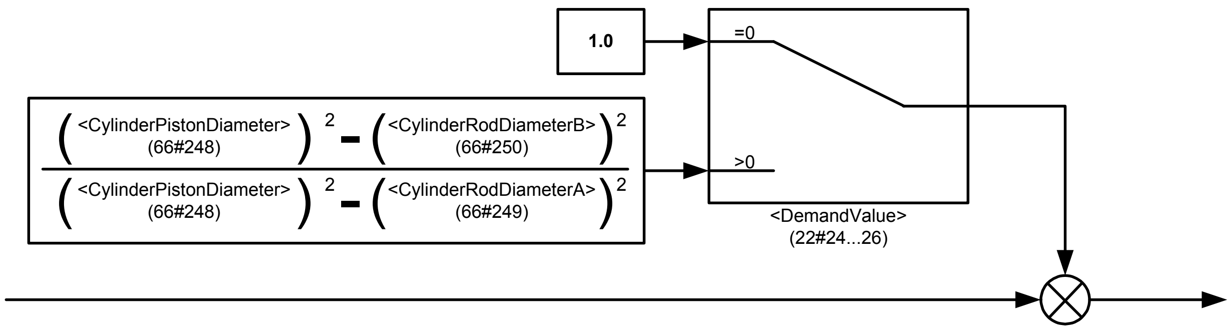


Figure 84: Alpha correction

7.5.18 Signal limitation 1

Signal limitation after the alpha correction.

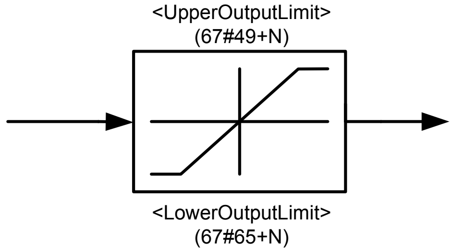


Figure 85: Signal limitation 1

7.5.18.1 Object 67#49…64: Upper output limit

This parameter contains the upper limit of the limiter.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#49…64	UpperOutputLimit	0x230A#1…16	INT16	rw	Y	<LowerOutputLimit> (67#65+N)…32767	16384

7.5.18.2 Object 67#65…80: Lower output limit

This parameter contains the lower limit of the limiter.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#65…80	LowerOutputLimit	0x230B#1…16	INT16	rw	Y	32767… <UpperOutputLimit> (67#49+N)	–16384

7.5.19 Feed forward

For some applications a feed forward control is advantageous. With the feed forward function a selectable signal, e.g. the pressure setpoint value, can be forwarded to the output of the controller. The signal can be scaled and an offset can be added.

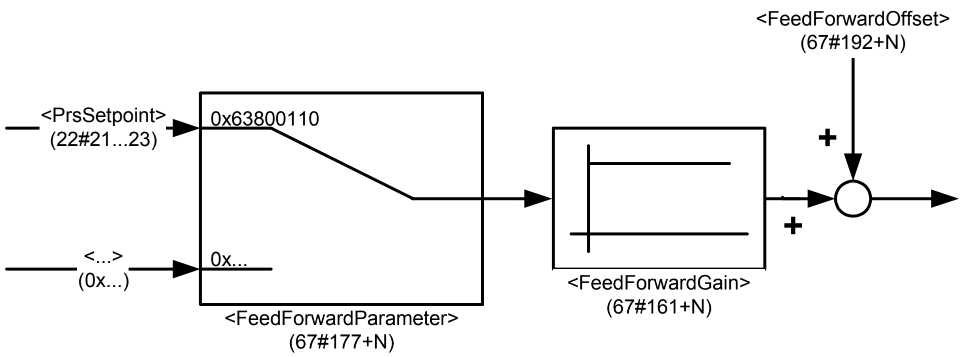


Figure 86: Feed forward

7.5.19.1 Object 67#161…176: Feed forward gain

This parameter contains the feed forward gain.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#161…176	FeedForwardGain	0x5867#1…16	FLOAT32	rw	Y	FLOAT32	0.0

7.5.19.2 Object 67#193…208: Feed forward offset

This parameter contains the feed forward offset.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
67#193…208	FeedForwardOffset	0x5870#1…16	INT16	rw	Y	INT16	0

7.5.19.3 Object 67#177…192: Feed forward parameter

With this parameter the source signal of the feed forward block will be selected. As default the pressure setpoint value <PrsSetpoint> (22#21) is mapped.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
67#177…192	FeedForwardParameter	0x5868#1…16	UINT32	rw	Y	UINT32	0x63800110

Value description

<Parameter>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length: 0x10
Default	0x63	0x80	0x01	0x10

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

The default value is 0x63800110, which refers to the <PrsSetpoint> (22#21...23), with the CANopen index 0x6380 and the CANopen), sub-index 0x01 with a length of 16 bit (16=0x10).

7.5.20 Signal limitation 2

This block limits the controller output.

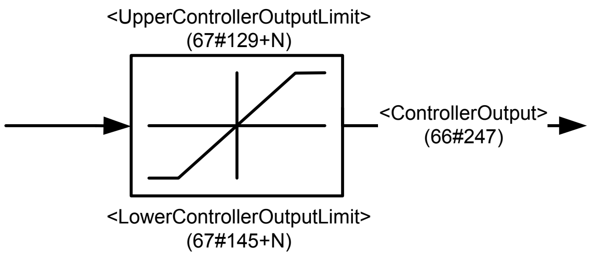


Figure 87: Signal limitation 2

7.5.20.1 Object 67#129…144: Upper controller output limit

This parameter contains the upper limit of the limitation.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
67#129…144	UpperControllerOutputLimit	0x5865#1…16	INT16	rw	Y	<LowerControllerOutputLimit> (67#145+N)…32767	16384

7.5.20.2 Object 67#145…160: Lower controller output limit

This parameter contains the lower limit of the limitation.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
67#145…160	LowerControllerOutputLimit	0x5866#1…16	INT16	rw	Y	–32768… <UpperControllerOutputLimit> (67#129+N)	–16384

7.5.21 Automatic parameterization of the pressure controller

The tuning of the proportional, integral & differential gains (P-I-D) of the pressure controller is not straightforward. The automatic parameterization simplifies this by using only one gain value, namely the hydraulic capacitance. There are two preconditions that have to be considered. The automatic parameterization is possible for:

- Pressure control where the oil volume is nearly constant
- Small pressure control range

In this case a simple linear model of the plant can be used. The dynamic parameters of the linearized servo valves are well known. Only the gain V_{qu} of the servo valve depends on the actual pressure. This issue can be solved by changing the hydraulic capacity, because this parameter influences the whole plant gain. Following parameters are used in the model:

Parameter name	Description
V_{qu}	Linear gain between setpoint value and flow (depends on working point)
D_v	Servo valve damping (depends on setpoint amplitude)
w_v	Servo valve natural frequency in [rad/s]
$C_H = \frac{V}{E_{Oil}}$	Hydraulic capacity C_H [10^{-6} l/bar] with: V Oil in the pipes volume [m^3] E_{Oil} Compressibility module ~$1.8*10^{-9}$ [Pa]
K_P	Pressure controller proportional gain (calculation depends on C_H)
K_I	Pressure controller integrator gain (calculation depends on C_H)
K_D	Pressure controller differential gain (calculation depends on C_H)
T_1	Pressure controller differential time constant (calculation depends on C_H)

Table 66: Parameters used in a linear plant model

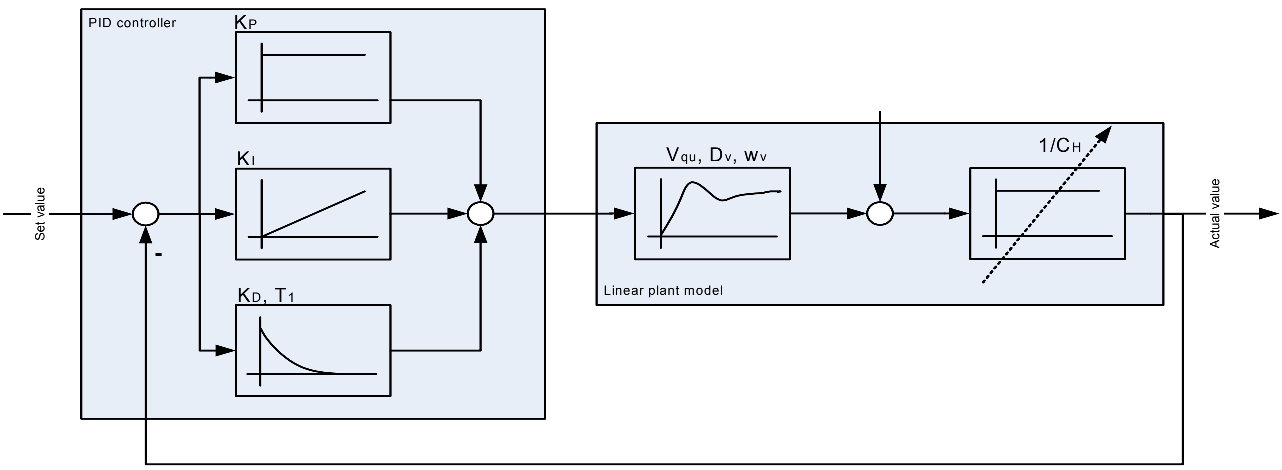


Figure 88: Parameterization of the pressure controller

An applicable way to find controller parameters for a stable system behavior is to use the 'worst case' pressure setpoint value for controller tuning. The worst case pressure setpoint value is the highest required value. The <SysPressureReference> (1#95) must be set for the used pressure sensor interface. The <HydraulicCapacity> (66#81+N) should be increased slowly up till the pressure controller behavior becomes unstable. Then it should be reduced until the controller becomes stable again. Setting the parameter <HydraulicCapacity> (66#81+N) to zero turns off the automatic parameter calculation.

7.5.21.1 Object 66#81…96: Hydraulic capacity

The parameter <HydraulicCapacity> (66#81…96) is defined as $\frac{V}{E_{Oil}}$. The unit is defined as [10^{-6} l/bar].

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
66#81…96	HydraulicCapacity	0x230C#1…16	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.21.2 Object 1#95: Sys Pressure Reference

This parameter holds the system pressure which is the reference to calculate the gains for the pressure controller in the automatic parameterization.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
1#95	SysPressureReference	0x231C#1	INT16	rw	N	INT16	400
1#96	Unit	0x231C#2	UINT8	ro	-	UINT8	0
1#97	Prefix	0x231C#3	INT8	ro	-	INT8	0

7.6 Spool position (Q) / pressure (P) switchover

The following structure is used to switch between spool position control and pressure control. This block is effective in the pQ control mode, the <ControlMode> (0#40) is set to 5 (p/Q-control servo valve). The output of this switch is routed to the spool position controller. Three strategies are implemented to influence the switchover criteria.

⇨ Chapter "7.1.1 Object 0#40: Control mode", page 119

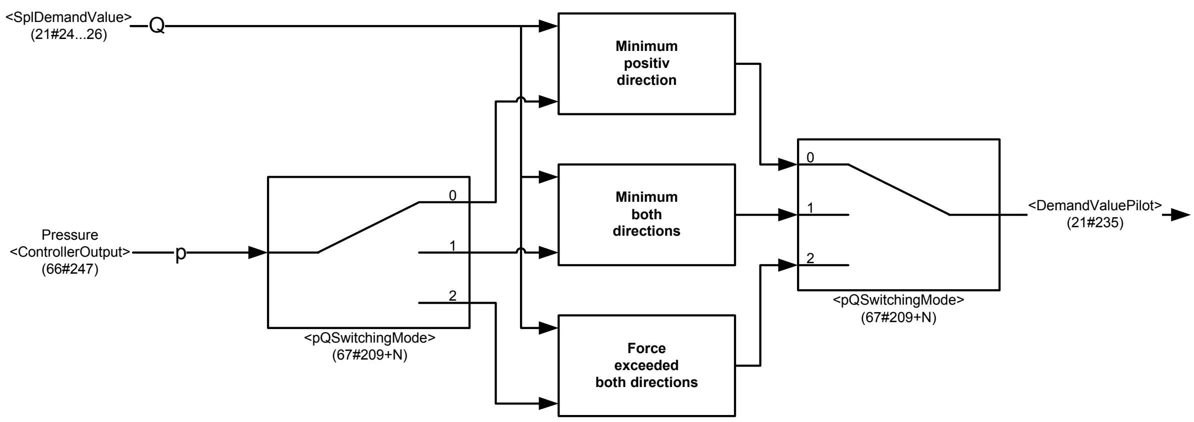


Figure 89: Spool position (Q) / pressure (P) switchover

7.6.1 Object 67#209…224: pQ switching mode

This parameter defines the p/Q switching mode.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
67#209…224	pQSwitchingMode	0x586C#1…16	UINT8	rw	Y	UINT8	0

Value description

<pQSwitchingMode>	Description
0	Minimum criterion in positive direction.
1	Minimum criterion in both directions.
2	Force exceeded in both directions.

Table 67: Possible values of parameter <pQSwitchingMode> (67#209+N)

7.6.1.1 Object 21#235: Demand value pilot

This parameter contains the output signal of the p/Q switchover function.

ValvePressureControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#235	DemandValuePilot	0x3300#0	INT16	ro	-	INT16	None

7.6.2 Minimum criterion in positive direction (switching mode 0)

If the <pQSwitchingMode> (67#209+N) is set to 0 (minimum criterion in positive direction), the following state machine is used to switch between spool position control and pressure control. In case of a two stage valve only mode 0 is possible and the parameter <DemandValvePilot> (21#235) is used instead of <SplDemandValue> (21#24…26).

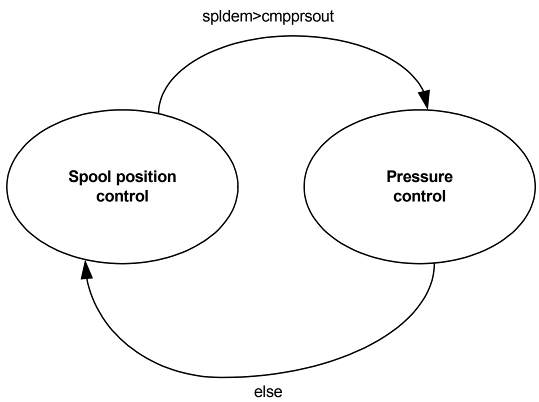


Figure 90: State machine used to switch between spool position control and pressure control

Value	Parameter
spldem	<SplDemandValue> (21#24…26)
cmpprsout	<ControllerOutput> (66#247)

7.6.3 Minimum criterion in both directions (switching mode 1)

If the <pQSwitchingMode> (67#209+N) is set to 1 (minimum criterion in both directions), the following state machine is used to switch between spool position control and pressure control.

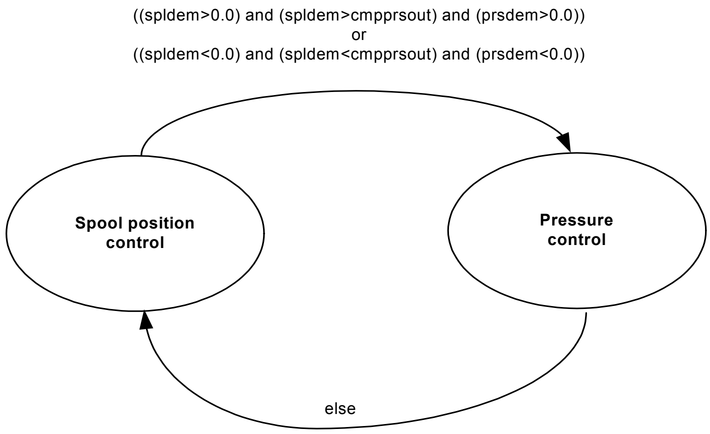


Figure 91: State machine used to switch between spool position control and pressure control

Value	Parameter
spldem	<SplDemandValue> (21#24…26)
prsdem	<PrsDemandValue> (22#24…26)
cmpprsout	<ControllerOutput> (66#247)

7.6.4 Force exceeded in both directions (switching mode 2)

If the <pQSwitchingMode> (67#209+N) is set to 2 (force exceeded in both directions), the following state machine is used to switch between spool position control and pressure control.

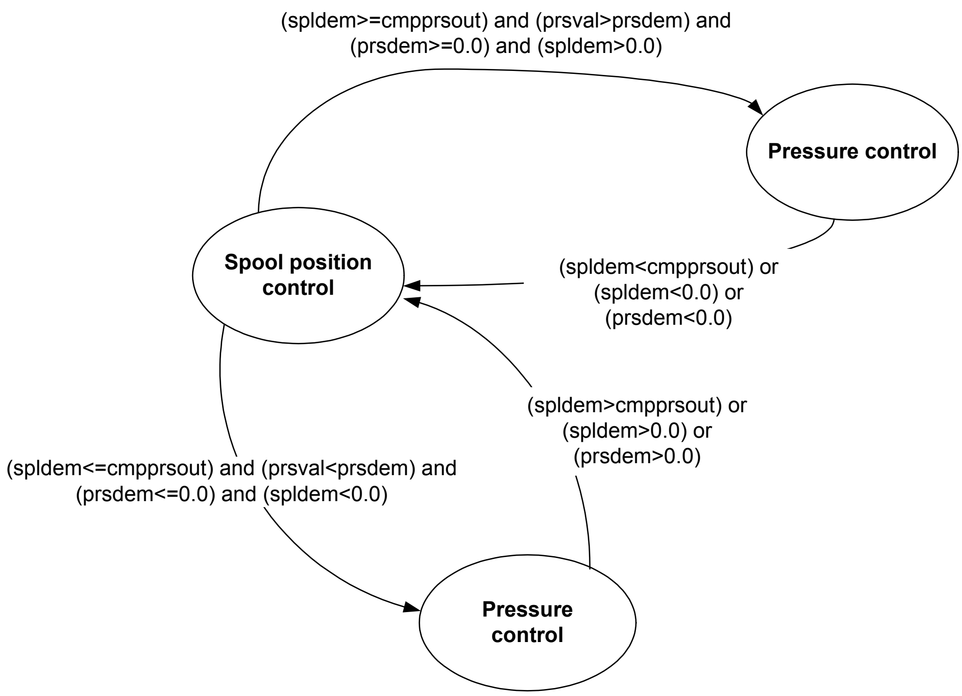


Figure 92: State machine used to switch between spool position control and pressure control

Value	Parameter
spldem	<SplDemandValue> (21#24…26)
prsdem	<PrsDemandValue> (22#24…26)
prsval	<PrsActualValue> (22#144…146)
cmpprsout	<ControllerOutput> (66#247)

7.7 Axis position setpoint conditioning / demand value generator

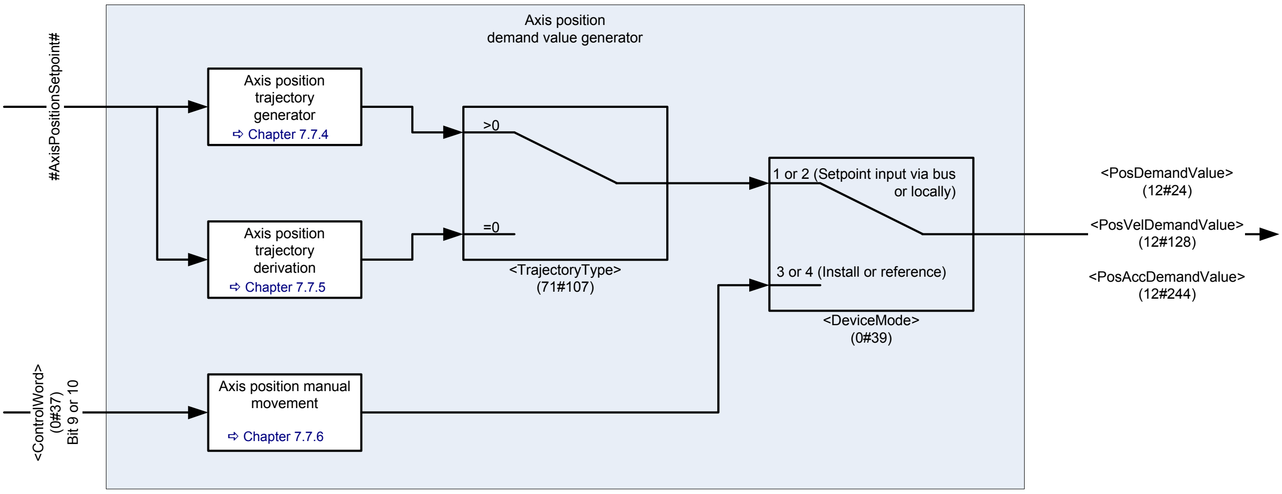


Figure 93: Axis position setpoint conditioning / demand value generator

The internal signal #AxisPositionSetpoint# is used to link the axis position setpoint value to the axis position demand value generator.
⇨ Chapter "6.2.5 Axis position setpoint value path", page 58

7.7.1 Object 12#24: Position demand value

The axis position demand value is generated from the #AxisPositionSetpoint# by means of the functions in the demand value generator and forwarded to the axis position controller.

DrivePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#24	PosDemandValue	0x6610#1	INT32	ro	-	INT32	None

7.7.2 Object 12#128: Velocity demand value

The axis velocity demand value is generated from the #AxisPositionSetpoint# by means of the functions in the demand value generator and forwarded to the axis position controller.

DrivePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#128	PosVelDemandValue	0x561E	INT32	ro	-	INT32	None

7.7.3 Object 12#244: Acceleration demand value

The axis acceleration demand value is generated from the #AxisPositionSetpoint# by means of the functions in the demand value generator and forwarded to the axis position controller.

DrivePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#244	PosAccDemandValue	0x5616	INT32	ro	-	INT32	None

7.7.4 Axis position trajectory generator

The trajectory generator is active, if the <TrajectoryType> (71#107) = 1 or 2. Then the trajectory generator allows a smooth <PosDemandValue> (12#24) if the <PosSetpoint> (12#12) jumps to another position.

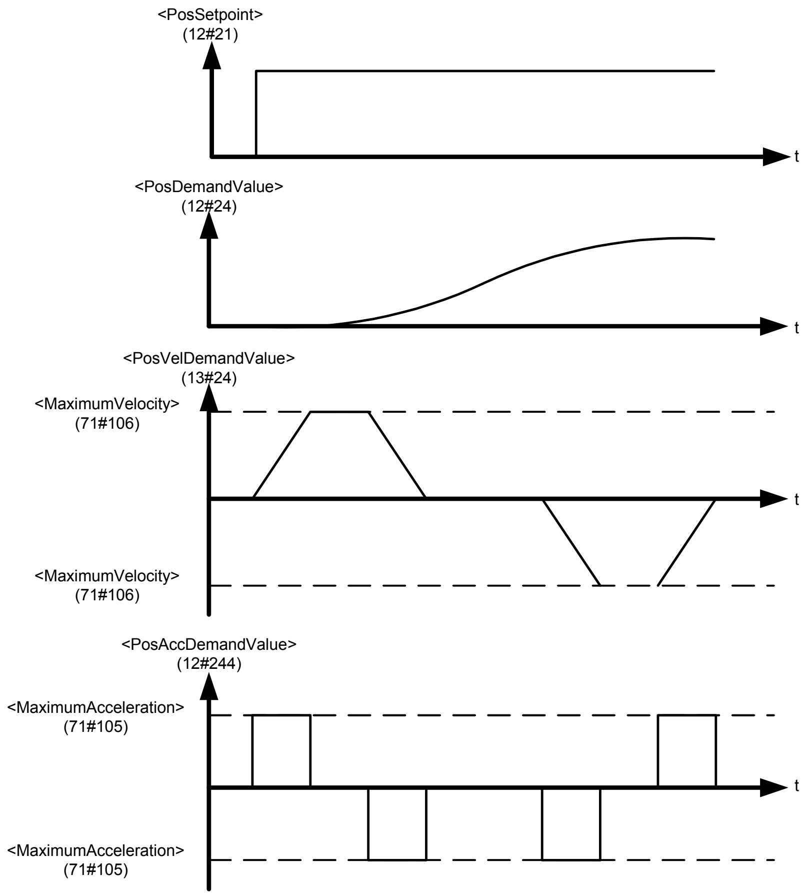


Figure 94: Axis position trajectory generator

The trajectory generator is switched off, if the <TrajectoryType> (71#107) = 0.
In this case the <PosDemandValue> (12#24) is equal to the <PosSetpoint> (12#12).
The <PosVelDemandValue> (13#24) is the derivation of the <PosDemandValue> (12#24).
The <PosAccDemandValue> (12#244) is the derivation of the <PosVelDemandValue> (13#24).

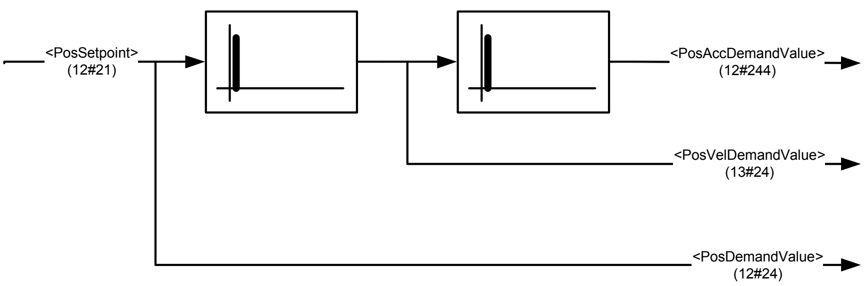


Figure 95: Axis position derivation

7.7.4.1 Object 71#107: Trajectory type

If <TractoryType> = 0 then the trajectory generator is switched off.
If <TractoryType> = 1 then the trajectory generator is enabled while the device state machine is in the 'ACTIVE' state.
If <TractoryType> = 2 then the trajectory generator is enabled while the device state machine is in the 'HOLD' or 'ACTIVE' state.

DrivePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#107	TractoryType	0x5700	INT8	rw	Y	0…2	0

7.7.4.2 Object 71#106: Maximum velocity

This parameter limits the maximal velocity of the axis.

DrivePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#106	MaximumVelocity	0x5703	INT32	rw	Y	0…2147483647	100000

7.7.4.3 Object 71#105: Maximum acceleration

This parameter limits the maximal acceleration of the axis.

DrivePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#105	MaximumAcceleration	0x5702	INT32	rw	Y	0…2147483647	100000

7.7.5 Axis position demand value filter

This first order lag element (PT1) is used to filter the axis position demand value.

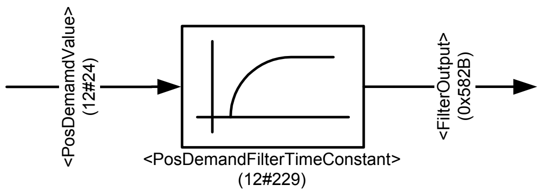


Figure 96: Axis position demand value filter

7.7.5.1 Object 12#229: Demand value filter time constant

This parameter sets the time constant in seconds of the first order lag element (PT1).

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
12#229	PosDemandFilterTimeConstant	0x582A	FLOAT32	rw	Y	0…+inf	0.0

7.7.6 Axis position manual movement

In the <DeviceMode> (0#39) = 3 (install mode) or = 4 (reference mode), the hydraulic axis can be moved manually using the trajectory generator. The axis can be moved in positive direction with bit 9 or in negative direction with bit 10 of the <ControlWord> (0#37). If an incremental encoder is used, then the <DeviceMode> (0#39) = 4 (reference mode) allows a homing position run using the Z pulse of the encoder. The maximal velocity is limited with the parameter <ManualMaxSpeed> (69#2).

➪ Chapter "6.2.1 Object 0#39: Device mode", page 53
➪ Chapter "5.1.2 Object 0#37: Control word", page 40
➪ Chapter "6.6.4 Incremental Encoder", page 89
➪ Chapter "7.7 Axis position setpoint conditioning / demand value generator", page 181

7.7.6.1 Object 69#2: Manual maximum speed

This maximum speed limits the manual velocity in <DeviceMode> (0#39) = 3 (install mode) or = 4 (reference mode).

DrivePositionControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#2	ManualMaxSpeed	0x5614	INT32	rw	Y	1000...2147483647	1000

7.8 Axis position controller

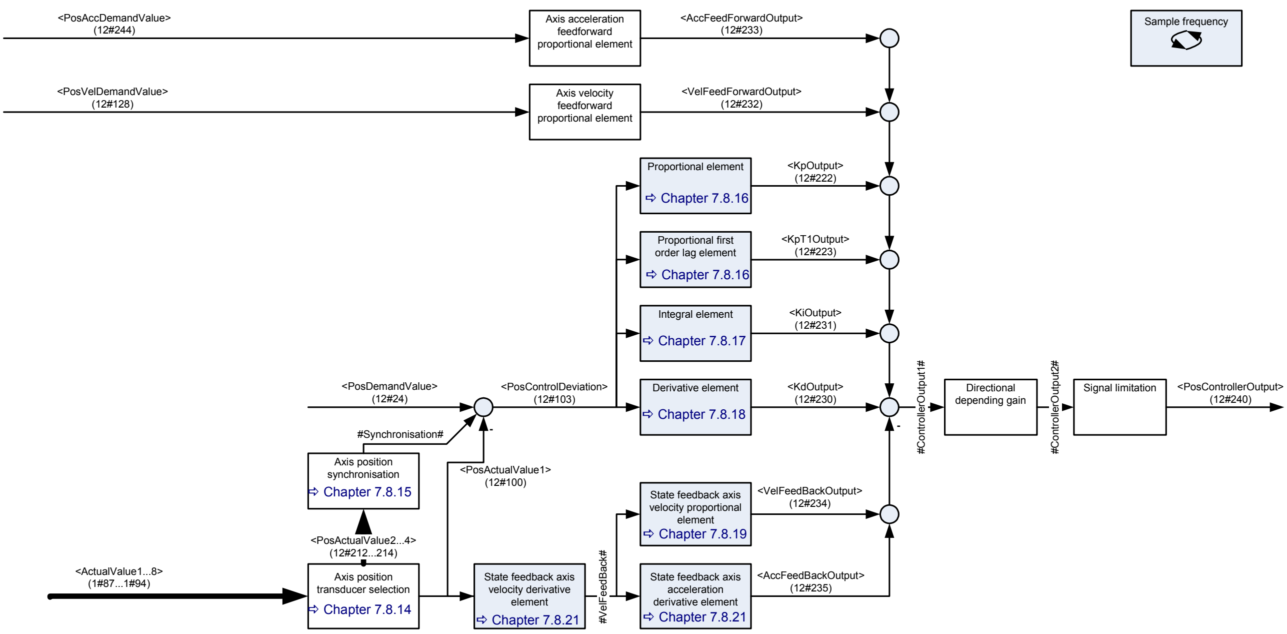


Figure 97: Axis position controller

7.8.1 Object 12#100, 12#212…214: Actual value

The actual position of the position controller has to be mapped to the parameter <PosActualValue1> (12#100). It is possible to synchronize the hydraulic axis with up to 3 other axis.

The actual positions of these 3 axis are expected on the parameters <PosActualValue2...4> (12#212…214). The mapping of the actual positions to the interface outputs is done with the transducer interface numbers <PosActualValueReference1...4> (12#20, 12#215…217).

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#100	PosActualValue1	0x6601#1	INT32	ro	-	INT32	None
12#212	PosActualValue2	0x6601#2	INT32	ro	-	INT32	None
12#213	PosActualValue3	0x6601#3	INT32	ro	-	INT32	None
12#214	PosActualValue4	0x6601#4	INT32	ro	-	INT32	None

7.8.2 Object 12#221: Demand filter output

This parameter holds the output value of the demand filter element.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#221	PosDemandFilterOutput	0x582B	FLOAT32	ro	-	FLOAT32	None

7.8.3 Object 12#103: Control deviation

This parameter holds the deviation between the axis position demand filter value and the axis position actual value.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#103	PosControlDeviation	0x6650	INT32	ro	-	INT32	None

7.8.4 Object 12#222: Kp output

This parameter holds the output of the proportional element of the axis position controller.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#222	KpOutput	0x552A	FLOAT32	ro	-	FLOAT32	None

7.8.5 Object 12#223: Kp T1 output

This parameter holds the output of the proportional first order lag element of the axis position controller.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#223	KpT1Output	0x552A#2	FLOAT32	ro	-	FLOAT32	None

7.8.6 Object 12#231: Ki output

This parameter holds the output of the integral element of the axis position controller.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#231	KiOutput	0x552C	FLOAT32	ro	-	FLOAT32	None

7.8.7 Object 12#230: Kd output

This parameter holds the output of the first differential element of the axis position controller.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#230	KdOutput	0x552B	FLOAT32	ro	-	FLOAT32	None

7.8.8 Object 12#232: Velocity feedforward output

This parameter holds the output of the axis velocity feedforward proportional element of the axis position controller.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#232	VelFeedForwardOutput	0x552E	FLOAT32	ro	-	FLOAT32	None

7.8.9 Object 12#233: Acceleration feedforward output

This parameter holds the output of the axis acceleration feedforward proportional element of the axis position controller.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#233	AccFeedForwardOutput	0x552F	FLOAT32	ro	-	FLOAT32	None

7.8.10 Object 12#234: Velocity feedback output

This parameter holds the output of the axis velocity feedback proportional element of the axis position controller.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#234	VelFeedBackOutput	0x5530	FLOAT32	ro	-	FLOAT32	None

7.8.11 Object 12#235: Acceleration feedback output

This parameter holds the output of the axis acceleration feedback proportional element of the axis position controller.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#235	AccFeedBackOutput	0x5531	FLOAT32	ro	-	FLOAT32	None

7.8.12 Object 12#240: Controller output

This parameter holds the controller output of the axis position controller.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#240	PosControllerOutput	0x550F	FLOAT32	ro	-	FLOAT32	None

7.8.13 Sample frequency

The sample frequency of the axis position controllers is the <BasicSampleFrequency> (0x3030) divided by the <PosSampleFrequencyDivider> (12#209).
⇨ Chapter "7.1.13.1 Object 0x3030: Basic sample frequency", page 127

7.8.13.1 Object 12#209: Sample frequency divider

This parameter holds the sample frequency divider.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#209	PosSampleFrequencyDivider	0x553F	UINT8	rw	Y	2…255	10

7.8.14 Axis position transducer selection

The transducer interfaces are reading and scaling the axis positions, the <PosActualValueReference1…4> (12#20, 12#215…217) is used to select the corresponding interfaces. The actual axis position of the controlled axis is expected on <PosActualValue1> (12#100). The positions of the synchronised axis are expected on <PosActualValue2…4> (12#212…214).

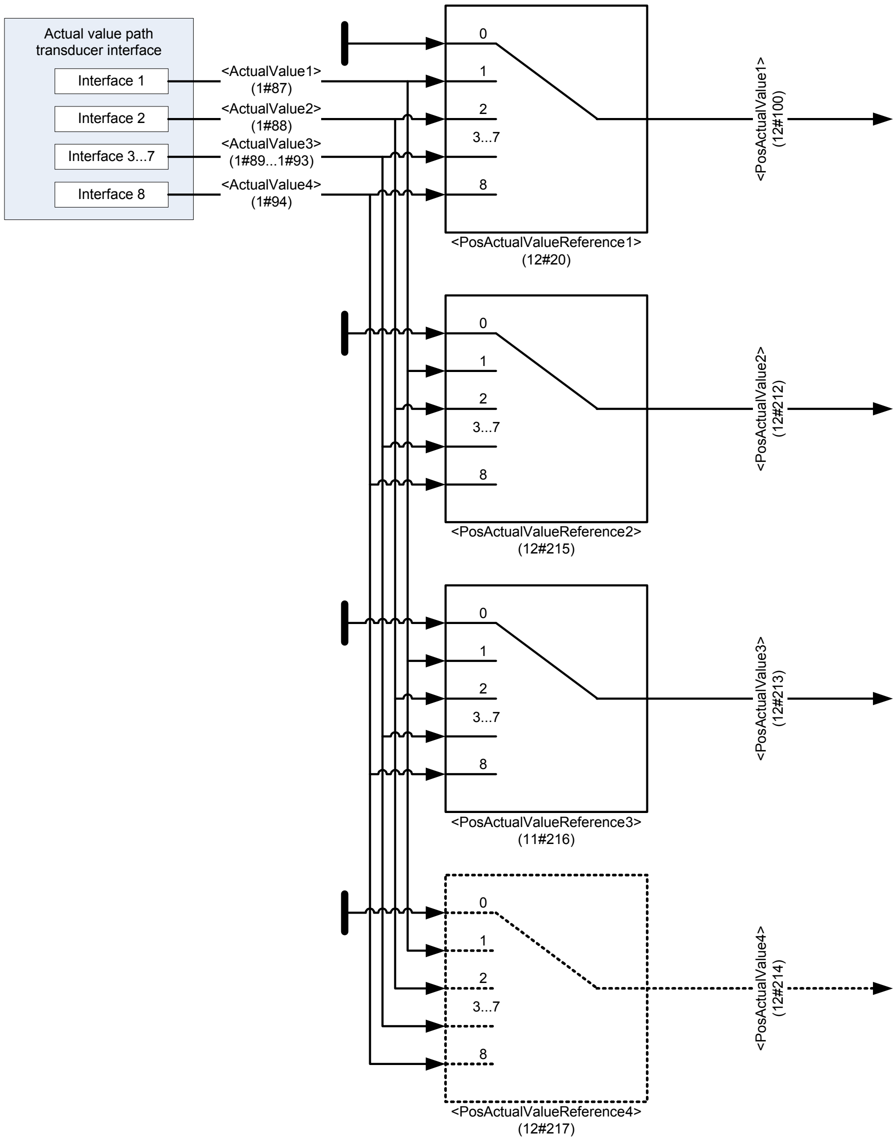


Figure 98: Axis position transducer selection

7.8.14.1 Object 12#20, 12#215…217: Actual value reference

This object allows to map four transducer interface inputs to the actual axis position inputs for the axis controller.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#20	PosActualValueReference1	0x6602#1	UINT8	rw	Y	0…8	1
12#215	PosActualValueReference2	0x6602#2	UINT8	rw	Y	0…8	0
12#216	PosActualValueReference3	0x6602#3	UINT8	rw	Y	0…8	0
12#217	PosActualValueReference4	0x6602#4	UINT8	rw	Y	0…8	0

7.8.15 Axis position synchronisation

This function allows the synchronisation of the actual position with the actual position of up to four external axis. For example: synchronise the pitch of one wing of a windmill with the pitch of the two other wings.

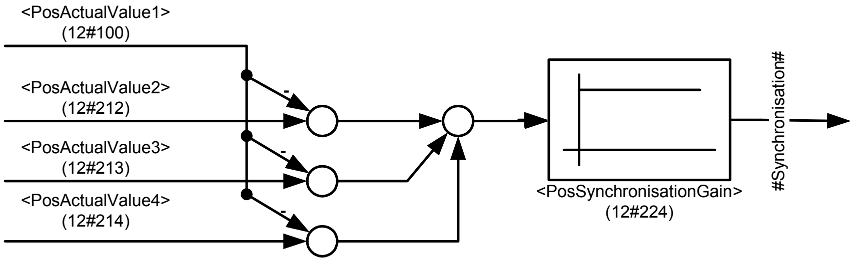


Figure 99: Axis position synchronisation

7.8.15.1 Object 12#224: Synchronisation gain

This parameter holds the gain of the proportional element of the axis position synchronisation. Zero will switch off the synchronous function.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#224	PosSynchronisationGain	0x5829	FLOAT32	rw	Y	FLOAT32	0.0

7.8.15.2 Object 12#249: Number of axis to synchronize

This parameter holds the number of axis which should be synchronous.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#249	NumberOfAxisToSynchronize	0x5830	INT8	rw	Y	1…4	3

7.8.16 Proportional element with first order lag element

The proportional part output is generated by the proportional gain parameter <ProportionalGainKp> (12#225) plus the first order lag element with the gain <ProportionalGainKpT1> (12#226) and the time constant <TimeConstantKpT> (12#206). The input signal is the axis position control deviation.

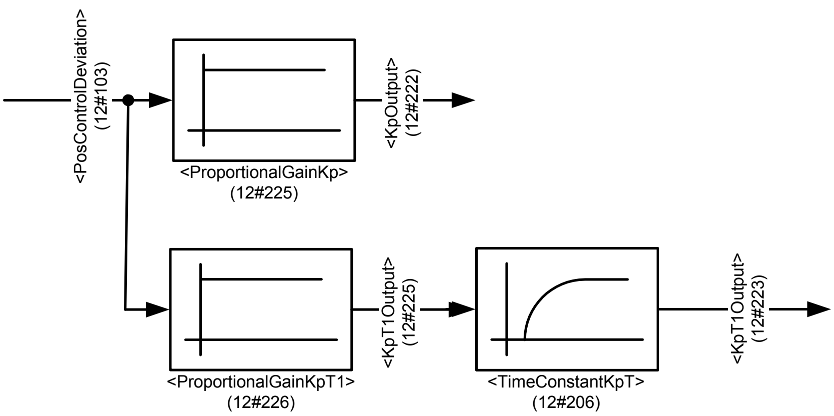


Figure 100: Proportional element with first order lag element

7.8.16.1 Object 12#225…226: Proportional gain

This object contains two proportional gain parameters. <ProportionalGainKp> (12#225) is the gain of the proportional part and <ProportionalGainKpT1> (12#226) is the gain of the first order lag element.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#225	ProportionalGainKp	0x5501#1	FLOAT32	rw	Y	FLOAT32	0.0
12#226	ProportionalGainKpT1	0x5501#2	FLOAT32	rw	Y	FLOAT32	0.0

7.8.16.2 Object 12#206: Time constant

This parameter sets the time constant in seconds of the first order lag element (PT1).

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#206	TimeConstantKpT	0x5508	FLOAT32	rw	Y	0.0…+inf	0.0

7.8.17 Integral element

The integral element is composed of an input depending gain, the integrator and a limiter.

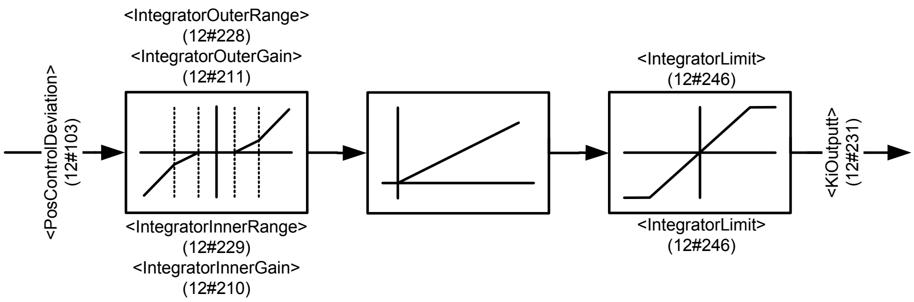


Figure 101: Integral element

7.8.17.1 Object 12#210…211: Integral gain

The gain of the integral part depends on the value of the absolute input signal |Input|.

Zero range: |Input| less than <IntegralInnerRange> (12#229). Output = Zero.
Inner range: |Input| between <IntegralInnerRange> (12#229) and <IntegralOuterRange> (12#228).

Output = <IntegralInnerGain> (12#210) x Input

Outer range: |Input| greater than <IntegralOuterRange> (12#228).

Output = <IntegralOuterGain> (12#211) x Input

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#210	IntegralInnerGain	0x5504#1	FLOAT32	rw	Y	FLOAT32	0.0
12#211	IntegralOuterGain	0x5504#2	FLOAT32	rw	Y	FLOAT32	0.0

7.8.17.2 Object 12#229: Integral inner range

Border of the absolute input signal to switch between gain = 0.0 and gain = <IntegralInnerGain> (12#210).

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#229	IntegralInnerRange	0x5511	FLOAT32	rw	Y	0.0…2147483647	0.0

7.8.17.3 Object 12#228: Integral outer range

Border of the absolute input signal to switch between gain = <IntegralInnerGain> (12#210) and gain = <IntegralOuterGain> (12#211).

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#228	IntegralOuterRange	0x5505	FLOAT32	rw	Y	0.0…2147483647	0.0

7.8.17.4 Object 12#246: Integral limit

Limit of the output signal of the integral element.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#246	IntegralLimit	0x5513	FLOAT32	rw	Y	0…+inf	16384000000

7.8.18 Derivative element

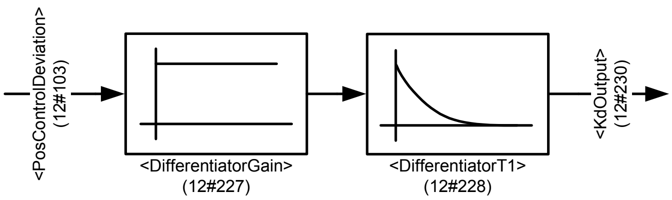


Figure 102: Derivative element

7.8.18.1 Object 12#227: Differentiator gain

This parameter contains the gain of the first differentiator.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#227	DifferentiatorGain	0x5502	FLOAT32	rw	Y	0…+inf	0.0

7.8.18.2 Object 12#208: Differentiator T1

This parameter contains the time constant of the first differentiator in seconds.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#208	DifferentiatorT1	0x5503	FLOAT32	rw	Y	0…+inf	0.0

7.8.19 Axis velocity feed forward proportional element

To realize a feed forward control for the velocity, the axis velocity demand value will be multiplied with this gain and added to the other parts of the controller.

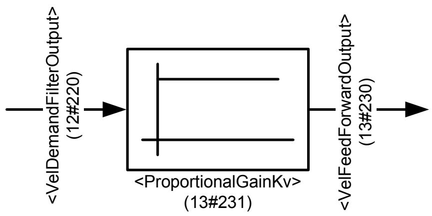


Figure 103: Axis velocity feed forward proportional element

7.8.19.1 Object 12#236: Velocity feed forward proportional gain

This parameter contains the velocity feed forward gain.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#236	ProportionalGainKv	0x5506	FLOAT32	rw	Y	FLOAT32	0.0

7.8.20 Axis acceleration feed forward proportional element

To realize a feed forward control for the acceleration, the axis acceleration demand value will be multiplied with this gain and added to the other parts of the controller.

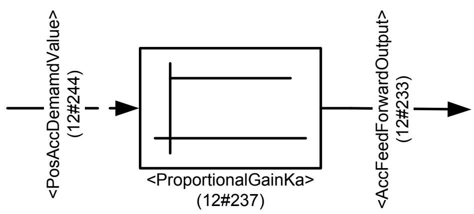


Figure 104: Axis acceleration feed forward proportional element

7.8.20.1 Object 12#237: Acceleration feed forward proportional gain

This parameter contains the velocity feed forward gain.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#237	ProportionalGainKa	0x5507	FLOAT32	rw	Y	FLOAT32	0.0

7.8.21 State feedback axis velocity and acceleration derivative elements

To realize a state space controller, the first derivation of the axis position values is necessary. To get a better signal, the axis position value is filtered with a first order lag element.

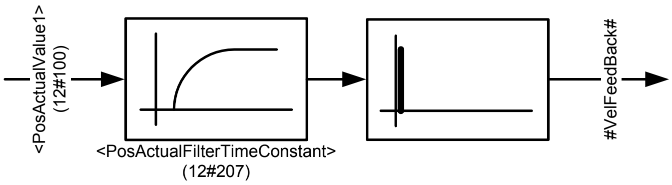


Figure 105: State feedback axis velocity and acceleration derivative element

7.8.21.1 Object 12#207: Actual value filter time constant

This parameter sets the time constant in seconds of the first order lag filter (PT1).

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#207	PosActualFilterTimeConstant	0x550C	FLOAT32	rw	Y	0.0…+inf	0.0

7.8.21.2 Object 12#239: State feedback axis velocity proportional gain

The first derivation of the axis position value multiplied with this gain will be added to the other parts of the controller.

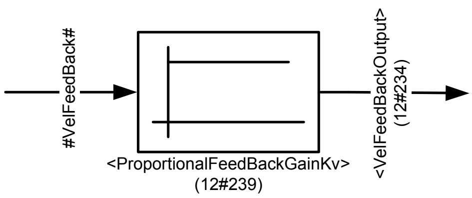


Figure 106: State feedback axis velocity proportional gain

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#239	ProportionalFeedBackGainKv	0x550B	FLOAT32	rw	Y	FLOAT32	0.0

7.8.21.3 Object 12#238: State feedback axis acceleration proportional gain

The second derivation of the axis position value (or the derivation of the #VelFeedBack#) is the acceleration feedback. To realize a state space controller, this second derivation multiplied with the <ProportionalFeedBackGainKa> (12#238) will be added to the other parts of the controller.

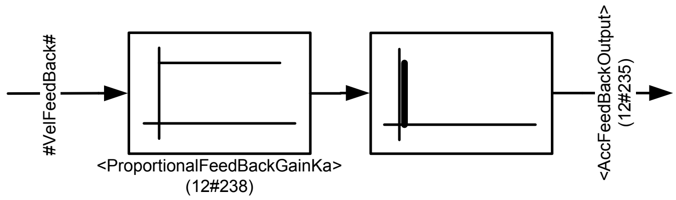


Figure 107: State feedback axis acceleration proportional gain

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#238	ProportionalFeedBackGainKa	0x550A	FLOAT32	rw	Y	FLOAT32	0.0

7.8.22 Directional depending gain

To compensate a not symmetrical movement between moving in and out a directional depending gain is implemented.

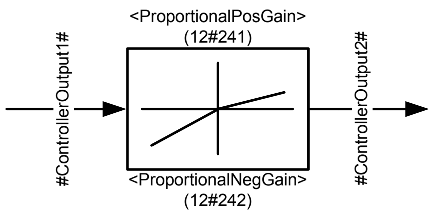


Figure 108: Directional depending gain

7.8.22.1 Object 12#241: Directional depending positive proportional gain

This gain is used while the input signal is positive.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#241	ProportionalPosGain	0x550D	FLOAT32	rw	Y	FLOAT32	1.0

7.8.22.2 Object 12#242: Directional depending negative proportional gain

This gain is used while the input signal is negative.

DrivePositionControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#242	ProportionalNegGain	0x550E	FLOAT32	rw	Y	FLOAT32	1.0

7.8.23 Signal limitation

The output of the controller is limited to ±16384 (±100 %).

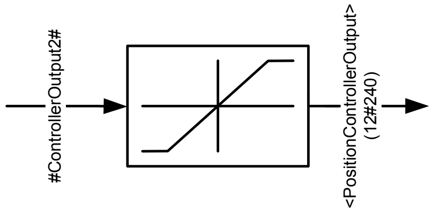


Figure 109: Signal limitation

7.9 Axis velocity setpoint conditioning / demand value generator

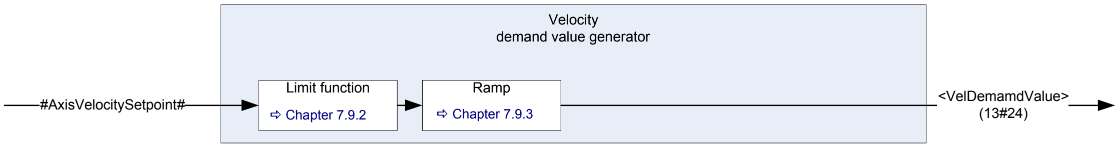


Figure 110: Axis velocity setpoint conditioning / demand value generator

The internal signal #AxisVelocitySetpoint# is used to link the pressure setpoint value to the axis velocity demand value generator.
⇨ Chapter "6.2.6 Axis velocity setpoint value path", page 62

7.9.1 Object 13#24: Demand value

The demand value indicated by this parameter is generated from the #AxisVelocitySetpoint# by means of the functions in the demand value generator and forwarded to the axis velocity controller.

ValveVelocityControl_DemandValueGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#24	VelDemandValue	0x5615	INT32	ro	N	INT32	None

7.9.2 Limit function

This function limits the value range of the input signal. The limit is defined by setting the upper and lower limit parameters.

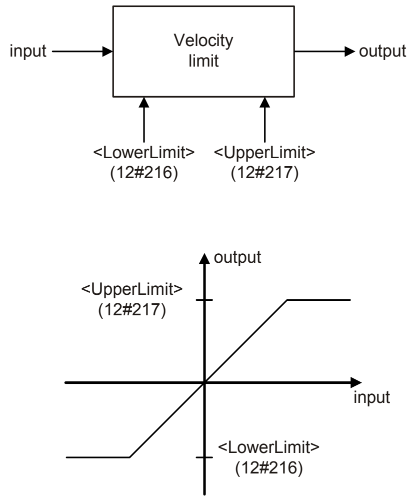


Figure 111: Limit function

The <UpperLimit> (12#216) must be greater than the <LowerLimit> (12#217).
If <LowerLimit> (12#217) will be set greater than the <UpperLimit> (12#216), the <UpperLimit> (12#216) will be set to the value of the <LowerLimit> (12#217).

7.9.2.1 Object 12#217: Upper limit

ValvePositionControl_DemandValueGenerator_Limit							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#217	UpperLimit	0x6521	INT16	rw	Y	<LowerLimit> (12#216)… 32767	16384

7.9.2.2 Object 12#216: Lower limit

ValvePositionControl_DemandValueGenerator_Limit							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#216	UpperLimit	0x6520	INT16	rw	Y	–32767… <UpperLimit> (12#217)	–16384

7.9.3 Ramp

The ramp function limits the slew rate of the input signal.

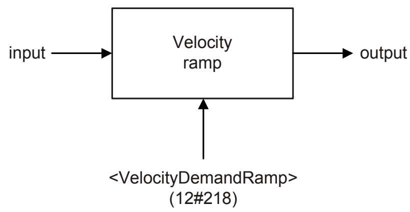


Figure 112: Ramp function

This function limits the output signal's rate of change to <VelocityDemandRamp> (12#218) per 1/10000 second. The ramp function is disabled (output = input) if this parameter is zero.

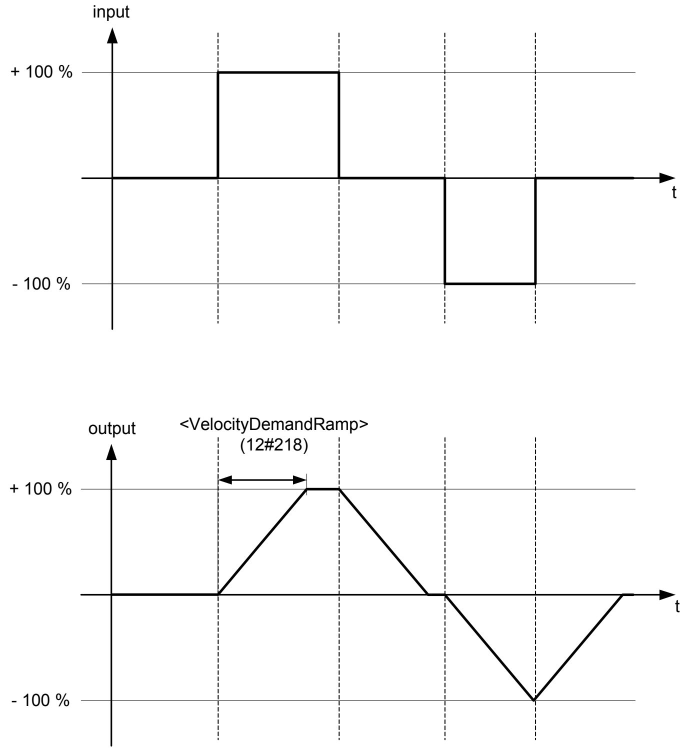


Figure 113: Velocity demand ramp

7.9.3.1 Object 12#218: Velocity demand ramp

This parameter defines the demand signal's maximum rate of change.

Max rate of change per second = <VelocityDemandRamp> (12#218) x 10000

ValveVelocityControl_DemandValueGenerator_Ramp							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#218	VelocityDemandRamp	0x5553	INT32	rw	Y	INT32	0

7.10 Axis velocity controller

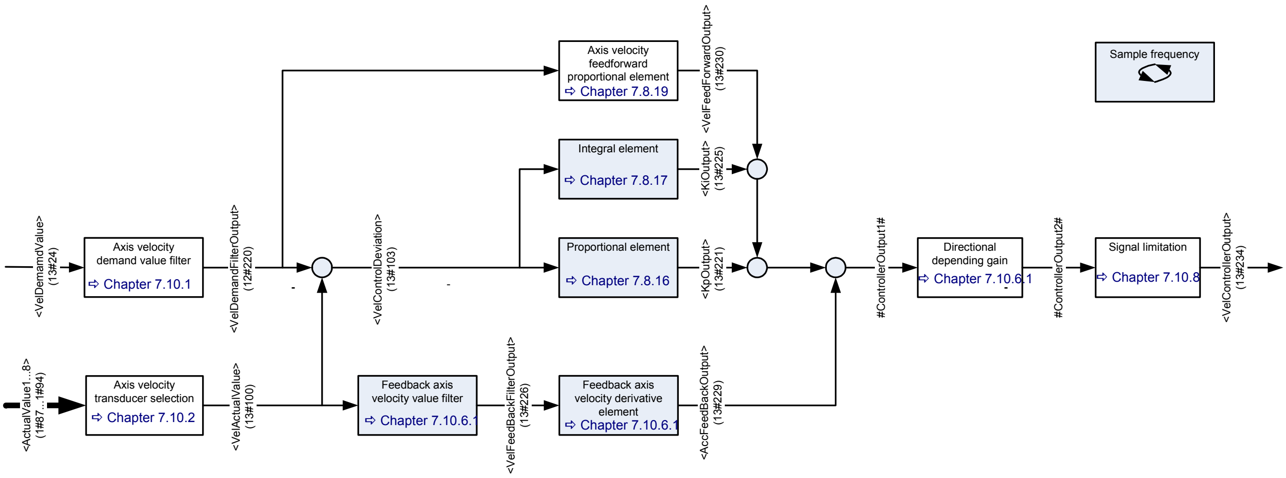


Figure 114: Axis velocity controller

7.10.1 Axis velocity demand value filter

The velocity demand value is filtered with a first order lag element (PT1).

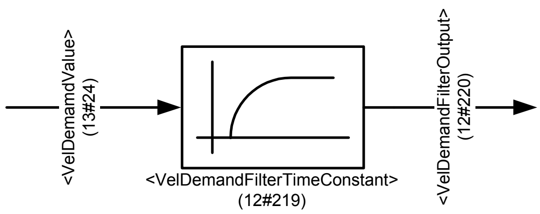


Figure 115: Axis velocity demand value filter

7.10.1.1 Object 12#219: Demand value filter time constant

This parameter sets the time constant in seconds of the first order lag element (PT1).

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#219	VelDemandFilterTimeConstant	0x5533	FLOAT32	rw	Y	0.0…+inf	0.0

7.10.1.2 Object 12#220: Demand value filter output

This parameter holds the output of the demand filter element.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#220	VelDemandFilterOutput	0x553B	FLOAT32	ro	-	FLOAT32	None

7.10.2 Axis velocity actual value

The axis velocity actual signal can either be read in via the transducer interface (if the encoder supplies a velocity signal) or it can be calculated by derivation from the actual position value.
<VelActualPathMode> (13#210) will select the used mode.
The parameter <VelActualValueReference> (13#20) will select the interface in case the encoder supplies the velocity signal. Otherwise the <PosActualValue1> (12#100) needs to be filtered and derived to get the actual velocity. In both modes, the velocity is multiplied with <VelActualGain> (13#213) to get the <VelActualValue> (12#100).

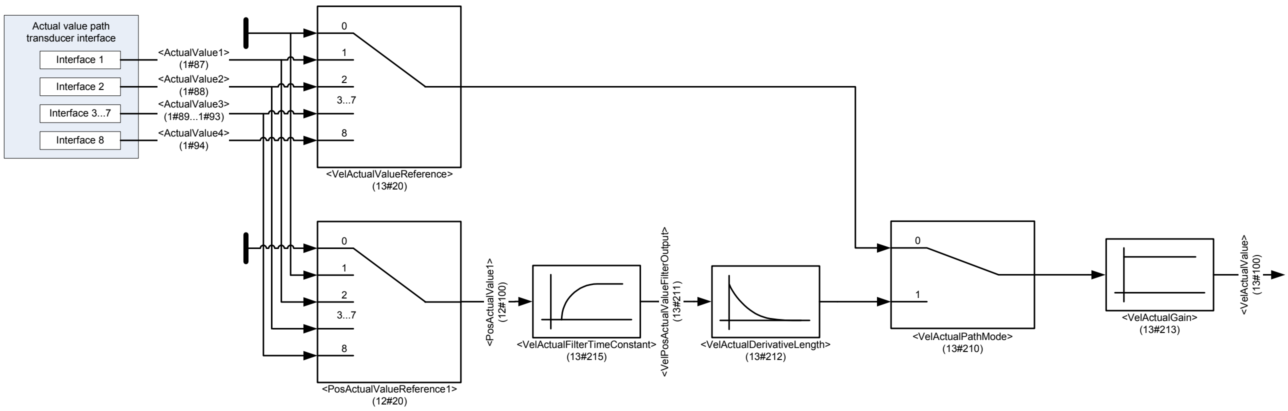


Figure 116: Axis velocity actual value

➪ Chapter "7.8.14.1 Object 12#20, 12#215…217: Actual value reference", page 188
➪ Chapter "7.8.1 Object 12#100, 12#212…214: Actual value", page 185
➪ Chapter "7.10.3.3 Object 13#103: Control deviation", page 202

7.10.2.1 Object 13#20: Actual value reference

This parameter selects the transducer with the actual velocity input.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#20	VelActualValueReference	0x5542	UINT8	rw	Y	0…8	1

7.10.2.2 Object 13#215: Actual value filter time constant

This parameter sets the time constant in seconds of the first order lag element (PT1).

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#215	VelActualFilterTimeConstant	0x5548	FLOAT32	rw	Y	0…+inf	0.0

7.10.2.3 Object 13#211: Actual value filter output

This parameter holds the output of the axis velocity actual filter.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#211	VelPosActualValueFilterOutput	0x5552	FLOAT32	ro	-	FLOAT32	None

7.10.2.4 Object 13#212: Actual derivative length

There is a special algorithm to calculate the derivation. The slope between two or up to twenty sampled position points is used for calculation to get the average over a longer period. A high <VelActualDerivativeLength> (13#212) will give a smoother average signal but also a delay to the actual signal.

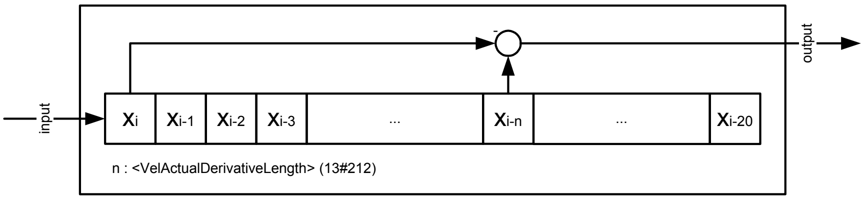


Figure 117: Actual derivative length

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#212	VelActualDerivativeLength	0x5550	UINT8	rw	Y	1…20	1

7.10.2.5 Object 13#210: Actual value path mode

This switch selects the path of the controller input for the actual velocity.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#210	VelActualPathMode	0x5545	UINT8	rw	Y	0…1	0

Value description

<VelActualPathMode>	**Description**
0	The source for the actual velocity is the velocity transducer input.
1	The source for the actual velocity is the derivation of the position transducer input.

Table 68: Possible values of parameter <VelActualPathMode> (13#210)

7.10.2.6 Object 13#213: Actual value proportional gain

The actual velocity or the derived actual position is multiplied with this gain to get the <VelActualValue> (13#100).

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#213	VelActualGain	0x5544	FLOAT32	rw	Y	FLOAT32	0.0

7.10.2.7 Object 13#100: Axis velocity actual value

The velocity controller needs the actual velocity <VelActualValue> (13#100) of the hydraulic axis. This actual velocity can be read in via the transducer interface from the encoder (if the encoder supports the velocity signal) or the velocity can be calculated as the derivation of the actual position signal <PosActualValue1> (12#100). The parameter <VelActualPathMode> (13#210) will switch between the two variants.

⇨ Chapter "7.10.2 Axis velocity actual value", page 199

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#100	VelActualValue	0x6501	INT32	ro	-	INT32	None

7.10.3 Axis velocity feed forward proportional element

To realize a feed forward control, the axis velocity demand value will be multiplied with this gain and added to the other control elements.

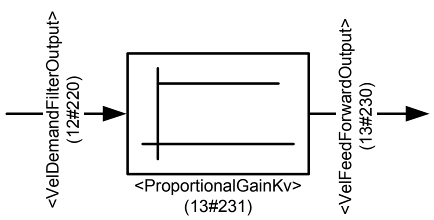


Figure 118: Axis velocity feed forward proportional element

7.10.3.1 Object 13#231: Velocity feed forward proportional gain

This parameter contains the velocity feed forward gain.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#231	ProportionalGainKv	0x5547	FLOAT32	rw	Y	FLOAT32	0.0

7.10.3.2 Object 13#230: Velocity feed forward output

This parameter holds the output of the axis velocity feed forward proportional element.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#230	VelFeedForwardOutput	0x5546	FLOAT32	ro	-	FLOAT32	None

7.10.3.3 Object 13#103: Control deviation

This parameter holds the deviation between the axis velocity demand filter value and the axis velocity actual value.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#103	VelDemandFilterOutput	0x6550	INT32	ro	-	INT32	None

7.10.4 Proportional element

The proportional part output is generated by the proportional gain parameter <ProportionalGainKp> (13#222). The input signal is the axis position control deviation.

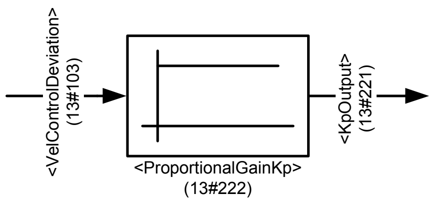


Figure 119: Proportional element

7.10.4.1 Object 13#222: Proportional gain

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#222	ProportionalGainKp	0x5514	FLOAT32	rw	Y	FLOAT32	0.0

7.10.4.2 Object 13#221: Kp output

This parameter holds the output of the proportional element of the axis velocity controller.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#221	KpOutput	0x5537	FLOAT32	ro	-	FLOAT32	None

7.10.5 Integral element

The integral element is composed of an input depending gain, the integrator and a limiter.

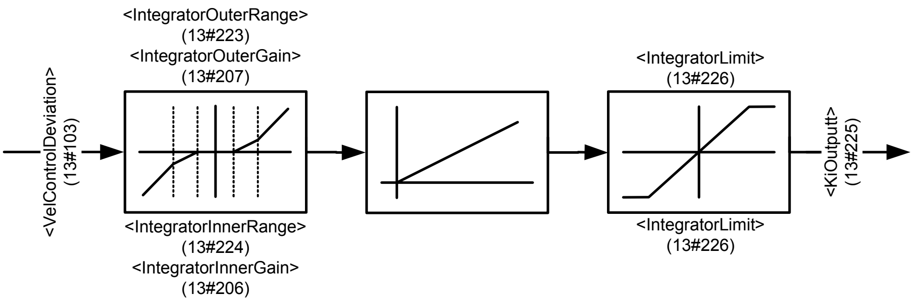


Figure 120: Integral element

7.10.5.1 Object 13#206…207: Integral gain

The gain of the integral part depends on the value of the absolute input signal |Input|.

Zero range: |Input| less than <IntegralInnerRange> (13#224). Output = Zero.
Inner range: |Input| between <IntegralInnerRange> (13#224) and <IntegralOuterRange> (13#223).

Output = <IntegralInnerGain> (13#206) x Input

Outer range: |Input| greater than <IntegralOuterRange> (13#223).

Output = <IntegralOuterGain> (13#207) x Input

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#206	IntegralInnerGain	0x5522#1	FLOAT32	rw	Y	FLOAT32	0.0
13#207	IntegralOuterGain	0x5522#2	FLOAT32	rw	Y	FLOAT32	0.0

7.10.5.2 Object 13#224: Integral inner range

Border of the absolute input signal to switch between gain = 0.0 and gain = <IntegralInnerGain> (13#206).

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
12#224	IntegralInnerRange	0x5517	FLOAT32	rw	Y	0.0…2147483647	0.0

7.10.5.3 Object 13#223: Integral outer range

Border of the absolute input signal to switch between gain = <IntegralInnerGain> (13#206) and gain = <IntegralOuterGain> (13#207).

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#223	IntegralOuterRange	0x5516	FLOAT32	rw	Y	0.0…2147483647	0.0

7.10.5.4 Object 13#236: Integral limit

Limit of the output signal of the integral element.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#236	IntegralLimit	0x5518	FLOAT32	rw	Y	0…+inf	16384000000

7.10.5.5 Object 13#225: Ki output

This parameter holds the output of the integral element of the axis velocity controller.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#225	KiOutput	0x553C	FLOAT32	ro	-	FLOAT32	None

7.10.6 Acceleration feedback

7.10.6.1 Feedback axis velocity value filter

This first order lag element (PT1) is used to filter the axis velocity actual value.
⇨ Chapter "7.10.2.2 Object 13#215: Actual value filter time constant", page 199

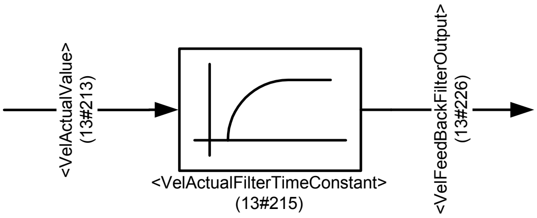


Figure 121: Feedback axis velocity filter

7.10.6.2 Object 13#226: Velocity feedback filter output

This is the filtered output of the axis velocity actual value.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#226	VelFeedBackFilterOutput	0x5549	FLOAT32	ro	-	FLOAT32	None

7.10.6.3 Feedback axis velocity derivative element

Calculation of the axis acceleration actual value by derivation of the axis velocity actual value.
⇨ Chapter "7.10.2.4 Object 13#212: Actual derivative length", page 200

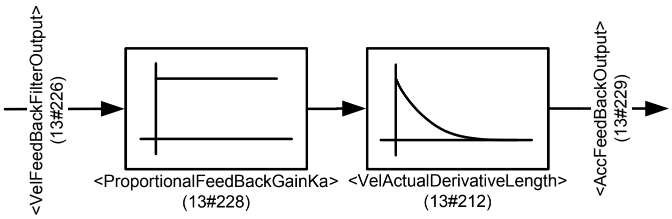


Figure 122: Feedback axis velocity derivative element

7.10.6.4 Object 13#228: Feedback axis velocity derivative gain

Gain for the acceleration feedback.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#228	ProportionalFeedBackGainKa	0x5329	FLOAT32	rw	Y	FLOAT32	0.0

7.10.6.5 Object 13#229: Acceleration feedback output

Output of the acceleration feedback to the controller.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#229	AccFeedBackOutput	0x5539	FLOAT32	ro	-	FLOAT32	None

7.10.7 Directional depending gain

To compensate a not symmetrical movement between moving in and out a directional depending gain is implemented.

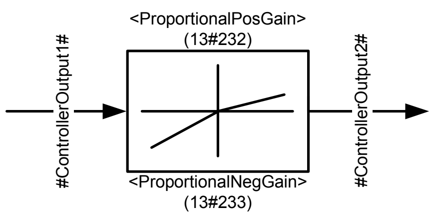


Figure 123: Directional depending gain

7.10.7.1 Object 13#232: Directional depending positive proportional gain

This gain is used while the input signal is positive.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#232	ProportionalPosGain	0x5534	FLOAT32	rw	Y	FLOAT32	1.0

7.10.7.2 Object 13#233: Directional depending negative proportional gain

This gain is used while the input signal is positive.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#233	ProportionalNegGain	0x5535	FLOAT32	rw	Y	FLOAT32	1.0

7.10.8 Signal limitation

The output of the controller is limited to ±16384 (±100 %).

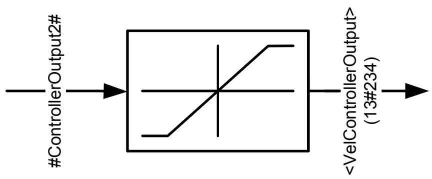


Figure 124: Signal limitation

7.10.8.1 Object 13#234: Controller output

This parameter holds the controller output of the axis velocity controller.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#234	VelControllerOutput	0x5519	FLOAT32	ro	-	FLOAT32	None

7.10.9 Sample frequency

The sample frequency of the axis position controllers is the <BasicSampleFrequency> (0x3030) divided by the <VelSampleFrequencyDivider> (13#214).

➪ Chapter "7.1.13.1 Object 0x3030: Basic sample frequency", page 127

7.10.9.1 Object 13#214: Sample frequency divider

This parameter holds the sample frequency divider.

DriveVelocityControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
13#214	VelSampleFrequencyDivider	0x5540	UINT8	rw	Y	1…255	2

7.11 Flow setpoint conditioning / demand value generator

The flow controller does not have a demand value generator.

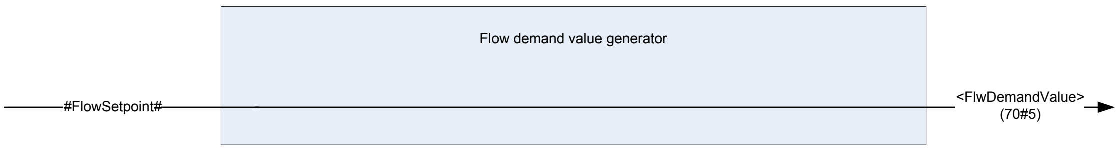


Figure 125: Flow setpoint conditioning / demand value generator

7.12 Flow control

The flow control mode is used to control the flow (and so the cylinder velocity) independent to the load (pressure) on the cylinder. To reach this target, the non-linear influence of the pressure in the cylinder cambers is compensated with the so called Bernoulli compensator.

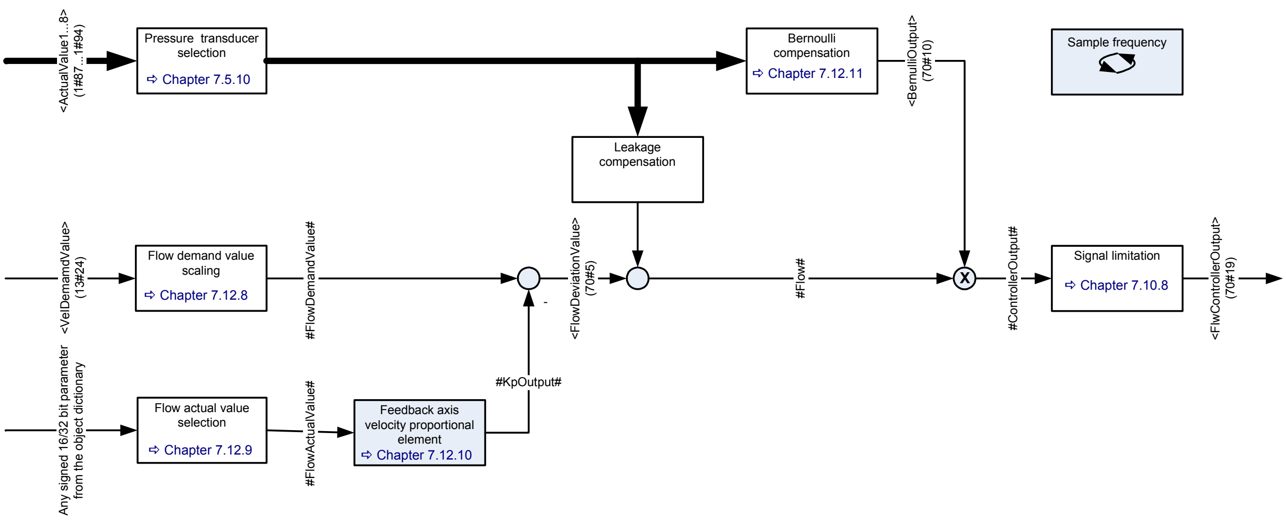


Figure 126: Flow control

⇨ Chapter "7.10.2 Axis velocity actual value", page 199

7.12.1 Object 70#5: Deviation value

This parameter holds the deviation between the flow demand filter value and the flow feedback value.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#5	FlwDeviationValue	0x5213	INT16	ro	-	INT16	None

7.12.2 Object 70#10: Bernoulli output

This parameter holds the Bernoulli output of the flow position controller.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#5	BernoulliOutput	0x5200	INT16	ro	-	INT16	None

7.12.3 Object 70#19: Controller output

This parameter holds the controller output of the flow position controller.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#19	FlwControllerOutput	0x520F	INT16	ro	-	INT16	None

7.12.4 Sample frequency

The sample frequency is equal to the <BasicSampleFrequency> (no access via Profibus).
⇨ Chapter "7.1.13.1 Object 0x3030: Basic sample frequency", page 127

7.12.5 Flow control mode

The <FlwControlMode> (70#16) is used to select the connection variants of the existing pressure sensors.

Edge for y>0	Edge for y<0	Pressure sensor at cylinder connector A	Pressure sensor at cylinder connector B	Flow control mode <FlwControlMode> (70#16)
P-A	P-B	x	x	0
P-A	A-T	x		1
B-t	P-B		x	Not implemented

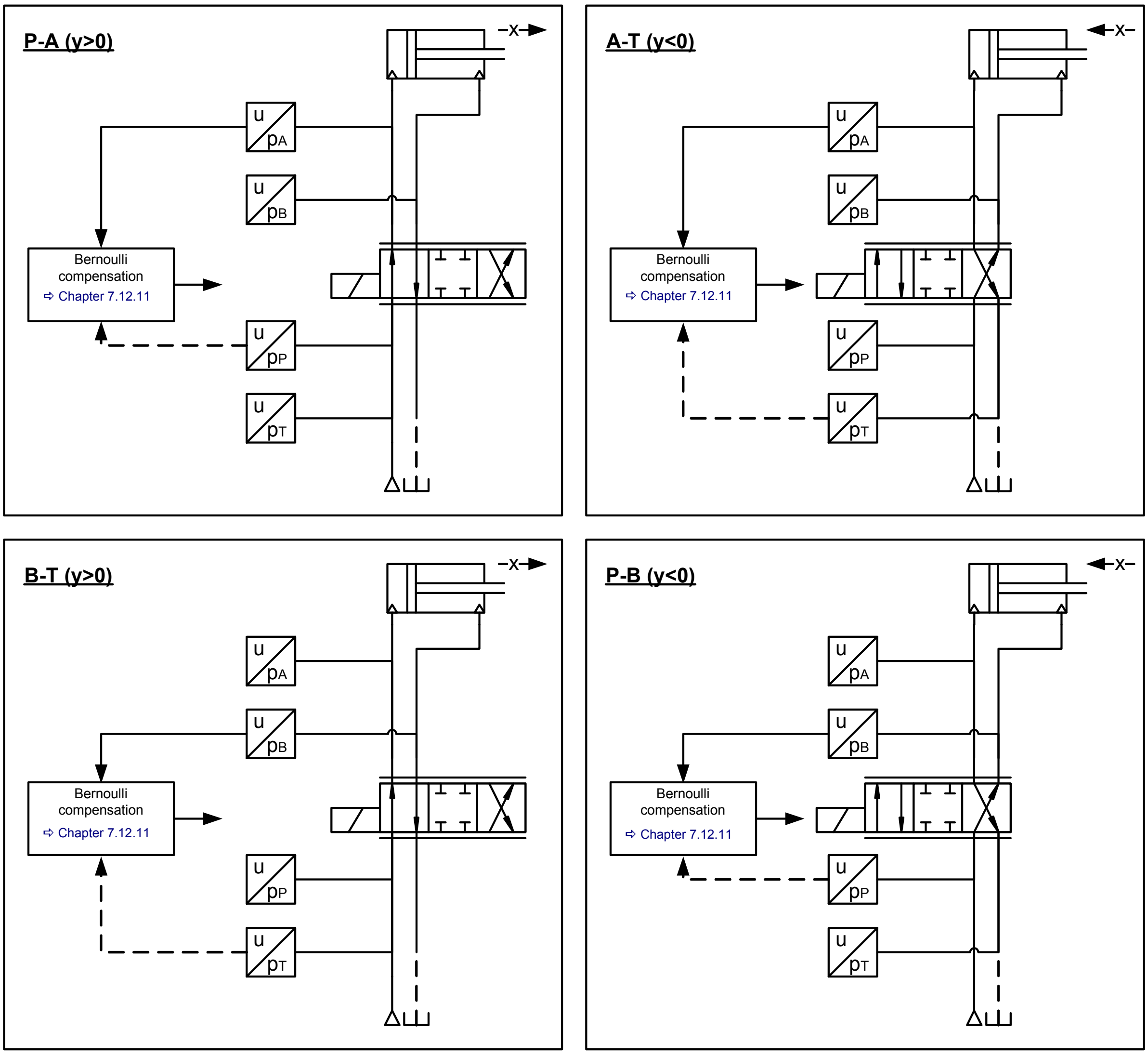


Figure 127: Flow control mode

7.12.5.1 Object 70#16: Flow control mode

The flow control mode selects the pressure sensor configuration.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#16	FlwControlMode	0x5205	UINT8	rw	Y	0…1	0

Value description

<FlwControlMode>	**Description**
0	P-A, P-B edge
1	P-A, A-T edge

Table 69: Possible values of parameter <FlwControlMode> (70#16)

7.12.6 Flow transducer selection

The pressure actual values are read and scaled via the transducer interfaces. The parameter <ActiveTransducerInterfaceAreaA> (66#17…32) keeps the number of the interface for the pressure in cylinder A-side. If the pressure on A-side is rising, the cylinder has to move in positive direction.

The actual pressure on A-side is necessary for the flow control. The pressure in cylinder B-side is optional as well as the system pressure and the tank pressure (corresponding interface numbers: <ActiveTransducerInterfaceAreaB> (66#33…48), <ActiveTransducerInterfaceSystem> (70#1) and <ActiveTransducerInterfaceTank> (70#24). All actual pressure values are filtered with a low pass filter.

➪ Chapter "7.5.10.1 Object 66#17…32: Active transducer interface area A", page 163
➪ Chapter "7.5.10.2 Object 66#33…48: Active transducer interface area B", page 164

➪ Chapter "7.5.11.1 Object 67#1: Actual pressure filter cutoff frequency", page 165
➪ Chapter "7.5.11.2 Object 67#2: Actual pressure filter order", page 165

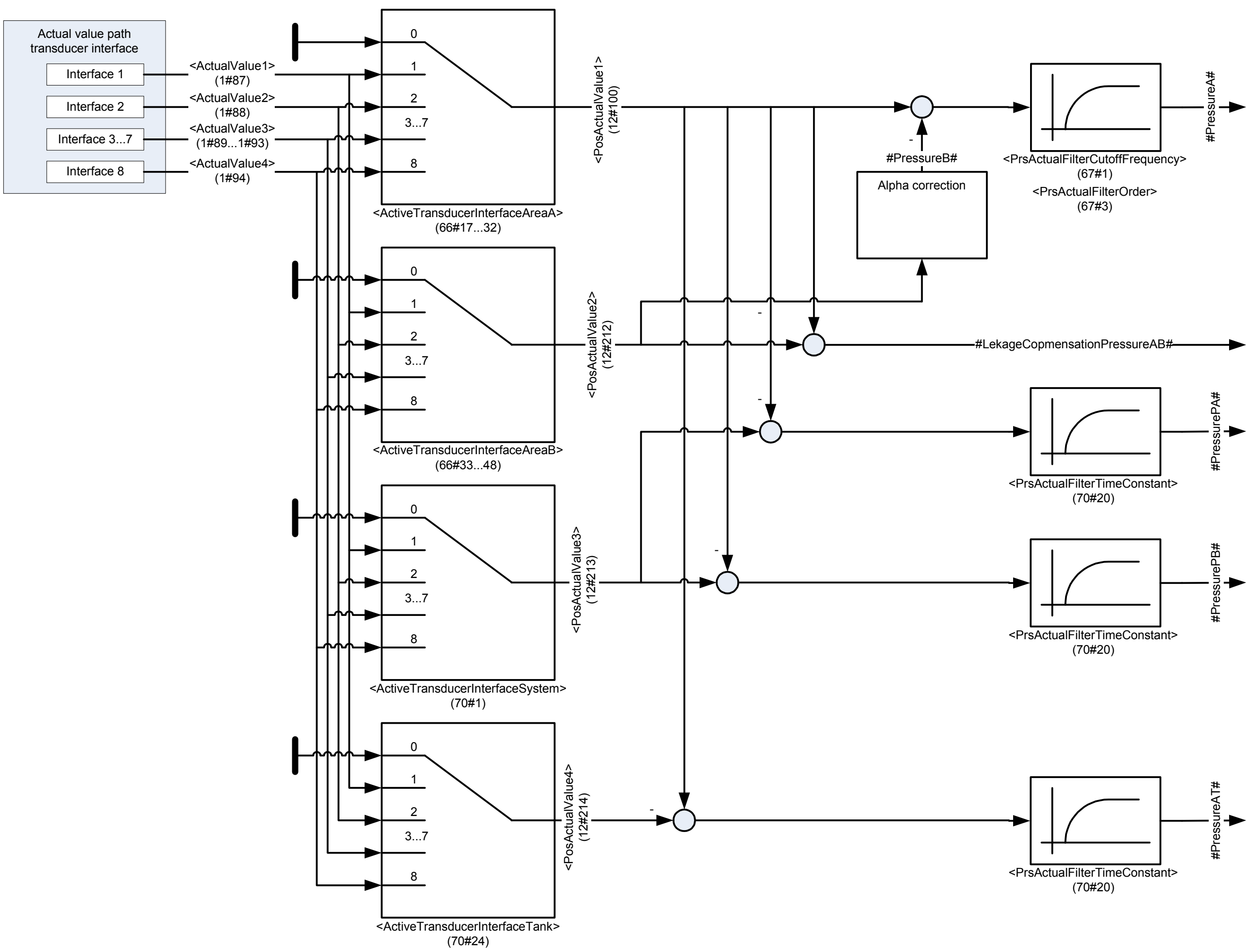


Figure 128: Flow transducer selection

7.12.6.1 Object 70#1: Active transducer interface system

This parameter keeps the number of the transducer interface for the actual system pressure value. Instead of using a pressure sensor it is also possible to use a constant value.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#1	ActiveTransducerInterfaceSystem	0x2330	INT8	rw	Y	0…8	0

7.12.6.2 Object 70#24: Active transducer interface tank

This parameter keeps the number of the transducer interface for the actual tank pressure value. Instead of using a pressure sensor, it is also possible to use a constant value.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#24	ActiveTransducerInterfaceTank	0x5217	INT8	rw	Y	0…8	0

7.12.7 Alpha correction

The alpha correction can be used to compensate the area differences between A-side and B-side of a differential cylinder.

⇨ Chapter "7.5.10.3 Object 66#248: Cylinder piston diameter", page 164
⇨ Chapter "7.5.10.4 Object 66#249: Cylinder rod diameter A", page 164
⇨ Chapter "7.5.10.5 Object 66#250: Cylinder rod diameter B", page 164

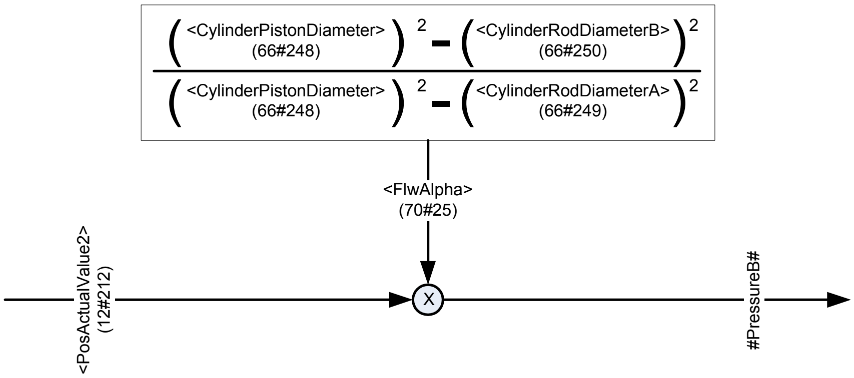


Figure 129: Alpha correction

7.12.7.1 Object 70#25: Flow alpha

This parameter is the multiplier for the pressure in B-side to get the alpha corrected actual pressure.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#25	FlwAlpha	0x5219	FLOAT32	ro	-	FLOAT32	None

7.12.7.2 Object 70#20: Actual value filter time constant

This parameter sets the time constant in seconds of the first order lag element (PT1). The same time constant is used for the actual pressure in B-side, system and tank.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#20	PrsActualFilterTimeConstant	0x5210	FLOAT32	rw	Y	0.0…+inf	0.0

7.12.8 Flow demand value scaling

The flow demand value can be reduced, increased or negated with this scaling factor.

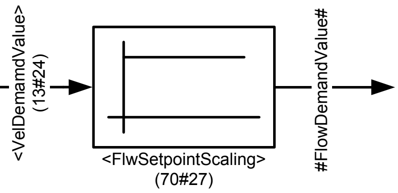


Figure 130: Flow demand value scaling

7.12.8.1 Object 70#27: Demand value scaling

This parameter sets the demand value scaling.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#27	FlwSetpointScaling	0x521B	FLOAT32	rw	Y	–1.0…1.0	1.0

7.12.9 Flow actual value selection

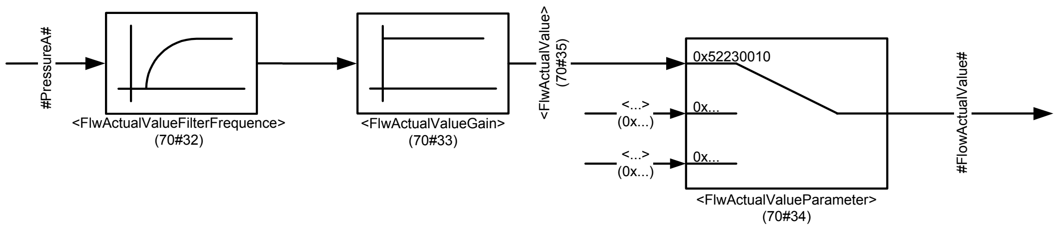


Figure 131: Flow actual value selection

7.12.9.1 Object 70#35: Actual value

This parameter sets the demand value scaling.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#35	FlwActualValue	0x5223	INT16	ro	-	INT16	None

7.12.9.2 Object 70#32: Actual value high pass filter frequency

This parameter holds the cut off frequency in Hertz of the high pass filter.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#32	FlwActualValueFilterFrequence	0x5220	FLOAT32	rw	Y	FLOAT32	20.0

7.12.9.3 Object 70#33: Actual value proportional gain

This parameter holds the proportional gain for the flow actual value.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#33	FlwActualValueGain	0x5221	FLOAT32	rw	Y	FLOAT32	1.0

7.12.9.4 Object 70#34: Actual value parameter

By default this parameter is pointing to <FlwActualValue> (70#35) as feedback input for the axis velocity controller. But also each other signal can be mapped to the feedback input.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#34	FlwActualValueParameter	0x5222	FLOAT32	rw	Y	FLOAT32	0x52230010

Value description

<FlwActualValueParameter>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length:0x10
Default	0x52	0x23	0x01	0x10

Table 70: Possible values of parameter <FlwActualValueParameter> (70#34)

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

The default value is 0x52230110, which refers to the <FlwActualValue> (70#35), sub-index 0x01 with a length of 16 bit (16 = 0x10).

7.12.10 Feedback axis velocity proportional element

This element represents a feedback proportional gain for the velocity actual value which is proportional to the flow actual value.

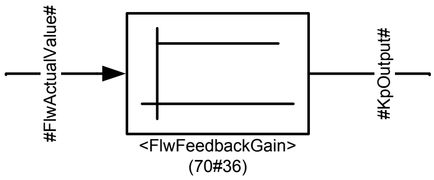


Figure 132: Feedback axis velocity proportional element

7.12.10.1 Object 70#36: Feedback axis velocity proportional gain

This parameter holds the proportional gain for the flow controller.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#36	FlwFeedbackGain	0x5224	FLOAT32	rw	Y	FLOAT32	0.0

7.12.11 Bernoulli compensation

The Bernoulli compensator compensates the influence of the load on the hydraulic cylinder.

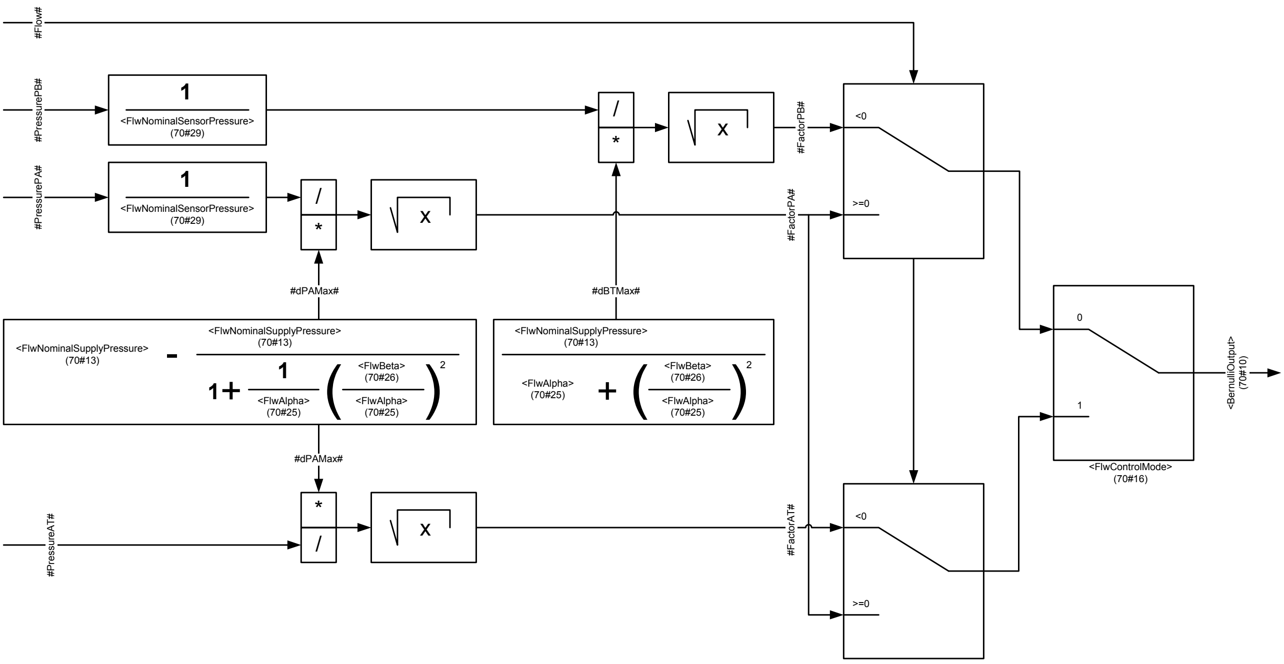


Figure 133: Bernoulli compensator

7.12.11.1 Object 70#22: Nominal flow A side

Maximal flow on port A of the servo valve.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#22	FlwNominalFlowASide	0x5215	FLOAT32	rw	Y	0.01…+inf	0.0

7.12.11.2 Object 70#23: Nominal flow B side

Maximal flow on port B of the servo valve.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#23	FlwNominalFlowBSide	0x5216	FLOAT32	rw	Y	0.01…+inf	0.0

7.12.11.3 Object 70#26: Flow beta

This is a calculated servo valve depending factor. It equals the <FlwNominalFlowBSide> (70#23) divided by <FlwNominalFlowASide> (70#22) (at the same system pressure).

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#26	FlwBeta	0x521A	FLOAT32	ro	-	FLOAT32	0.0

7.12.11.4 Object 70#13: Nominal supply pressure

The nominal supply pressure of the servo valve is necessary to normalize the command value.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#13	FlwNominalSupplyPressure	0x5203	INT16	rw	Y	1…32767	400

7.12.11.5 Object 70#29: Nominal sensor pressure

The nominal sensor pressure displays the reference pressure of the transducer interface. It is necessary to normalize the pressure difference P->B and A->T.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#29	FlwNominalSensorPressure	0x521D	INT16	ro	-	INT16	None

7.12.11.6 Object 70#31: Maximal flow A side

For the flow control the maximal flow on A-side depending on the nominal pressure on A-side is necessary.

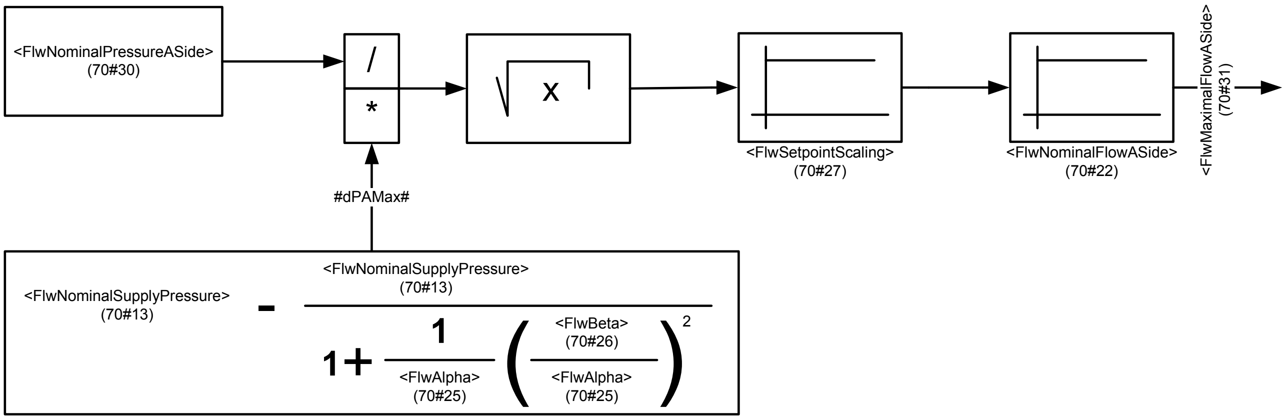


Figure 134: Maximal flow A side

This parameter contains the result of the calculation of the maximal flow.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#31	FlwMaximalFlowASide	0x521F	FLOAT32	ro	-	FLOAT32	None

7.12.11.7 Object 70#30: Nominal pressure A side

For the calculation of <FlwMaximalFlowASide> (70#31), the nominal pressure is necessary (the pressure used to measure the nominal flow <FlwNominalFlowASide> (70#22)).

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#30	FlwNominalPressureASide	0x521E	FLOAT32	ro	-	FLOAT32	None

7.13 Flow / pressure (P) switchover

The switch-over from flow (velocity) control to pressure (force) control allows a limit for the maximal force on the piston rod. The valve will switch to pressure control if the pressure actual value is bigger than the demand pressure. It will switch back to flow control if the output of the pressure controller is bigger than the output of the flow controller (absolute values). The demand signal to the pilot valve will change continuous (filtered with PT1 elements).

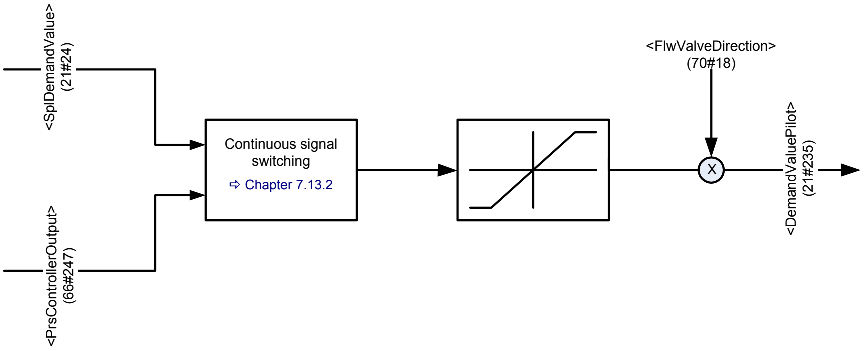


Figure 135: Flow / pressure switchover

7.13.1 Object 70#18: Flow valve direction

Signal to negate the output to the valve. Thus, the direction of the axis movement can be inverted.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#18	FlwValveDirection	0x520E	INT8	rw	Y	–1…1	1

7.13.2 Continuous signal switching

The switch-over criteria between the control modes are the PressureDemandValue (<PrsDemandValue> (22#24)), the PressureActualValue (<PrsActualValue> (22#144)), the output of the PressureControllerOutput (<ControllerOutput> (66#247)) and the output of the FlowControllerOutput (<ControllerOutput> (70#19)). Both controllers are calculated then the switch over function will decide which controller output is used as input for the pilot spool position controller.

Criterion to switch from flow control to pressure control:

- Positive PressureDemandValue: PressureActualValue > PressureDemandValue
  Or
- Negative PressureDemandValue: PressureActualValue < PressureDemandValue

Only if the criterion for the switch over to pressure control is not fulfilled, the criterion for the switch over to flow control is checked:

- Positive PressureDemandValue: PressureControllerOutput > FlowControllerOutput
  Or
- Or negative PressureDemandValue: PressureControllerOutput < FlowControllerOutput
  Or
- Or sign of the FlowControllerOutput has changed.

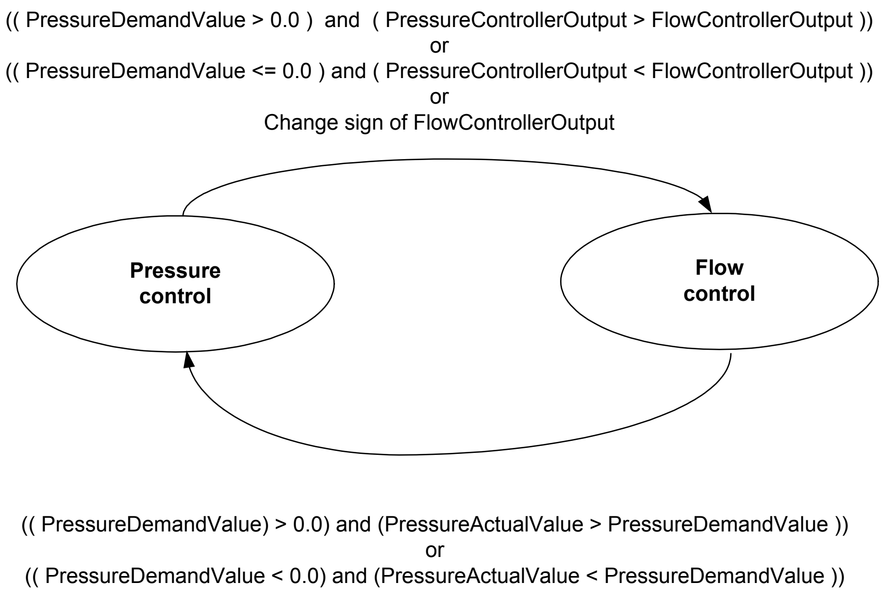


Figure 136: Continuous signal switching

7.13.2.1 Object 70#14…15: Switching time constant

Time constants for the PT1 filter elements to get a continuous input signal to the pilot position controller if the control mode is switching from PressureControllerOutput (<ControllerOutput> (66#247)) to FlowControllerOutput (<ControllerOutput> (70#19)) and back.

ValveFlowControl							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
70#14	SwitchingTimeConstantFlow2Pressure	0x5204#1	FLOAT32	rw	Y	0.0…+inf	30.0
70#15	SwitchingTimeConstantPressure2Flow	0x5204#2	FLOAT32	rw	Y	0.0…+inf	3.0

7.14 Axis status

7.14.1 Object 69#13: Axis status word

Drive							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
69#13	AxisStatusWord	0x561D	UINT16	ro	-	UINT16	None

Value description

<AxisStatusWord>		
	Name	**Description**
Bit 0 0x0001	Axis is referenced	Reset at begin of reference run. Set after reference run has finished.
Bit 1 0x0002	Position window 1 reached and stopped	Active while the position difference (|<DemandValue> (12#24) – <ActualValue1> (12#100)|) is less than <PositionWindow1> (12#247) and the velocity |<ActualValue> (13#100)| is less than <VelocityLimitWindow> (13#209).
Bit 2 0x0004	Velocity window reached	Active while the velocity difference (|<ActualValue> (13#100) – <VelocityLimit> (13#208)|) is less than <VelocityLimitWindow> (13#209).
Bit 3 0x0008	Velocity stopped	Active while the actual velocity (|<ActualValue> (13#100) | is less than <VelocityLimitWindow> (13#209).
Bit 4 0x0010	Force reached	Active while the pressure difference (|<prsActualvalue> (22#144) – <PrsDemandValue> (22#24)|) is less than <PressureWindow> (69#11).
Bit 5 0x0020	Position window 2 reached	Active while the position difference (|<DemandValue> (12#24) – <ActualValue1> (12#100) |) is less than <PositionWindow2> (12#248).
Bit 6 0x0040	Maximum force reached	Active while the actual pressure <prsActualvalue> (22#144) is greater than the <MaximumPressureWindow> (69#12).
Bit 7 0x0080	Axis limit touched	Not implemented.
Bit 8 0x0100	Install or reference mode active	Active while the <DeviceMode> (0#39) = 3 (install mode) or = 4 (reference mode).
Bit 9 0x0200	Reference search end of stroke	Active while the end of stroke is searched in reference run.
Bit 10 0x0400	Reference search Z-pulse	Active while the Z-pulse is searched in reference run.

Table 71: Possible values of parameter <AxisStatusWord> (69#13)

7.14.2 Object 69#4: Axis status and device status word

This 32 bit status word is a combination of the 16 bit <AxisStatusWord> (69#13) and the 16 bit <StatusWord> (0#38).

⇨ Chapter "5.2.3 Object 0#38: Status word", page 48

Drive							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
69#4	AxisStatusAndDeviceStatusWord	0x561C	UINT32	ro	-	UINT32	None

Value description

<AxisStatusAndDeviceStatusWord>		
Bit	**Word**	**Description**
0…15	<AxisStatusWord> (69#13)	Axis status word.
16…31	<StatusWord> (0#38)	Device status word.

Table 72: Possible values of parameter <AxisStatusAndDeviceStatusWord> (69#4)

7.14.3 Object 12#247…248: Axis position deviation window

The position deviation window is used to set bit 1 and bit 5 in the <AxisStatusWord> (69#13).

Bit 1 is active while the absolute position difference (|<DemandValue> (12#24) – <ActualValue1> (12#100)|) is less than <PositionWindow1> (12#247) and the velocity |<ActualValue> (13#100)| is less than <VelocityLimitWindow> (13#209).

Bit 5 is active while the absolute position difference (|<DemandValue> (12#24) – <ActualValue1> (12#100)|) is less than <PositionWindow2> (12#247).

Drive							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
12#247	PositionWindow1	0x570C#1	INT32	rw	Y	0…2147483647	0
12#248	PositionWindow2	0x570C#2	INT32	rw	Y	0…2147483647	0

7.14.4 Object 13#209: Axis velocity actual value window

The velocity actual value window is used to set bit 1 and bit 3 in the <AxisStatusWord> (69#13).

Bit 1 is active while the absolute position difference (|<DemandValue> (12#24) – <ActualValue1> (12#100)|) is less than <PositionWindow1> (12#247) and the velocity |<ActualValue> (13#100)| is less than <VelocityLimitWindow> (13#209).

Bit 3 is active while the absolute velocity |<ActualValue> (13#100)| is less than the <VelocityLimitWindow> (13#209).

Drive							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
13#209	VelocityLimitWindow	0x570D	INT32	rw	Y	0…2147483647	0

7.14.5 Object 13#208: Axis velocity deviation window

The velocity deviation window is used to set bit 2 in the <AxisStatusWord> (69#13).

Bit 2 is active while the velocity difference (|<ActualValue> (13#100) – <VelocityLimit> (13#208|) is less than the <VelocityLimitWindow> (13#209).

Drive							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
13#208	VelocityLimit	0x570E	INT32	rw	Y	INT32	0

7.14.6 Object 69#11: Pressure deviation window

The pressure deviation window is used to set bit 4 in the <AxisStatusWord> (69#13).

Bit 4 is active while the absolute pressure difference (|<prsActualvalue> (22#144) – <PrsDemandValue> (22#24)|) is less than the <PressureWindow> (69#11).

Drive							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
69#11	PressureWindow	0x5710	INT16	rw	Y	0…32767	0

7.14.7 Object 69#12: Pressure actual value limit

The Pressure limit window used to set bit 6 in the <AxisStatusWord> (69#13).

Bit 6 is active while the actual pressure <prsActualvalue> (22#144) is greater than the <MaximumPressureWindow> (69#12).

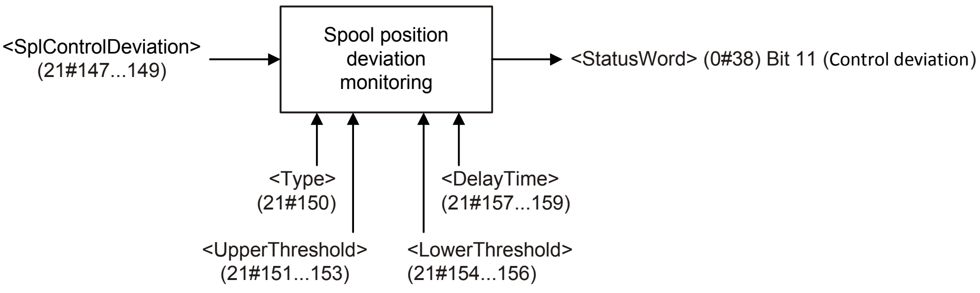
Drive							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sistence	Value range	Default
69#12	MaximumPressureWindow	0x5711	INT16	rw	Y	0…32767	0

7.15 Monitoring

The control deviation monitoring is only active if the associated controller is active. In p/Q mode, the <ControlMode> (0#40) is set to 5 (p/Q controller), the effective deviation monitoring depends on the active controller indicated by the bit 8 (pressure controller effective) of the <StatusWord> (0#38).

⇨ Chapter "7.1.1 Object 0#40: Control mode", page 119
⇨ Chapter "5.2.3 Object 0#38: Status word", page 48

7.15.1 Spool position control deviation monitoring



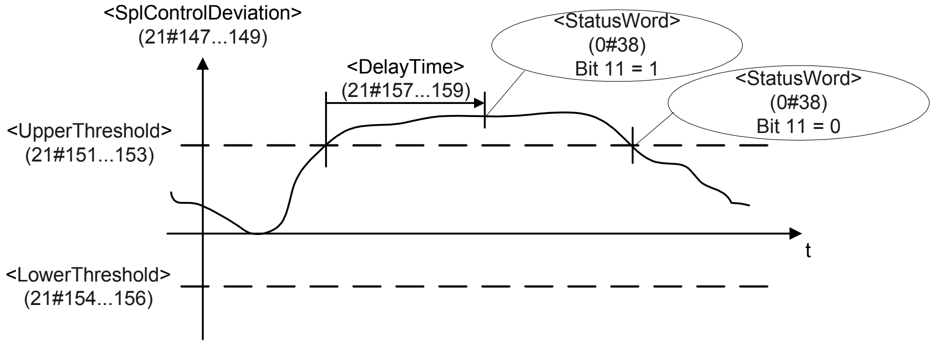


Figure 137: Spool position control deviation monitoring

➩ Chapter "7.3.5.1 Object 21#147…149: Control deviation", page 147

7.15.1.1 Object 21#150: Type

The parameter <Type> (21#150) is used to activate or deactivate the standard spool position control deviation monitoring function.

ValvePositionControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#150	Type	0x6351#0	INT8	rw	Y	0…1	0

Value description

<Type>	**Description**
0	Spool position control deviation monitoring off.
1	Spool position control deviation monitoring on.

Table 73: Possible values of parameter <Type> (21#50)

7.15.1.2 Object 21#157…159: Delay time

The delay time defines the minimal duration of a control deviation before a fault is active. The time is set to zero if the position is inside the window, if the fault is acknowledged or if one of the following parameters is changed: <Type> (21#150), <UpperThreshold> (21#151), <LowerThreshold> (21#154) or <DelayTime> (21#157).

ValvePositionControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#157	DelayTime	0x6352#1	UINT16	rw	Y	UINT16	30
21#158	Unit	0x6352#2	UINT8	ro	-	UINT8	3
21#159	Prefix	0x6352#3	INT8	ro	-	INT8	–3

➪ Chapter "2.5.3 Units and prefix parameter", page 13

7.15.1.3 Object 21#151…153: Upper threshold

This parameter defines the upper threshold of the control deviation's tolerance band.

ValvePositionControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#151	UpperThreshold	0x6354#1	INT16	rw	Y	INT16	512
21#152	Unit	0x6354#2	UINT8	ro	-	UINT8	0
21#153	Prefix	0x6354#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

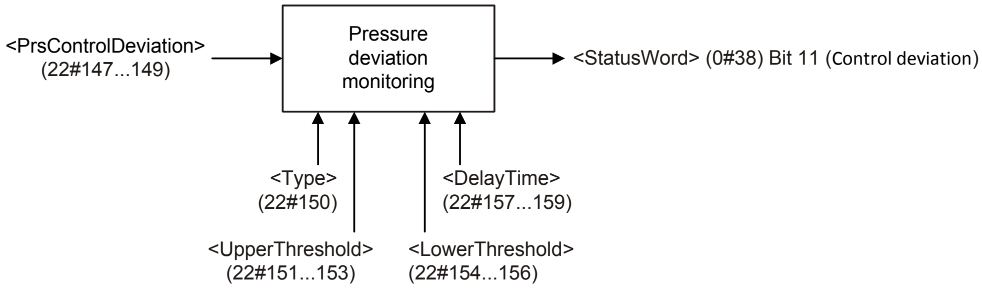
7.15.1.4 Object 21#154…156: Lower threshold

This parameter defines the lower threshold of the control deviation's tolerance band.

ValvePositionControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
21#154	LowerThreshold	0x6355#1	INT16	rw	Y	INT16	–512
21#155	Unit	0x6355#2	UINT8	ro	-	UINT8	0
21#156	Prefix	0x6355#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

7.15.2 Pressure control deviation monitoring



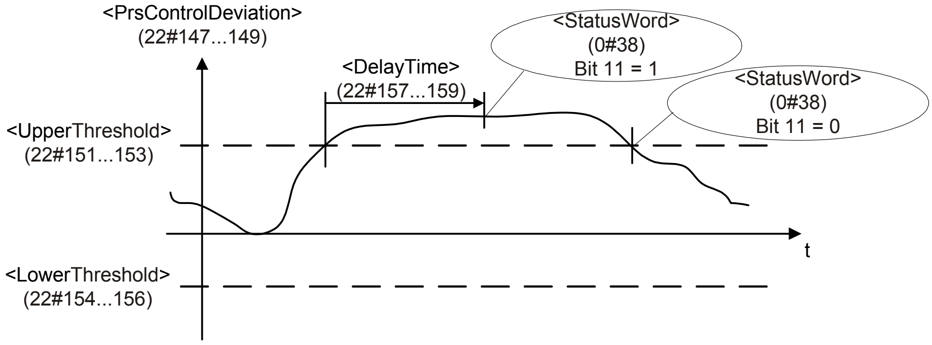


Figure 138: Pressure control deviation monitoring

➪ Chapter "7.5.2 Object 22#147…149: Control deviation", page 159

7.15.2.1 Object 22#150: Type

The parameter <Type> (22#150) is used to activate or deactivate the standard pressure control deviation monitoring function.

ValvePressureControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#150	Type	0x63D1#0	INT8	rw	Y	0…1	0

Value description

<Type>	**Description**
0	Pressure control deviation monitoring off.
1	Pressure control deviation monitoring on.

Table 74: Possible values of parameter <Type> (22#150)

7.15.2.2 Object 22#157…159: Delay time

The delay time defines the minimal duration of a control deviation before a fault is active. The Time is set to zero if the position is inside the window, if the fault is acknowledged or if one of the following parameters is changed: <Type> (22#150), <UpperThreshold> (22#151), <LowerThreshold> (22#154) or <DelayTime> (22#157).

ValvePositionControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#157	DelayTime	0x63D2#1	UINT16	rw	Y	UINT16	30
22#158	Unit	0x63D2#2	UINT8	ro	-	UINT8	3
22#159	Prefix	0x63D2#3	INT8	ro	-	INT8	–3

➪ Chapter "2.5.3 Units and prefix parameter", page 13

7.15.2.3 Object 22#151…153: Upper threshold

This parameter defines the upper threshold of the control deviation's tolerance band.

ValvePositionControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#151	UpperThreshold	0x63D4#1	INT16	rw	Y	INT16	512
22#152	Unit	0x63D4#2	UINT8	ro	-	UINT8	0
22#153	Prefix	0x63D4#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

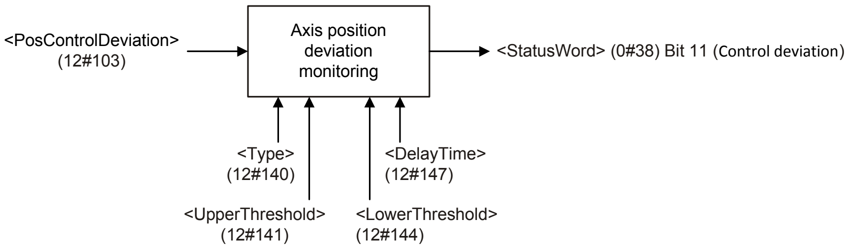
7.15.2.4 Object 22#154…156: Lower threshold

This parameter defines the lower threshold of the control deviation's tolerance band.

ValvePositionControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
22#154	LowerThreshold	0x63D5#1	INT16	rw	Y	INT16	–512
22#155	Unit	0x63D5#2	UINT8	ro	-	UINT8	0
22#156	Prefix	0x63D5#3	INT8	ro	-	INT8	0

➪ Chapter "2.5.3 Units and prefix parameter", page 13

7.15.3 Axis position control deviation monitoring



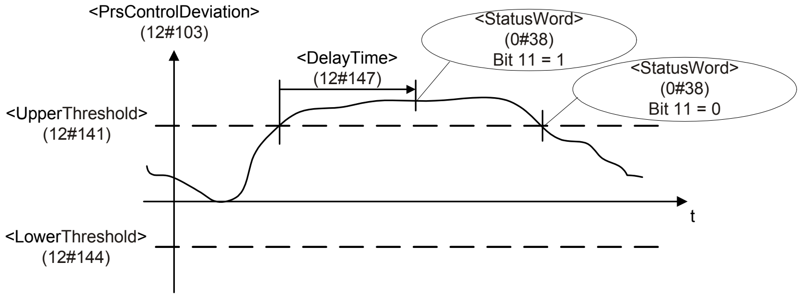


Figure 139: Axis position control deviation monitoring

⇨ Chapter "7.8.3 Object 12#103: Control deviation", page 185

7.15.3.1 Object 12#140: Type

This parameter is used to activate or deactivate the standard pressure control deviation monitoring function.

DrivePositionControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#140	Type	0x6651	INT8	rw	Y	0…1	0

Value description

<Type>	**Description**
0	Axis position control deviation monitoring off.
1	Axis position control deviation monitoring on.

Table 75: Possible values of parameter <Type> (12#140)

7.15.3.2 Object 12#147: Delay time

The delay time defines the minimal duration of a control deviation before a fault is active. The time is set to zero if the position is inside the window, if the fault is acknowledged or if one of the following parameters is changed:

- <Type> (12#140)
- <UpperThreshold> (12#140)
- <LowerThreshold> (12#144)
- <DelayTime> (12#140)

DrivePositionControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#147	DelayTime	0x6652	UINT16	rw	Y	UINT16	30

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.15.3.3 Object 12#141: Upper threshold

This parameter defines the upper threshold of the control deviation's tolerance band.

DrivePositionControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#141	UpperThreshold	0x6654	INT16	rw	Y	INT16	512

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

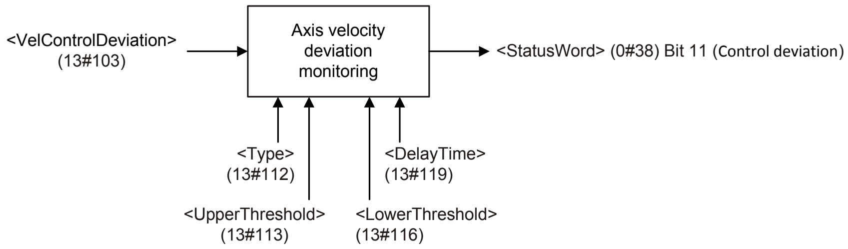
7.15.3.4 Object 12#144: Lower threshold

This parameter defines the lower threshold of the control deviation's tolerance band.

DrivePositionControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#144	LowerThreshold	0x6655	INT16	rw	Y	INT16	–512

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.15.4 Axis velocity control deviation monitoring



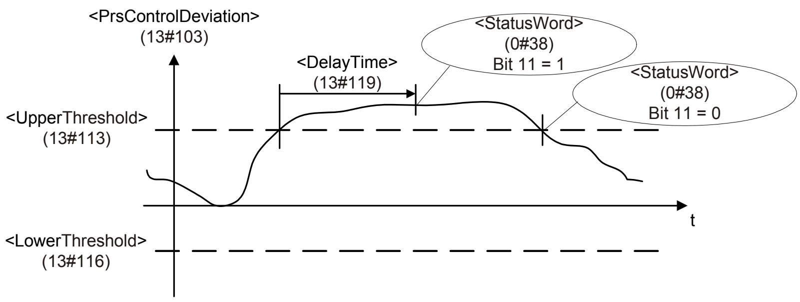


Figure 140: Axis velocity control deviation monitoring

⇨ Chapter "7.8.3 Object 12#103: Control deviation", page 185

7.15.4.1 Object 13#112: Type

This parameter is used to activate or deactivate the standard pressure control deviation monitoring function.

DriveVelocityControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#112	Type	0x6551	INT8	rw	Y	0…1	0

Value description

<Type>	**Description**
0	Axis position control deviation monitoring off.
1	Axis position control deviation monitoring on.

Table 76: Possible values of parameter <Type> (13#112)

7.15.4.2 Object 13#119: Delay time

The delay time defines the minimal duration of a control deviation before a fault is active. The time is set to zero if the position is inside the window, if the fault is acknowledged or if one of the following parameters is changed:

- <Type> (13#112)
- <UpperThreshold> (13#113)
- <LowerThreshold> (13#116)
- <DelayTime> (13#119)

DriveVelocityControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#119	DelayTime	0x6552	UINT16	rw	Y	UINT16	30

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.15.4.3 Object 13#113: Upper threshold

This parameter defines the upper threshold of the control deviation's tolerance band.

DriveVelocityControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#113	UpperThreshold	0x6554	INT16	rw	Y	INT16	512

⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.15.4.4 Object 13#116: Lower threshold

This parameter defines the lower threshold of the control deviation's tolerance band.

DriveVelocityControl_ControlMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#116	LowerThreshold	0x6555	INT16	rw	Y	INT16	–512

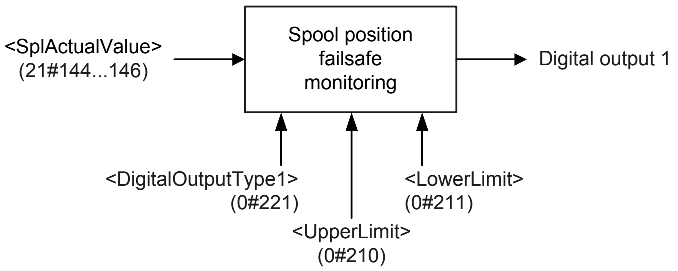
⇨ Chapter "2.5.3 Units and prefix parameter", page 13

7.15.5 Failsafe monitoring

This monitoring is used to monitor the so called "failsafe position" that means the spring centered spool position. The control window is defined by an <UpperLimit> (0#210) and a <LowerLimit> (0#211), which are set by the factory. If the failsafe spool position monitoring function is enabled by setting the parameter <DigitalOutputType1> (0#221) to 1 (failsafe spool position monitoring on), the servo valve monitors the failsafe position of the spool position <ActualValue> (21#144…146) and sets the digital output 1.

⇨ Chapter "6.8.2 Object 0#220…221: Digital output configuration", page 94

WARNING
Moving machine parts! The word "failsafe" does not mean personnel safe hydraulic motion control system. ▶ If personnel safety needs to be guaranteed, additional electrical and hydraulic components are necessary!



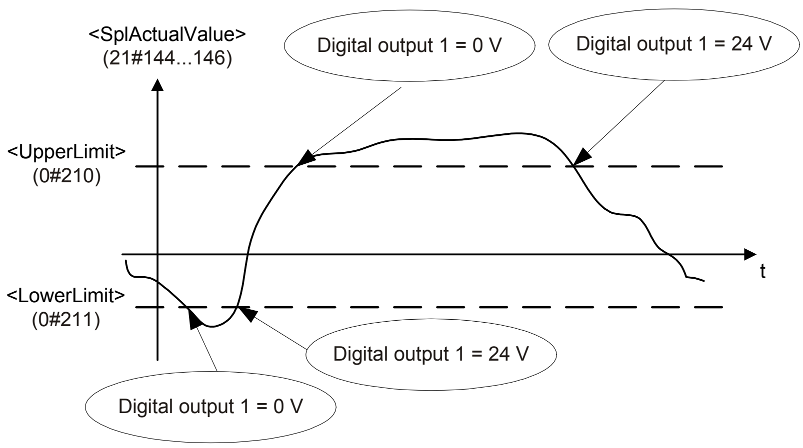


Figure 141: Failsafe monitoring

Electrical signal

Spool position <SplActualValue> (21#144…146)	Electrical signal (negative logic)
Actual spool position value is within the failsafe monitoring window	Digital output 1 = 24 V
Actual spool position value is outside the failsafe monitoring window	Digital output 1 = 0 V

Failsafe spool position monitoring behavior depending on the DSM state <StatusWord> (0#38)

<StatusWord> (0#38) (DSM state)	Failsafe spool position monitoring behavior
'INIT'	No failsafe spool position monitoring active.
'DISABLED', 'HOLD', 'ACTIVE', 'FAULT DISABLED', 'FAULT HOLD'	Failsafe spool position monitoring active if one <DigitalOutputType> (0#220…221) is set to 1 (failsafe spool position monitoring on).

7.15.5.1 Object 0#210: Upper limit

The parameter <UpperLimit> (0#210) shows the upper limit of the spool position failsafe monitoring window.

ValveFailSafeWindowMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
0#210	UpperLimit	0x2421#0	INT16	ro	-	<LowerLimit> (0#211)…32767	16384

7.15.5.2 Object 0#211: Lower limit

The parameter <LowerLimit> (0#211) shows the lower limit of the spool position failsafe monitoring window.

ValveFailSafeWindowMonitoring							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
0#211	LowerLimit	0x2422#0	INT16	ro	-	–32768… <UpperLimit> (0#210)	–16384

7.15.5.3 Object 75#18: Spring Position Minimum

This parameter is optional set on request and depends on the valve model number. The parameter holds the lower limit of the failsafe position. This position will be valve specific calibrated and stored during production.

The expected failsafe position of the valve should be between <SpringPositionMinimum> (330775#18) and <SpringPositionMaximum> (330875#19).

SpringPositionMinimum							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
75#18	SpringPositionMinimum	0x3307#0	INT16	ro	Y	INT16	0

7.15.5.4 Object 75#19: Spring Position Maximum

This parameter is optional set on request and depends on the valve model number. The parameter holds the upper limit of the failsafe position. This position will be valve specific calibrated and stored during production.

The expected failsafe position of the valve should be between <SpringPositionMinimum> (330775#18) and <SpringPositionMaximum> (330875#19).

SpringPositionMaximum							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Per-sis-tence	Value range	Default
75#19	SpringPositionMinimum	0x3308#0	INT16	ro	Y	INT16	0

7.15.6 Pilot/single stage actual spool position monitoring

In case of a cable break, the fault code 0x11 (Pilot/single stage LVDT cable break) is thrown and the device state machine state changes to 'NOT READY'.

The servo valve must be serviced by Moog service technicians.
The servo valve may behave unpredictably.

7.15.7 Main/dual stage actual spool position monitoring

Monitoring is only active in case of device state greater 'INIT'. In case of a cable break, the fault code 0x14 (Main stage LVDT cable break) is generated and the device state machine state changes to 'NOT READY'.

The servo valve must be serviced by Moog service technicians.
The servo valve may behave unpredictably.

7.15.8 Analog input cable break monitoring

Cable break monitoring is available for the analog input signals and for the sensor supply wires. The following table shows the possible cable break monitoring features. The cable break monitoring depends on the input type. This dependence is explained in this chapter.

	Power supply cable break monitoring	**Sensor cable break monitoring**		
		0…10 mA –10…10 mA	**4…20 mA**	**0…10 V –10…10 V**
Analog input 0	No	No	Yes (fault code 31)	No
Analog input 1	No	No	Yes (fault code 32)	No
Analog input 2	Yes (fault code 28)	No	Yes if <AnalnMonitorCurrent2> (75#3) is set to 0 (fault code 33)	Yes if <AnalnMonitorCurrent2> (75#3) is set to 1 (fault code 33)
Analog input 3	Yes (fault code 29)	No	Yes if <AnalnMonitorCurrent3> (75#6) is set to 0 (fault code 34)	Yes if <AnalnMonitorCurrent3> (75#6) is set to 1 (fault code 34)
Analog input 4	Yes (fault code 30)	No	Yes if <AnalnMonitorCurrent4> (75#9) is set to 0 (fault code 35)	Yes if <AnalnMonitorCurrent4> (75#9) is set to 1 (fault code 35)

Table 77: Cable break monitoring features

The following fault codes are generated in the case of a cable break. The fault reaction itself can be configured using the fault reaction logic.
⇨ Chapter "8.1.3 Fault reaction type", page 261

Input	**Fault code**		**Fault description**
	Dec.	**Hex.**	
Analog input 0	31	0x1F	Analog input 0 current too low (4…20 mA)/ADC overflow (voltage)
Analog input 1	32	0x20	Analog input 1 current too low (4…20 mA)/ADC overflow (voltage)
Analog input 2	33	0x21	Analog input 2 current too low (4…20 mA)/ADC overflow (voltage)
	28	0x1C	Analog input 2 supply cable break/short circuit
Analog input 3	34	0x22	Analog input 3 current too low (4…20 mA)/ADC overflow (voltage)
	29	0x1D	Analog input 3 supply cable break/short circuit
Analog input 4	35	0x23	Analog input 4 current too low (4…20 mA)/ADC overflow (voltage)
	30	0x1E	Analog input 4 supply cable break/short circuit

Table 78: Possible fault codes

7.15.8.1 Object 75#3: Cable break monitoring analog input 2

If this parameter is set to 1 the cable break monitoring for the analog input 2 is enabled.

AnalogInput							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#3	AnaInMonitorCurrent2	0x3217#0	UINT8	rw	Y	0…1	0

7.15.8.2 Object 75#6: Cable break monitoring analog input 3

If this parameter is set to 1 the cable break monitoring for the analog input 3 is enabled.

AnalogInput							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#6	AnaInMonitorCurrent3	0x3228#0	UINT8	rw	Y	0…1	0

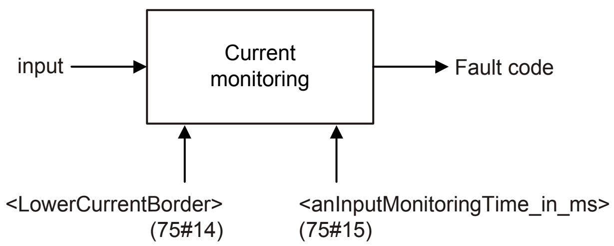
7.15.8.3 Object 75#9: Cable break monitoring analog input 4

If this parameter is set to 1 the cable break monitoring for the analog input 4 is enabled.

AnalogInput							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#9	AnaInMonitorCurrent4	0x3227#0	UINT8	rw	Y	0…1	0

If the 0…10 mA or the ±10 mA analog input type is used, no sensor wire cable break monitoring is available. In this case only the sensor power supply monitoring for the analog inputs 2…4 is active.

If the 4…20 mA analog input type is used, cable break monitoring for all analog inputs is available.



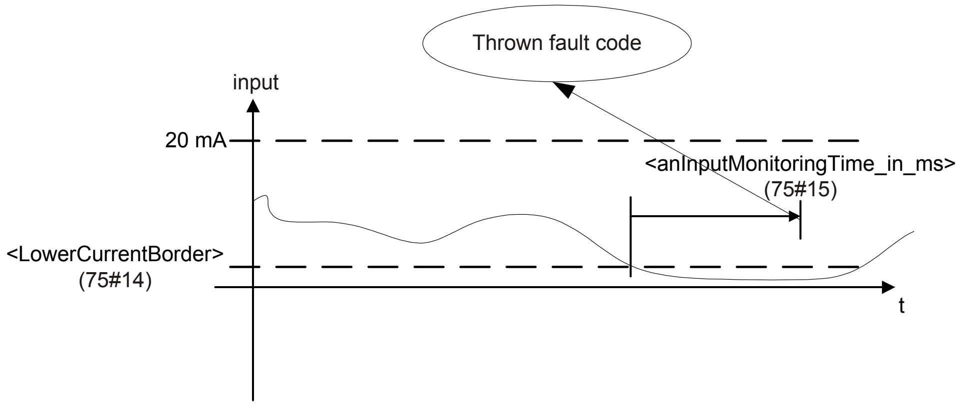


Figure 142: 4…20 mA analog input signal monitoring

If the current is below <LowerCurrentBorder> (75#14) for <anInputMonitoringTime_in_ms> (75#15) ms, cable break is detected.

If the 0…10 V or –10…10 V analog input type is used, cable break monitoring for all analog inputs is available. The sensor needs to be able to sink a current of at least 0.1 mA. A fault code is also generated by an input voltage above 11 V or an input voltage less than –11 V.

7.15.8.4 Object 75#14: Lower current border

This parameter contains the lower current boarder.

AnalogInput							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#14	LowerCurrentBorder	0x3250#0	FLOAT32	rw	Y	2.2…20.0	3.0

7.15.8.5 Object 75#15: Analog input monitoring time

This parameter contains the delay time (in ms) before generating the fault code.

AnalogInput							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#15	anInputMonitoringTime_in_ms	0x3251#0	UINT16	rw	Y	0…60000	10

7.15.9 Sensor power supply monitoring

For the analog inputs 2…4 a sensor power supply cable break monitoring exists. A sensor load current < 1 mA is interpreted as cable break. A short circuit of the sensor power supply is also recognized. Each sensor has its own fault detection. A short circuit on one of the sensors leads to a common fault response for all sensors.

7.15.10 Hardware monitoring

The hardware monitoring feature provides some hardware specific parameters such as power supply, board temperature and operating time.

7.15.10.1 Object 72#10: CPU supply voltage

This parameter contains the value of the CPU supply voltage (in mV). A fault is generated if the parameter is outside its nominal range.

Nominal range: 3.1 V ≤ CpuSupplyVoltage ≤ 3.5 V

Hardware_DiagnosticData							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#10	CpuSupplyVoltage	0x2803#0	UINT16	ro	-	UINT16	None

Value description

<CpuSupplyVoltage>	**Thrown fault code**	**Fault description**
CpuSupplyVoltage < 3.1 V	0x07	Internal supply voltage too low.
CpuSupplyVoltage > 3.5 V	0x08	Internal supply voltage too high.

Table 79: Fault codes

The power supply voltage should be in the range of 18…32 V to ensure proper operation.

7.15.10.2 Object 72#11: Power supply voltage

This parameter holds the value of the power supply voltage (in mV). A fault is generated if the parameter is outside its nominal range.

Nominal range: 17 V ≤ PowerSupplyVoltage ≤ 32.5 V

Hardware_DiagnosticData							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#11	PowerSupplyVoltage	0x2804#0	UINT16	ro	-	UINT16	None

Fault description

<PowerSupplyVoltage>	**Thrown fault code**	**Fault description**
PowerSupplyVoltage < 17 V	0x05	Power supply voltage too low.
PowerSupplyVoltage > 32.5 V	0x06	Power supply voltage too high.

Table 80: Fault codes

7.15.10.3 Object 72#12: PCB temperature

This parameter contains the temperature (in °C) of the servo valve electronics. A fault is generated if the parameter value is below or exceeds the following temperature values:

PCB temperature < –20 °C
PCB temperature > 85 °C
PCB temperature > 105 °C

Hardware_DiagnosticData							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#12	PcbTemperature	0x2805#0	INT16	ro	-	INT16	None

Fault description

<PowerSupplyVoltage>	**Thrown fault code**	**Fault description**
PCB temperature < –20 °C	0x0D	Electronics temperature too low.
PCB temperature > 85 °C	0x0E	Electronics temperature too high.
PCB temperature > 105 °C	0x0F	Electronics temperature exceeded.

Table 81: Fault codes

The PCB temperature should not exceed the range of –20…85 °C to ensure proper operation. The electronics temperature has a big impact on the electronics lifetime. The longest service life is achieved when the higher temperatures are avoided.

7.15.10.4 Object 72#9: Maximal PCB temperature

This parameter shows the maximal reached temperature of the PCB. The customer is able to reset the value. Regardless of which value is written to this parameter, the value is set to 0. The valve will automatically increase this parameter to the actual maximum temperature.

Hardware_DiagnosticData							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#9	PcbMaxTemperatureCustomer	0x2809#0	INT16	rw	Y	INT16	0

7.15.10.5 Object 72#24…25: Operating time

The parameter <PowerOnTime> (72#24) contains the power on time (in minutes) since production of the servo valve.

The parameter <OperatingTime> (72#25) contains the time (in minutes) the servo valve is in the device state machine (DSM) states 'HOLD', 'FAULT HOLD' or 'ACTIVE'.

Hardware_DiagnosticData							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#24	PowerOnTime	0x280D#1	UINT32	ro	-	UINT32	0
72#25	OperatingTime	0x280D#2	UINT32	ro	-	UINT32	0

7.16 Event handler

With the event handler, the user can define expressions to calculate a value depending on internal and external parameter values and assign it to any parameter.

All actions are based on parameter access, so the behavior of the servo valve can be influenced similar to an external parameter access via the field bus. The variables calculated by the event handler can be accessed via the field bus. There are in total eight event handlers, which will be configured by eight associated strings processed by the expression parser.

After an event handler is configured, it needs to be enabled in order to process its low-level code. The parser processes the expressions according the syntax and the processing order similar to that of the programming language C. All event calculations are processed every main task cycle of the firmware, which is processed at least every 2 ms.

7.16.1 Event expressions

The event expressions are strings with maximal 192 characters and are built in the following manner:

- The parameters to be used can only be accessed through their short names. The short names of a parameter can be found in the chapter "Object dictionary"
  ⇨ Chapter "10 Object dictionary", page 288
- Integer constants

Prefix	Description
	Decimal integer constants
0x	Hexadecimal integer constants

- Floating point constants are not allowed.

There are groups of operators. Inside a group there is no priority of operation, the expressions are processed from left to right. The groups are listed in order of their priority.

- Operators with one operant

Symbol	Description
–	Negate / negative sign
+	Positive sign
#	Absolute
~	Bitwise NOT
!	Logical NOT

- Mathematical operators high priority

Symbol	Description
/	Divide
*	Multiply

- Mathematical operators low priority

Symbol	Description
–	Subtract
+	Add

- Shift operators

Symbol	Description
>>	Shift right
<<	Shift left

- Logical compare operators

Symbol	Description
<, <=	Smaller, smaller or equal
>, >=	Greater, greater or equal

- Logical operators for conditions

Symbol	Description
==	Equal
!=	Not equal

- Bitwise operator AND

Symbol	Description
&	Bitwise AND

- Bitwise operator OR

Symbol	Description
|	Bitwise OR

- Logical operator AND

Symbol	Description
&&	Logical AND

- Logical operator OR

Symbol	Description
||	Logical OR

- If-then-else command operators
  Condition ? ifstatement : elsestatement '

Symbol	Description
?	If and then operator
:	Else operator
'	If-then-else terminators

- Assignment operator

Symbol	Description
=	Assign

- Expressions can be concatenated using a semicolon

Symbol	Description
;	Separate

- Brackets are restricted to 3 levels

Symbol	Description
(	Open bracket
)	Close bracket

7.16.1.1 Object 71#31: Event expression 1

This parameter contains the expression string of the first event handler.

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#31	EventExpression_1	0x2901#0	STRING	rw	Y	None	""

7.16.1.2 Object 71#32: Event expression 2

This parameter contains the expression string of the second event handler.

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#32	EventExpression_2	0x2902#0	STRING	rw	Y	None	""

7.16.1.3 Object 71#33: Event expression 3

This parameter contains the expression string of the third event handler.

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#33	EventExpression_3	0x2903#0	STRING	rw	Y	None	""

7.16.1.4 Object 71#34: Event expression 4

This parameter contains the expression string of the fourth event handler.

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#34	EventExpression_4	0x2904#0	STRING	rw	Y	None	""

7.16.1.5 Object 71#35: Event expression 5

This parameter contains the expression string of the fifth event handler.

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#35	EventExpression_5	0x2905#0	STRING	rw	Y	None	""

7.16.1.6 Object 71#36: Event expression 6

This parameter contains the expression string of the sixth event handler.

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#36	EventExpression_6	0x2906#0	STRING	rw	Y	None	""

7.16.1.7 Object 71#3: Event expression 7

This parameter contains the expression string of the seventh event handler.

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#37	EventExpression_7	0x2907#0	STRING	rw	Y	None	""

7.16.1.8 Object 71#38: Event expression 8

This parameter contains the expression string of the eighth event handler.

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#38	EventExpression_8	0x2908#0	STRING	rw	Y	None	""

7.16.1.9 Object 71#39…46: Event enable

These parameters switch the event handler on or off.

Eventhandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#39	EventEnable_1	0x2909#1	UINT8	rw	Y	0…1	0
71#40	EventEnable_2	0x2909#2	UINT8	rw	Y	0…1	0
71#41	EventEnable_3	0x2909#3	UINT8	rw	Y	0…1	0
71#42	EventEnable_4	0x2909#4	UINT8	rw	Y	0…1	0
71#43	EventEnable_5	0x2909#5	UINT8	rw	Y	0…1	0
71#44	EventEnable_6	0x2909#6	UINT8	rw	Y	0…1	0
71#45	EventEnable_7	0x2909#7	UINT8	rw	Y	0…1	0
71#46	EventEnable_8	0x2909#8	UINT8	rw	Y	0…1	0

7.16.2 Event handler examples

The following three simple examples explain the event handler behavior. For many event handler tasks, buffer parameters are needed. The parameters with the short names varu[…], vars[…], dumu[…] and dums[…] are reserved for these tasks.

⇨ Chapter "6.10 Free to use parameters", page 114

Example 1:

varu32[1]=varu32[1]+1;splset=varu32[1]*5

- Calculations are processed every main task cycle (no condition).
- Increase varu32[1] by one.
- Calculates the spool position setpoint value as five times the varu32[1] variable.

Example 2:

splval>10000?splset=0:(splval<1000?splset=11000)

- Two separate events in one expression using a semicolon as separator.
- Spool setpoint value is only changed if the spool position actual value meets the condition splval greater 10000 or smaller 1000.

Example 3:

Expression 1 <EventExpression_1> (71#31):
prsval>10000?ctlmod=4;evtena[0]=0;evtena[1]=1

Expression 2 <EventExpression_2> (71#32):
posset<1000?ctlmod=9;evtena[0]=1;evtena[1]=0

- Expressions can disable themselves and activate other expressions.
- Expression 1 activates pressure control.
- Expression 2 activates spool position control.
- evtena[0] enables event expression 1, evtena[1] enables event expression 2, etc.

Example 4:

Combination of example 1 and example 2:
splval>10000?splset=0:(splval<1000?splset=11000)' varu32[1]=varu32[1]+1

- If-then-else in combination with permanently expression.
- Two separate events in one expression using a semicolon as separator.
  - Spool setpoint value is only changed if the spool position actual value meets the condition splval greater 10000 or smaller 1000.
- Terminate If-then-else-if condition with '.
- Additional expression which is executed permanently to increase varu32[1] by one.

7.17 Data logger

The data logger is a four channel oscilloscope to trace the parameters inside the servo valve. The parameters which are monitored can be chosen. The trigger condition, pre trigger, trigger level, slopes and scaling can be set. The servo valve contains a 2 Kbytes volatile data memory which can be used to sample the data. The data can be downloaded at any time.

It is not possible to read the data logger memory via Profibus. This is only possible via the local CAN bus.

7.17.1 Data logger state machine

The data logger is controlled by a state machine. The states and the transitions are explained in this chapter.

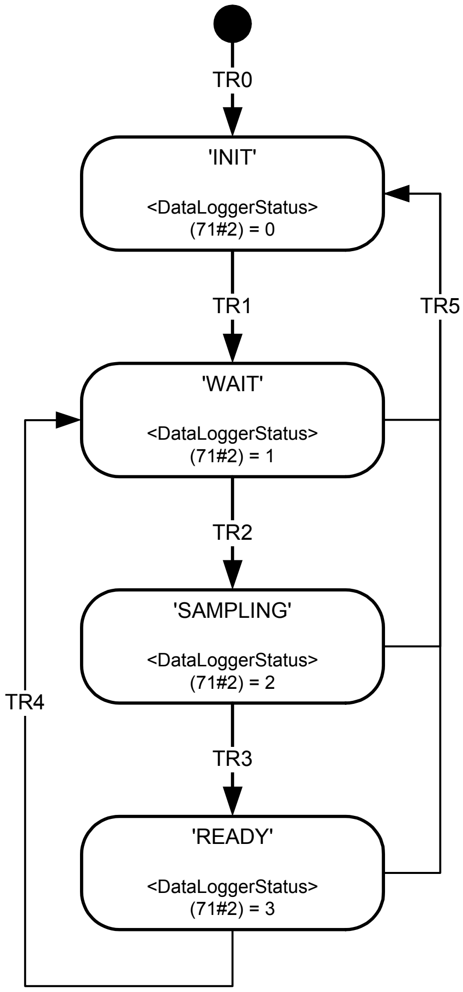


Figure 143: Data logger state machine

State	**<DataLoggerTriggerType> (71#22)**		
	0 (Free)	**1 (Normal)**	**2 (Single)**
'INIT'	Initialize data logger.		
'WAIT'	The trigger condition is always true. If the <Control> (71#1) is set to 1 (enable data logger), the state changes to 'SAMPLE'.	If the <Control> (71#1) is set to 1 (enable data logger), the channels are continuously sampled to allow pre trigger. If the trigger condition is active, the start index <SampleStartOffset> (71#21) in the ring buffer memory <Memory> (71#20) is defined and the state is change to 'SAMPLE'.	
'SAMPLE'	The channels are continuously sampled until the ring buffer is full. If the ring buffer is full the state changes to 'READY'.		
'READY'	If the <Control> (71#1) is set to 1 (enable data logger), the state changes to 'WAIT'.	State stays in 'READY' until <Control> (71#1) is set to 1 (enable data logger).	

Table 82: States of the data logger state machine

Transition (TR)	Description
TR0	Start program.
TR1	Triggered by setting <Control> (71#1) to 1 (enable data logger).
TR2	Trigger condition is active.
TR3	Data logger ring buffer is full.
TR4	Triggered by setting <Control> (71#1) to 1 (enable data logger).
TR5	One of the following parameters has changed: <Divider> (71#3) <EnableChannel1…4> (71#5…8) <ChannelParameter1…4> (71#9…12) <TriggerType> (71#22) <TriggerParameter> (71#23)

Table 83: Transitions of the data logger state machine

7.17.1.1 Object 71#1: Control

This parameter enables the data logger.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#1	Control	0x3180#0	UINT8	rw	N	0…1	None

Value description

<Control>	Description
0	Disable data logger.
1	Enable data logger.

Table 84: Possible values of parameter <Control> (71#1)

7.17.1.2 Object 71#2: Status

State of the data logger state machine.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#2	Status	0x3181#0	UINT8	ro	-	0…3	None

Value description

<Status>	Description
0	'INIT'
1	'WAIT'
2	'SAMPLING'
3	'READY'

Table 85: Possible values of parameter <Status> (71#2)

7.17.2 Channel settings

Four channels can be used within the data logger. All readable parameters are available as input for the channels.

7.17.2.1 Object 71#9…12: Channel parameter

The parameters <ChannelParameter1…4> (71#9…12) define the parameters which shall be sampled. The values are composed of the CANopen index, sub-index and bit length of the chosen parameter.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#9	ChannelParameter1	0x3185#1	UINT32	rw	Y	UINT32	0x63100110
71#10	ChannelParameter2	0x3185#2	UINT32	rw	Y	UINT32	0x63010110
71#11	ChannelParameter3	0x3185#3	UINT32	rw	Y	UINT32	0x63900110
71#12	ChannelParameter4	0x3185#4	UINT32	rw	Y	UINT32	0x63810110

Value description

<ChannelParameter1…4>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length
Example	0x63	0x10	0x01	0x10

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

The example value is 0x63100110, which refers to the <SplDemandValue> (21#24...26), with the CANopen index 0x6310 and the CANopen), sub-index 0x01 with a length of 16 bit (16=0x10).

7.17.2.2 Object 71#5…8: Enable channel

Any channel can be switched on or off with this parameter.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#5	EnableParameter1	0x3184#1	UINT8	rw	Y	0…1	0
71#6	EnableParameter2	0x3184#2	UINT8	rw	Y	0…1	0
71#7	EnableParameter3	0x3184#3	UINT8	rw	Y	0…1	0
71#8	EnableParameter4	0x3184#4	UINT8	rw	Y	0…1	0

Value description

<EnableParameter1…4>	**Description**
0	Channel disabled.
1	Channel enabled.

Table 86: Possible values of parameter <EnableParameter> (71#5…8)

7.17.3 Sample frequency

The maximum sample frequency is limited to 10000 samples per seconds (10kHz). This sample frequency can be set to a smaller sample frequency with the parameter <Divider> (71#3).

$$\text{New sample frequency} = \frac{\text{Maximum sample frequency}}{\text{<Divider> (0x71\#3)}}$$

The measuring time is increased by the factor <Divider> (71#3).

7.17.3.1 Object 71#3: Divider

This parameter contains an integer number to reduce the sampling frequency.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#3	Divider	0x3182#0	UINT16	rw	Y	1…65535	1

Value description

<Divider>	**Description**
1	10000 samples per second (10 kHz).
2…65534	10000/2…10000/65534 samples per second.
65535	10000/65535 = 0.1526 samples per second.

Table 87: Possible values of parameter <Divider> (71#3)

7.17.4 Trigger settings

The trigger settings can be set like on a real oscilloscope.

- Trigger parameter (integer parameter for the trigger).
- Trigger type (FREE, NORMAL, SINGLE).
- Trigger level or bitmask (trigger level).
- Trigger coupling (AC, DC, BITMASK).
- Trigger slope (rising, falling, both).
- Trigger position (pre trigger, post trigger).

7.17.4.1 Object 71#23: Trigger parameter

The <TriggerParameter> (71#23) defines the parameter which is used as trigger signal.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#23	TriggerParameter	0x3189#0	UINT32	rw	Y	UINT32	0x63100110

Value description

<TriggerParameter>				
Byte	3	2	1	0
Description	CANopen index MSB	CANopen index LSB	CANopen sub-index	Parameter bit length
Default	0x63	0x10	0x01	0x10

This pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 0x10.

The default value is 0x63100110, which refers to the <SplDemandValue> (21#24...26), with the CANopen index 0x6310 and the CANopen, sub-index 0x01 with a length of 16 bit (16=0x10).

7.17.4.2 Object 71#22: Trigger type

This parameter contains the trigger type of the data logger.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#22	TriggerType	0x3188#0	UINT8	rw	Y	0…2	1

Value description

<TriggerType>	Description
0	FREE (the trigger condition is always true).
1	NORMAL (if the trigger condition matches, the data logger returns to wait state completed).
2	SINGLE (if the trigger condition matches, the data logger enters the ready state).

Table 88: Possible values of parameter <TriggerType> (71#22)

7.17.4.3 Object 71#26: Trigger level or bitmask

This parameter contains the trigger level if the <TriggerCoupling> (71#26) is set to 0 or 1.

This parameter contains the BITMASK if the <TriggerCoupling> (71#26) is set to 2. The BITMASK selects the bits to be compared with the trigger signal.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#26	TriggerLevelOrBitmask	0x318C#0	INT32	rw	Y	INT32	0

7.17.4.4 Object 71#24: Trigger coupling

This parameter contains the trigger coupling type of the data logger.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#24	TriggerCoupling	0x318A#0	UINT8	rw	Y	0…2	1

Value description

<TriggerCoupling>	**Description**
0	AC (AC part of the trigger signal is passed and is compared with the trigger level).
1	DC (the trigger signal is passed directly and is compared with the trigger level).
2	BITMASK (the trigger signal is passed directly and is compared bit-wise with the trigger bitmask).

Table 89: Possible values of parameter <TriggerCoupling> (71#24)

7.17.4.5 Object 71#25: Trigger slope

The <TriggerSlope> (71#25) defines the edge of the signal which starts the sampling procedure.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#25	TriggerSlope	0x318B#0	UINT8	rw	Y	1…3	1

Value description

<TriggerSlope>	**Description**
1	Rising (trigger on a rising edge).
2	Falling (trigger on a falling edge).
3	Both (trigger on both, rising or falling edge).

Table 90: Possible values of parameter <TriggerSlope> (71#25)

7.17.4.6 Object 71#27: Trigger position

The <TriggerPosition> (71#27) is provided as number of samples which shifts the starting point in the ring buffer.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#27	TriggerPosition	0x318D#0	INT32	rw	Y	INT32	0

Value description

<TriggerPosition>	**Description**
> 0	Post trigger.
= 0	No delay.
< 0	Pre trigger.

Table 91: Possible values of parameter <TriggerPosition> (71#27)

7.17.5 Data memory

The data logger memory is organized as a ring buffer. After the data logger has finished a sampling task and changed the state from 'SAMPLE' to 'READY', the sampled data are valid. The sampled data start from the byte number <SampleStartOffset> (71#21) until the byte 2047 and continue from byte 0 to byte <SampleStartOffset> (71#21) – 1.

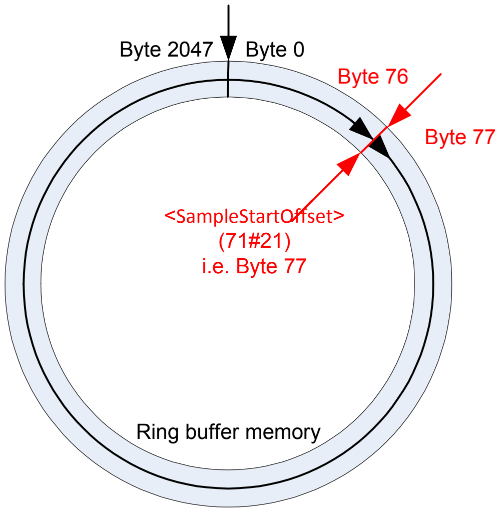


Figure 144: Data memory organization

The number of possible samples depends on the channel configuration. The next three examples illustrate the data logger memory management:

Example 1: Sampling three channels with mixed data types, 1, 2 and 4 bytes

In this example three channels are enabled, channels 1, 3 and 4. Each enabled channel contains a different data type with a different length (1, 2 and 4 byte). Channel one samples a one byte parameter, channel 3 a two byte parameter and channel 4 a four byte parameter. In every sample step 7 bytes memory are needed. The available memory is 2048 bytes. The possible number of samples is 2048 bytes/7 bytes per sample, so 292 samples can be saved in the memory. 292 samples multiplied with 7 bytes per sample are equal 2044 bytes. So an empty rest of 4 bytes remains at the end of the ring buffer. If the parameter <Divider> (71#3) is set to 1 (10000 samples per second), the memory is filled in 292/10000 = 29.2 ms.

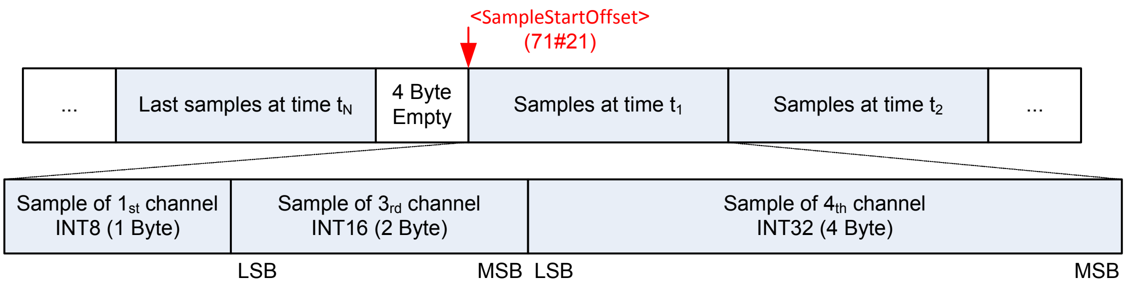


Figure 145: Data memory - mixed channel data

Example 2: Sampling one channel with a one byte parameter

In this example only one channel, channel number 3, is enabled. For every sample one byte memory is needed. The available memory is 2048 bytes. The possible number of samples is 2048 bytes divided by 1 byte per sample, so 2048 samples can be taken. No empty rest remains at the end of the ring buffer. If the parameter <Divider> (71#3) is set to 1 (10000 samples per second), the memory is filled in 2048/10000 = 204.8 ms.

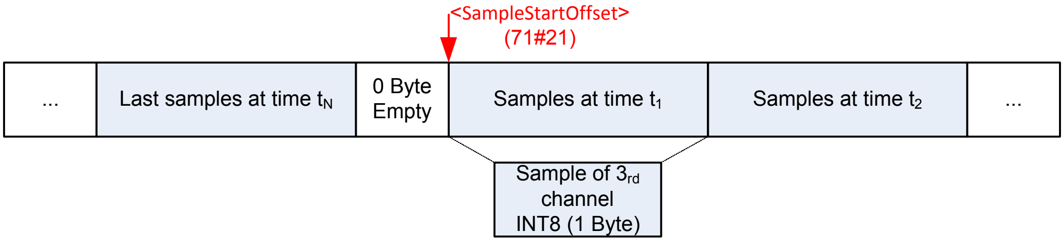


Figure 146: Data memory - one channel with INT8 parameter

Example 3: Sampling four channels with 4 four byte parameters

In this example all four channels are enabled. For every sample 16 bytes memory are needed. The available memory is 2048 bytes. The possible number of samples is 2048 bytes divided by 16 byte per sample, so 128 samples can be saved in the memory. No empty space remains at the end of the ring buffer. If the parameter <Divider> (71#3) is set to 1 (10000 samples per second), the memory is filled in 128/10000 = 12.8 ms.

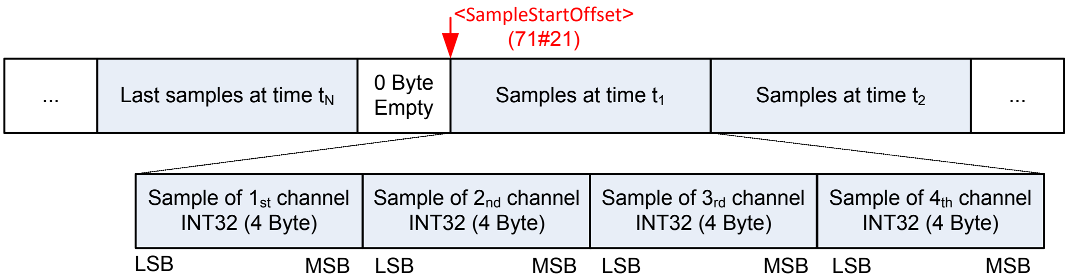


Figure 147: Data memory - four channels with INT32 parameters

7.17.5.1 Object 0x3186: Memory

The <Memory> is only available via CAN SDO. There is no access via Profibus for this data type domain.

The parameter <Memory> (0x3186) contains the sampled information of the four channels. The parameter is an array of UINT8 with 2048 entries/bytes.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
None	Memory	0x3186#0	DOMAIN	ro	-	None	None

7.17.5.2 Object 71#21: Sample start offset

The <SampleStartOffset> (71#21) contains the byte position where the recorded data start. It indicates the position of the first sample point.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#21	SampleStartOffset	0x3187#0	UINT32	ro	-	UINT32	None

7.17.5.3 Object 71#4: Number of samples

The parameter <NumberOfSamples> (71#4) contains the number of sample points. On each sample point the data of all active channels are recorded.

DataLogger							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#4	NumberOfSamples	0x3183#0	INT32	ro	-	0…2048	None

7.18 Function generator

The servo valve internal function generator can be used to generate a periodic signal with specific shapes, thereby enabling an engineer or technician to test and examine a servo valve.
The function generator has two outputs:

- An output signal
  This signal can be used e.g. as setpoint value for the servo valve to optimize the servo valves behavior.
  Different shapes, amplitude and offset can be configured.
  ⇨ Chapter "7.18.3.1 Object 71#121: Output signal", page 254
- A trigger signal
  This is a rectangular signal, with fixed amplitude, without offset.
  This signal can be used e.g. to trigger the data logger.
  ⇨ Chapter "7.18.3.2 Object 71#122: Square output (Trigger signal)", page 255

7.18.1 Function generator output signal shapes

The function generator output signal can be influenced by changing the function type, the magnitude, the offset, the sign or the frequency. The available shapes and the parameters to define the shapes are explained in this chapter.

7.18.1.1 Rectangular output signal (type 1)

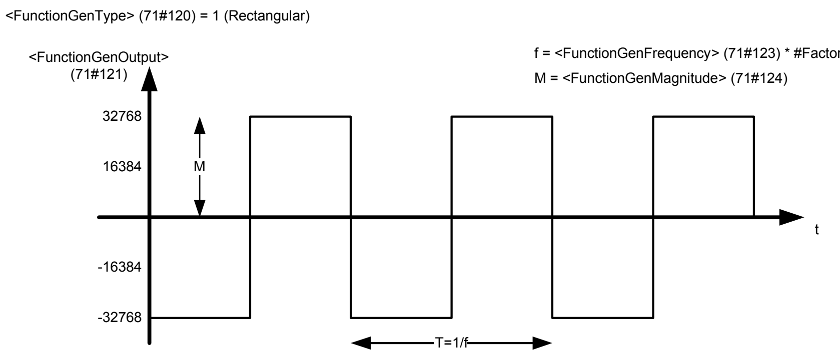


Figure 148: Rectangular output signal (type 1)

7.18.1.2 Triangle output signal (type 2)

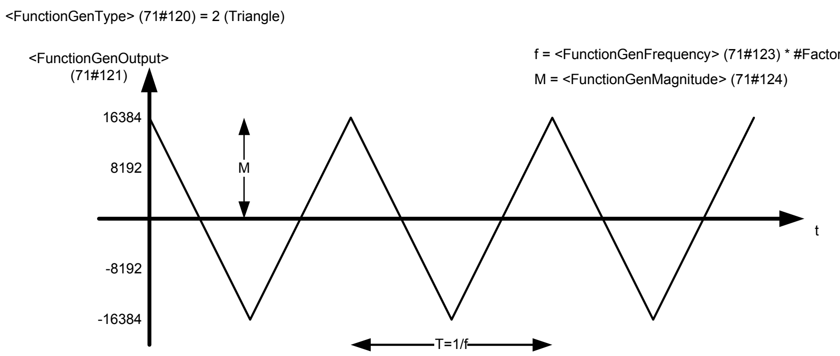


Figure 149: Triangle output signal (type 2)

7.18.1.3 Sawtooth signal (type 3)

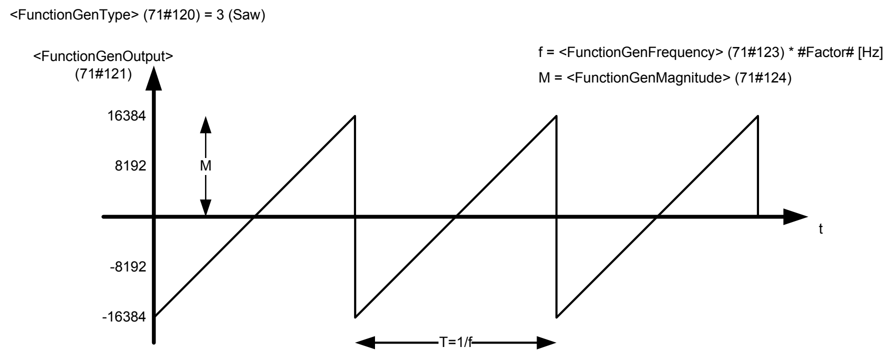


Figure 150: Sawtooth signal (type 3)

The output signal can be inverted by setting the parameter <FunctionGenSign> (71#126) to –1.
⇨ Chapter "7.18.1.9 Object 71#126: Sign", page 253

7.18.1.4 Trapezoid signal (type 4)

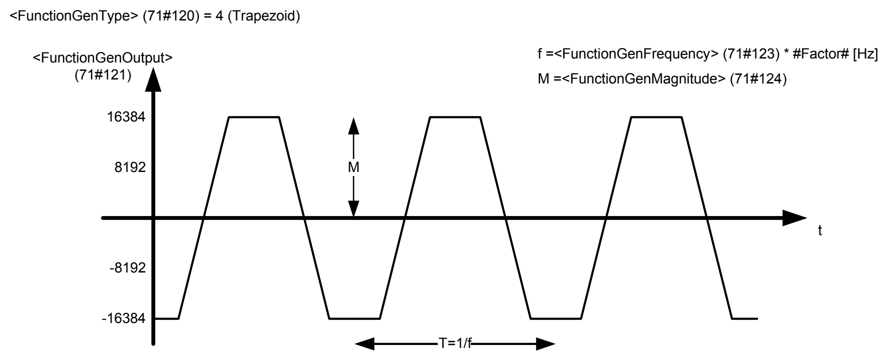


Figure 151: Trapezoid signal (type 4)

7.18.1.5 Sine signal (type 5)

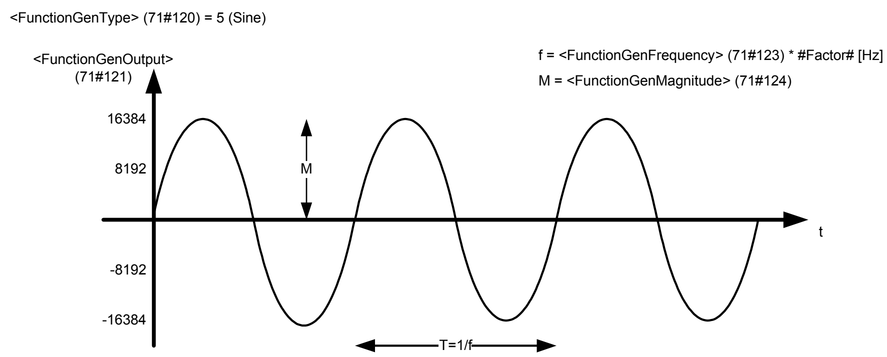


Figure 152: Sine signal (type 5)

7.18.1.6 Object 71#120: Type

This parameter defines the function generator output signal shape.

FunctionGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#120	FunctionGenType	0x3100#0	INT8	rw	Y	0…5	0

Value description

<FunctionGenType>	**Description**
0	Function generator switched off.
1	Rectangular signal output.
2	Triangle signal output.
3	Sawtooth signal output.
4	Trapezoid signal output.
5	Sine signal output.

Table 92: Possible values of parameter <Type> (71#120)

7.18.1.7 Object 71#124: Magnitude

This parameter is the magnitude of the function generator output signal in increments. To configure e.g. a set-point value of ±100 %, 16384 has to be set as magnitude.

FunctionGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#124	FunctionGenMagnitude	0x3104#0	INT16	rw	N	0…32767	0

7.18.1.8 Object 71#125: Offset

This parameter is the offset of the function generator output signal in increments.

FunctionGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#125	FunctionGenOffset	0x3105#0	INT16	rw	N	INT16	0

7.18.1.9 Object 71#126: Sign

This parameter is the sign of the function generator output signal.

FunctionGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#126	FunctionGenSign	0x3107#0	INT8	rw	Y	–1…1	1

7.18.2 Function generator output signal frequency

The function generator output frequency is defined as:

f = <FunctionGenFrequency> (71#123) · #Factor#

The #Factor# is dependent on the frequency prefix parameter <FunctionGenFrequencyPrefix> (71#127). The Frequency unit is Hertz [Hz] or [1/s].

7.18.2.1 Object 71#123: Frequency

This parameter defines the function generator output signals frequency which is multiplied with the frequency prefix factor to get the function generator output signals frequency in Hertz [Hz] or [1/s].

FunctionGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#123	FunctionGenFrequency	0x3103#0	UINT16	rw	Y	1…10000	10

7.18.2.2 Object 71#127: Frequency prefix

This parameter defines a factor which is multiplied with the parameter <FunctionGenFrequency> (71#123) to get the function generator output signals frequency in Hertz [Hz] or [1/s].

FunctionGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#127	FunctionGenFrequencyPrefix	0x3108#0	INT8	rw	Y	–4…0	0

Value description

<FunctionGenFrequencyPrefix>	**Resulting #Factor#**
0	1.0
–1	0.1
–2	0.01
–3	0.001
–4	0.0001

Table 93: Possible values of parameter <FunctionGenFrequencyPrefix> (71#127)

7.18.3 Function generator output signals

The function generator output signals can be used for several tasks. An important one is the use as the setpoint for the servo valve. This can be done by mapping the output signal to one of the following servo valve setpoint inputs:

- **Spool position setpoint value:** <SplSetpointParameter> (0#204)
  ➪ Chapter "6.2.3.2 Object 0#204: Setpoint parameter", page 55
- **Pressure setpoint value:** <PrsSetpointParameter> (0#203)
  ➪ Chapter "6.2.4.2 Object 0#203: Setpoint parameter", page 57

7.18.3.1 Object 71#121: Output signal

This is the function generator output signal.

FunctionGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#121	FunctionGenOutput	0x3101#0	INT16	ro	-	INT16	None

7.18.3.2 Object 71#122: Square output (Trigger signal)

This signal has the same frequency as the function generator output signal but is fixed to a rectangular shape and with a magnitude of 32767 increments with no offset.

FunctionGenerator							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#122	FunctionGenSquareOutput	0x3102#0	INT16	ro	-	INT16	None

Value description

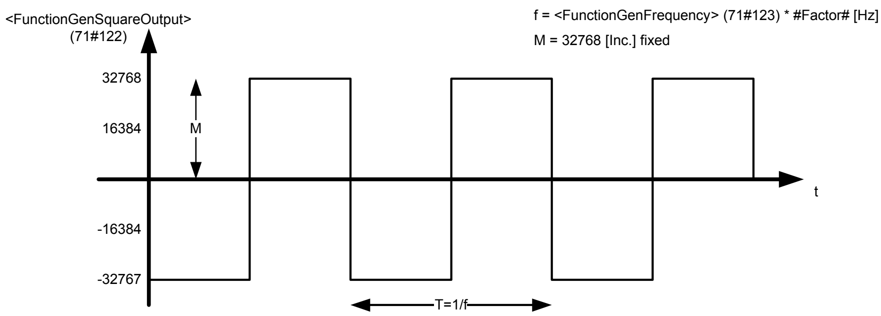


Figure 153: Trigger signal

8 Diagnostics

8.1 Fault reaction

If a malfunction occurs, the servo valve software throws a fault. The corresponding fault is set in the fault state <FaultStatus> (72#26…29) and <FaultRetainStatus> (72#42…45) parameters in bit coded form. If no fault reaction for this fault is defined within the parameter <FaultReactionType> (72#102…219) no further fault reaction is done. If a fault reaction is configured for the thrown fault code at least an emergency message is sent and the <ErrorRegister> (64#53) parameter will be set according to the error group of the fault code. Then the fault code and the error code are saved in an array <StandardErrorField> (64#35…51) which holds the last eight thrown faults. Afterwards the error description string is saved in the array <FaultReactionDescription> (72#40). The <FaultHistoryNumber> (72#41) is set to the number of thrown faults.

If the configured fault reaction of the actual thrown fault requests a change of the device state, the corresponding transition of the device state machine will be forced.

The emergency message is a diagnosis telegram to the diagnosis buffer.

Details about Profibus DP-V0 standard: ⇨ Chapter "2.10 Diagnostic (DP-V0)", page 27

8.1.1 Fault reaction flow chart

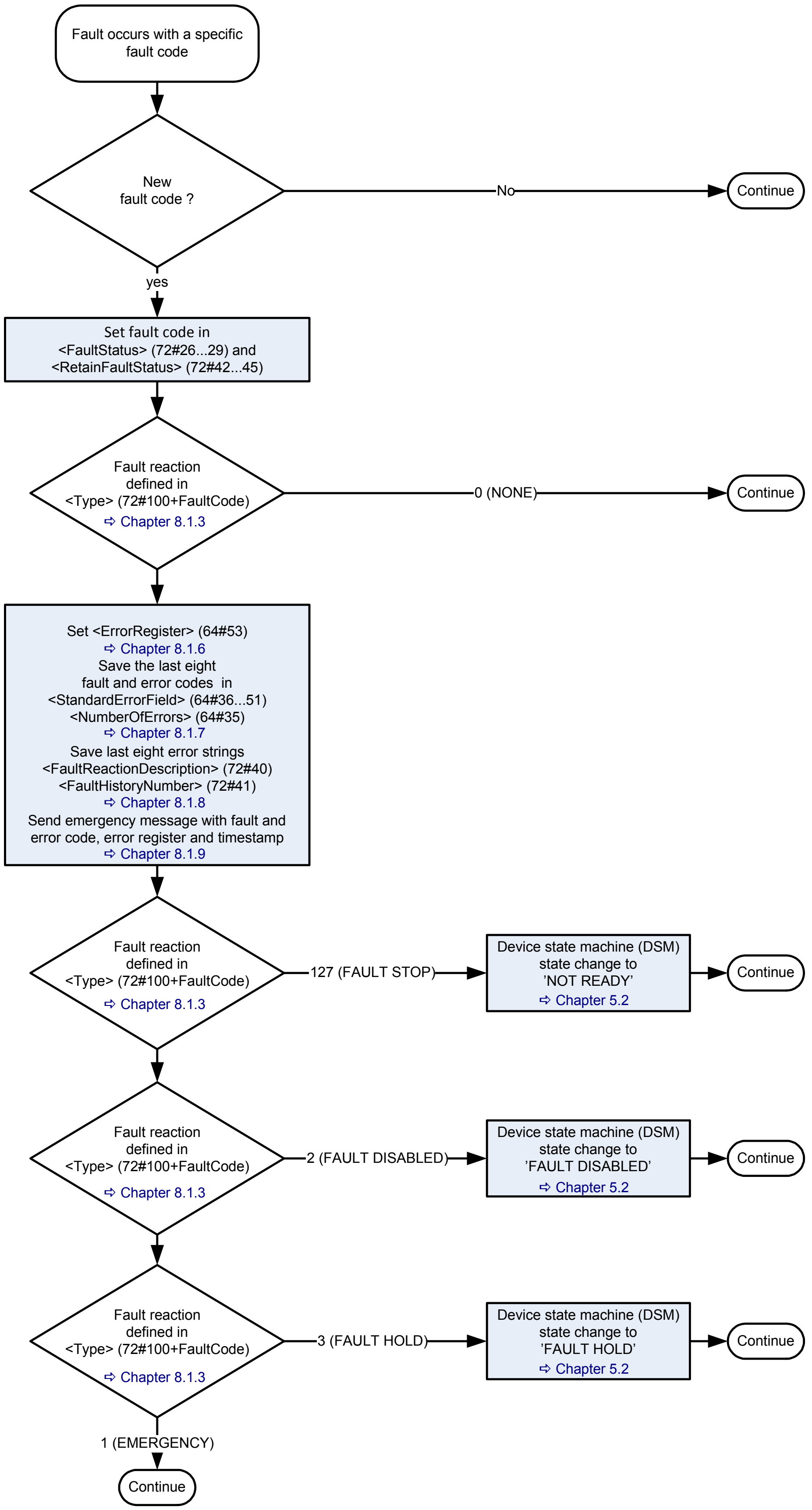


Figure 154: Fault reaction flow chart

8.1.2 Possible fault codes

The possible fault codes are shown in the following table. The used fault codes depend on the servo valve model and hardware. The fault code 0 means that no fault is pending.

To get the index of the fault reaction type <FaultReactionType> (72#101…219), the fault code must be incremented by 101.

Fault code			Dword		<FaultReactionType> (72#101…219)		
Dec.	Hex.	Fault description	Num	Bit	Index – 100	Value range	De-fault
0	0x00	Error reset or no error	-	-	1	0	0
1	0x01	Error microprocessor core (not implemented - for further use)	1	0	2	127	127
2	0x02	Error during startup	1	1	3	127	127
3	0x03	Error DSP program download	1	2	4	127	127
4	0x04	Error DSP realtime data transmission	1	3	5	0...3	2
5	0x05	Power supply voltage too low	1	4	6	0...3	2
6	0x06	Power supply voltage too high	1	5	7	0...3	2
7	0x07	Internal supply voltage too low	1	6	8	127	127
8	0x08	Internal supply voltage too high	1	7	9	127	127
9	0x09	Internal reference voltage too low (not implemented - for further use)	1	8	10	127	127
10	0x0A	Internal reference voltage too high (not implemented - for further use)	1	9	11	127	127
11	0x0B	Internal current too low (not implemented - for further use))	1	10	12	127	127
12	0x0C	Internal current too high (not implemented - for further use)	1	11	13	127	127
13	0x0D	Electronics temperature too low (< –20 °C)	1	12	14	0...3	2
14	0x0E	Electronics temperature too high (> 85 °C)	1	13	15	0...3	1
15	0x0F	Electronics temperature exceeded (> 105 °C)	1	14	16	0...3	2
16	0x10	Current sensor circuit failure (not implemented - for further use)	1	15	17	127	127
17	0x11	Pilot/single stage LVDT cable break	1	16	18	127	127
18	0x12	Pilot/single stage LVDT position out of range (not implemented - for further use)	1	17	19	127	127
19	0x13	Pilot/single stage LVDT circuit failure (not implemented - for further use)	1	18	20	127	127
20	0x14	Main stage LVDT cable break	1	19	21	0...3	0
21	0x15	Main stage LVDT position out of range (not implemented - for further use)	1	20	22	0...3	0
22	0x16	Main stage LVDT circuit failure (not implemented - for further use)	1	21	23	127	127
23	0x17	Internal pressure transducer cable break (not implemented - for further use)	1	22	24	127	127
24	0x18	Internal pressure transducer circuit failure (not implemented - for further use)	1	23	25	127	127
25	0x19	Internal pressure transducer pressure peak (not implemented - for further use)	1	24	26	0...3	0
26	0x1A	Analog input 0 supply cable break/short circuit (not implemented - for further use)	1	25	27	0...3	0
27	0x1B	Analog input 1 supply cable break/short circuit (not implemented - for further use)	1	26	28	0...3	0
28	0x1C	Analog input 2 supply cable break/short circuit	1	27	29	0...3	0
29	0x1D	Analog input 3 supply cable break/short circuit	1	28	30	0...3	0
30	0x1E	Analog input 4 supply cable break/short circuit	1	29	31	0...3	0
31	0x1F	Analog input 0 current too low (4...20 mA)/ADC overflow (voltage)	1	30	32	0...3	0

Table 94: Possible fault codes (part 1 of 4)

Fault code			Dword		<FaultReactionType> (72#101…219)		
Dec.	**Hex.**	**Fault description**	**Num**	**Bit**	**Index – 100**	**Value range**	**De-fault**
32	0x20	Analog input 1 current too low (4...20 mA)/ADC overflow (voltage)	1	31	33	0...3	0
33	0x21	Analog input 2 current too low (4...20 mA)/ADC overflow (voltage)	2	0	34	0...3	0
34	0x22	Analog input 3 current too low (4...20 mA)/ADC overflow (voltage)	2	1	35	0...3	0
35	0x23	Analog input 4 current too low (4...20 mA)/ADC overflow (voltage)	2	2	36	0...3	0
36	0x24	Analog input 0 circuit failure (not implemented - for further use)	2	3	37	0...3	0
37	0x25	Analog input 1 circuit failure (not implemented - for further use)	2	4	38	0...3	0
38	0x26	Analog input 2 circuit failure (not implemented - for further use)	2	5	39	0...3	0
39	0x27	Analog input 3 circuit failure (not implemented - for further use)	2	6	40	0...3	0
40	0x28	Analog input 4 circuit failure (not implemented - for further use)	2	7	41	0...3	0
41	0x29	Encoder channel a cable break	2	8	42	0...3	0
42	0x2A	Encoder channel b cable break	2	9	43	0...3	0
43	0x2B	Encoder channel z cable break	2	10	44	0...3	0
44	0x2C	SSI error	2	11	45	0...3	0
45	0x2D	Power driver (not implemented - for further use)	2	12	46	127	127
46	0x2E	Internal random access memory (not implemented - for further use)	2	13	47	127	127
47	0x2F	Internal program memory (not implemented - for further use)	2	14	48	127	127
48	0x30	Internal nonvolatile memory	2	15	49	127	127
49	0x31	Out of memory error (not implemented - for further use)	2	16	50	0...3	2
50	0x32	Software coding	2	17	51	0...3	2
51	0x33	Software reset (watchdog) occurred	2	18	52	0...3	2
52	0x34	Interrupt time exceeded	2	19	53	0...3	2
53	0x35	Task time exceeded	2	20	54	0...3	2
54	0x36	Parameter initialization error	2	21	55	0...3	2
55	0x37	Node identifier data memory corrupted	2	22	56	0...3	2
56	0x38	User data memory corrupted	2	23	57	0...3	2
57	0x39	Restore data memory corrupted (not implemented - for further use)	2	24	58	127	127
58	0x3A	Factory data memory corrupted	2	25	59	127	127
59	0x3B	Calibration data memory corrupted (not implemented - for further use)	2	26	60	127	127
60	0x3C	Diagnosis data memory corrupted	2	27	61	0...3	0
61	0x3D	Position control monitoring	2	28	62	0...3	0
62	0x3E	Velocity control monitoring	2	29	63	0...3	0
63	0x3F	Force control monitoring (not implemented - for further use)	2	30	64	0...3	0
64	0x40	Flow control monitoring (not implemented - for further use)	2	31	65	0...3	0
65	0x41	Pressure control monitoring	3	0	66	0...3	0
66	0x42	Current control monitoring	3	1	67	0...3	0
67	0x43	Spool position control monitoring	3	2	68	0...3	2
68	0x44	Trajectory generator processing error (not implemented - for further use)	3	3	69	0...3	0
69	0x45	Eventhandler exception	3	4	70	0...3	0
70	0x46	Local CAN general fault (not implemented - for further use)	3	5	71	0...3	0
71	0x47	Local CAN buffer overflow (not implemented - for further use)	3	6	72	0...3	0
72	0x48	Local CAN in error passive mode (not implemented - for further use)	3	7	73	0...3	0
73	0x49	Local CAN recovered from bus-off (not implemented - for further use)	3	8	74	0...3	0
74	0x4A	Local CAN RPD01 time out	3	9	75	0...3	0
75	0x4B	Local CAN RPD02 time out	3	10	76	0...3	0

Table 94: Possible fault codes (part 2 of 4)

Fault code			Dword		<FaultReactionType> (72#101…219)		
Dec.	Hex.	Fault description	Num	Bit	Index – 100	Value range	De-fault
76	0x4C	Local CAN RPD03 time out	3	11	77	0...3	0
77	0x4D	Local CAN RPD04time out	3	12	78	0...3	0
78	0x4E	Local CAN RPD01 data	3	13	79	0...3	0
79	0x4F	Local CAN RPD02 data	3	14	80	0...3	0
80	0x50	Local CAN RPD03 data	3	15	81	0...3	0
81	0x51	Local CAN RPD04 data	3	16	82	0...3	0
82	0x52	Local CAN TPD01 time out	3	17	83	0...3	0
83	0x53	Local CAN TPD02 time out	3	18	84	0...3	0
84	0x54	Local CAN TPD03 time out	3	19	85	0...3	0
85	0x55	Local CAN TPD04 time out	3	20	86	0...3	0
86	0x56	Local CAN TPD01 data	3	21	87	0...3	0
87	0x57	Local CAN TPD02 data	3	22	88	0...3	0
88	0x58	Local CAN TPD03 data	3	23	89	0...3	0
89	0x59	Local CAN TPD04 data	3	24	90	0...3	0
90	0x5A	CAN general fault	3	25	91	0...3	0
91	0x5B	CAN buffer overflow (not implemented - for further use)	3	26	92	0...3	0
92	0x5C	CAN in error passive mode (not implemented - for further use)	3	27	93	0...3	0
93	0x5D	CAN recovered from bus-off (not implemented - for further use)	3	28	94	0...3	0
94	0x5E	CAN RPD01 time out	3	29	95	0...3	0
95	0x5F	CAN RPD02 time out	3	30	96	0...3	0
96	0x60	CAN RPD03 time out	3	31	97	0...3	0
97	0x61	CAN RPD04 time out	4	0	98	0...3	0
98	0x62	CAN RPD01 data	4	1	99	0...3	0
99	0x63	CAN RPD02 data	4	2	100	0...3	0
100	0x64	CAN RPD03 data	4	3	101	0...3	0
101	0x65	CAN RPD04 data	4	4	102	0...3	0
102	0x66	CAN TPD01 time out	4	5	103	0...3	0
103	0x67	CAN TPD02 time out	4	6	104	0...3	0
104	0x68	CAN TPD03 time out	4	7	105	0...3	0
105	0x69	CAN TPD04 time out	4	8	106	0...3	0
106	0x6A	CAN TPD01 data	4	9	107	0...3	0
107	0x6B	CAN TPD02 data	4	10	108	0...3	0
108	0x6C	CAN TPD03 data	4	11	109	0...3	0
109	0x6D	CAN TPD04 data	4	12	110	0...3	0
110	0x6E	CAN life guard error or heartbeat error	4	13	111	0...3	0
111	0x6F	CAN SYNC producer time out	4	14	112	0...3	0
112	0x70	CAN SYNC consumer time out	4	15	113	0...3	0
113	0x71	EtherCAT communication fault	4	16	114	0...3	0
114	0x72	EtherCAT RPDO time out	4	17	115	0...3	1
115	0x73	EtherCAT RPDO data	4	18	116	0...3	0
116	0x74	EtherCAT TPDO time out	4	19	117	0...3	0
117	0x75	EtherCAT TPDO data	4	20	118	0...3	0
118	0x76	PROFIBUS general fault	4	21	119	0...3	0
119	0x77	I2C_general_fault	4	22	120	0...3	1
120	0x78	Reserved	4	23	121	unused	unused

Table 94: Possible fault codes (part 3 of 4)

Fault code			Dword		<FaultReactionType> (72#101…219)		
Dec.	Hex.	Fault description	Num	Bit	Index – 100	Value range	De-fault
121	0x79	Reserved	4	24	122	unused	unused
122	0x7A	Reserved	4	25	123	unused	unused
123	0x7B	Reserved	4	26	124	unused	unused
124	0x7C	Reserved	4	27	125	unused	unused
125	0x7D	Reserved	4	28	126	unused	unused
126	0x7E	Reserved	4	29	127	unused	unused
127	0x7F	Reserved	4	30	128	unused	unused
128	0x80	Reserved	4	31	129	unused	unused

Table 94: Possible fault codes (part 4 of 4)

8.1.3 Fault reaction type

The fault reaction parameter <FaultReactionType> (72#101…219) can be used to configure the fault behavior for each fault code.

8.1.3.1 Object 72#101…219: Fault reaction type

The fault reaction for each fault event can be configured by selecting different fault reaction types 0 to 3. The reaction 127 (non-removable error) is predefined. This fault reaction type cannot be changed.

To get the index of the fault reaction type <FaultReactionType> (72#101…219), the fault code must be incremented by 101.

FaultReaction							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#101…219	FaultReactionType	0x2830#1…119	INT8	rw	Y	⇨ Table 94, page 258	

Value description

<FaultReactionType>	Description
0 (NONE)	No fault reaction, error is ignored. Independent whether a malfunction for the monitored fault is detected, the device does not react on this event. The device continues to operate. Special care must be taken, as the malfunction may have an impact on the device.
1 (EMERGENCY)	Send an emergency message. If a malfunction for the monitored fault is detected, an emergency message will be sent onto the field bus. The device continues to operate. Special care must be taken, as the malfunction may have an impact on the device.
2 (FAULT_DISABLED)	Switch to device state 'FAULT_DISABLED'. If a malfunction for the monitored fault is detected, the device state machine enters the 'FAULT_DISABLED' state and an emergency message will be sent onto the field bus. The power stage of the device is switched off, while all device functions are still alive. The device must be re-enabled with the control word or the enable signal in order to return into normal operation.
3 (FAULT_HOLD)	Switch to device state 'FAULT_HOLD'. If a malfunction for the monitored fault is detected, the device enters the 'FAULT_HOLD' state and an emergency message will be sent onto the field bus. The hold setpoint of the device is in effect and adjusted. The device must be re-enabled with the control word or the enable signal in order to return into normal operation.
127 (FAULT STOP)	Switch to device state 'NOT_READY'. If a malfunction for the monitored fault is detected, the device enters the 'NOT_READY' state and an emergency message will be sent onto the field bus. The power stage of the device is switched off, while almost all device functions are stopped. The device must be serviced.
Otherwise	Reserved

Table 95: Fault reaction settings

8.1.4 Error codes depending on fault codes

The following table combines the Moog specific fault codes with the error codes, send with an emergency message.

Details about the error codes used for monitoring: ⇨ Chapter "7.15 Monitoring", page 222

Fault code	Error code defined in Device Profile Fluid Power	Error description
11, 12	2210	Internal current too high or low
16	2211	Over current in external sensor supply
8, 10	3210	Internal voltage too high
7, 9	3220	Internal voltage too low
6	3411	Power supply voltage too high
5	3412	Power supply voltage too low
14, 15	4211	Temperature of electronic components too high
13	4212	Temperature of electronic components too low
24	5211	Internal pressure transducer circuit failure
17, 18, 19	5212	Pilot/single stage actual spool position
36	5213	Analog input 0 circuit failure
37	5214	Analog input 1 circuit failure
38	5215	Analog input 2 circuit failure
39	5216	Analog input 3 circuit failure
40	5217	Analog input 4 circuit failure
22	5218	Main stage actual spool position sensor
1, 2, 3, 4	5220	Microprocessor core
23, 25	5231	Internal pressure transducer cable break / pressure peak

Table 96: Possible error codes depending on fault codes (part 1 of 3)

Fault code	Error code defined in Device Profile Fluid Power	Error description
41, 42, 43, 44	5232	Encoder cable break / SSI error
26, 31	5233	Analog input 0: supply cable error / signal out of range
27, 32	5234	Analog input 1: supply cable error / signal out of range
28, 33	5235	Analog input 2: supply cable error / signal out of range
29, 34	5236	Analog input 3: supply cable error / signal out of range
30, 35	5237	Analog input 4: supply cable error / signal out of range
20, 21	5238	Main stage actual spool position out of range / cable break
45	5410	Power driver
46	5510	RAM
47	5520	EPROM
48	5530	EEPROM
51	6010	Software reset (watchdog)
50	6101	Software coding
52	6102	Interrupt time exceeded
53	6103	Task time exceeded
49	6104	Out of memory
69	6201	Event handler
55	6311	Node identifier data
56	6312	User data
57	6313	Restore data
58	6314	Factory data
59	6315	Calibration data
60	6316	Diagnosis data
54	6320	Parameter error
119	7002	I2C_general_fault
90, 113, 118	8100	Fieldbus communication
70	8101	Local CAN communication
91	8110	CAN overrun
71	8111	Local CAN overrun (objects lost)
92	8120	CAN in error passive mode
72	8121	Local CAN in error passive mode
110	8130	Life guard error or heartbeat error
93	8140	CAN recovered from bus off
73	8141	Local CAN recovered from bus off
94, 114	8231	RPD01 time out
95	8232	RPD02 time out
96	8233	RPD03 time out
97	8234	RPD04 time out
102, 116	8235	TPD01 time out
103	8236	TPD02 time out
104	8237	TPD03 time out
105	8238	TPD04 time out
111	8239	SYNC producer time out
112	823A	SYNC consumer time out
98, 115	8241	RPD01 data
99	8242	RPD02 data
100	8243	RPD03 data

Table 96: Possible error codes depending on fault codes (part 2 of 3)

Fault code	Error code defined in Device Profile Fluid Power	Error description
101	8244	RPD04 data
106, 117	8245	TPD01 data
107	8246	TPD02 data
108	8247	TPD03 data
109	8248	TPD04 data
74	8251	Local RPD01 time out
75	8252	Local RPD02 time out
76	8253	Local RPD03 time out
77	8254	Local RPD04 time out
82	8255	Local TPD01 time out
83	8256	Local TPD02 time out
84	8257	Local TPD03 time out
85	8258	Local TPD04 time out
78	8261	Local RPD01 data
79	8262	Local RPD02 data
80	8263	Local RPD03 data
81	8264	Local RPD04data
86	8265	Local TPD01 data
87	8266	Local TPD02 data
88	8267	Local TPD03 data
89	8268	Local TPD04 data
67	8301	Position control monitoring
65	8302	Pressure control monitoring
61	8303	Position control monitoring
62	8304	Velocity control monitoring
63	8305	Force control monitoring
64	8306	Flow control monitoring
66	8307	Current control monitoring
68	8308	Trajectory generation

Table 96: Possible error codes depending on fault codes (part 3 of 3)

8.1.5 Fault status

The bit coded fault status indicates which faults are currently reported for the device. Each bit of the fault status array (4 words with 32 bit, built with the sub-indexes 1…4 of the fault status) stands for a fault code. The corresponding fault code is equal to the bit position in the 128 bit field (4x32 bit).

8.1.5.1 Object 72#26…29: Fault status

Actual reported faults in bit coded form.

FaultReaction							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#26	FaultStatus (fault code 1…31)	0x2831#1	UINT32	ro	-	UINT32	None
72#27	FaultStatus (fault code 32…63)	0x2831#2	UINT32	ro	-	UINT32	None
72#28	FaultStatus (fault code 64…95)	0x2831#3	UINT32	ro	-	UINT32	None
72#29	FaultStatus (fault code 96…128)	0x2831#4	UINT32	ro	-	UINT32	None

8.1.5.2 Object 72#42…45: Fault retain status

All reported faults in bit coded form since powering on the servo valve.

FaultReaction							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#42	FaultRetainStatus (fault code 1…31)	0x2834#1	UINT32	rw	N	UINT32	None
72#43	FaultRetainStatus (fault code 32…63)	0x2834#2	UINT32	rw	N	UINT32	None
72#44	FaultRetainStatus (fault code 64…95)	0x2834#3	UINT32	rw	N	UINT32	None
72#45	FaultRetainStatus (fault code 96…128)	0x2834#4	UINT32	rw	N	UINT32	None

8.1.6 Error register

The <ErrorRegister> (64#53) displays the error information about the last reported fault in bit-coded form. Bit 0 of the <ErrorRegister> (64#53) is set as soon as an error occurs on the servo valve.

The error codes of older faults are stored in the <PredefinedErrorField> (64#35…51).
⇨ Chapter "8.1.7 Last eight fault codes and error codes", page 266

8.1.6.1 Object 64#53: Error register

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#53	ErrorRegister	0x1001#0	UINT8	ro	-	UINT8	None

This object shall provide error information. The CANopen device maps internal errors into this object. This information is also sent with the emergency object.

Value description

<ErrorRegister>	**Description**
Bit	
0	Generic error (any error)
1	Current error
2	Voltage error
3	Temperature error
4	Communication error
5	Reserved
6	Reserved
7	Reserved

Table 97: Possible values of parameter <ErrorRegister> (64#53)

If a specific error occurs, the corresponding bit shall be set to 1_b. The bits are cleared automatically when the error has gone.

8.1.7 Last eight fault codes and error codes

Every time, a fault is thrown which triggered a fault reaction, information about the fault is stored to the <StandardErrorField> (64#35…51) parameter array. The <StandardErrorField> (64#35…51) parameter array contains a list of up to 8 entries. This error code provides information about the reason of the error. The parameter <NumberOfErrors> (64#35…51) holds information about the number of errors currently recorded. Every new error is stored in the first element of the parameter array <StandardErrorField> (64#35…51), the older ones move down in the list. If the maximum number of entries is reached and a new fault occurred the oldest fault information will be deleted.

Writing the value 0 to the object <NumberOfErrors> (64#35…51), parameter 64#35, deletes the entire error code entries.

ⓘ Only the first eight elements of the parameter array <StandardErrorField> (64#36…51), 36…43 are used.

8.1.7.1 Object 64#35…51: Predefined error field

This object contains the last eight error codes, fault codes and the number of recorded errors.

Device							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#35	NumberOfErrors	0x1003#0	UINT32	rw	N	UINT32	0
64#36…51	StandardErrorField	0x1003#1…16	UINT32	ro	-	UINT32	0

Value description

Parameter	Description
<NumberOfErrors>	Number of actual recorded errors.
<StandardErrorField>	Array of recorded errors.

Table 98: Possible values of parameter <PreDefinedErrorField> (64#35…51)

<StandardErrorField>				
Byte	3	2	1	0
Description	Additional information		Error code ⇨ Chapter "8.1.4 Error codes depending on fault codes", page 262	
	Reserved	Fault code ⇨ Chapter "8.1.2 Possible fault codes", page 258		

Example

The parameter <StandardErrorField> (64#36) holds the decimal value 3167536 (corresponds to 0x305530 hex).
The coding of the value is shown in the following table:

<StandardErrorField>				
Byte	3	2	1	0
Description	Additional information		Error code	
	Reserved	Fault code		
Content	0x00	0x30	0x5530	

Result:

Error code 0x5530: EEPROM error
Fault code 0x30: Internal non-volatile memory

8.1.8 Last eight error message descriptions

The last eight error description strings can be read by the parameter <FaultReactionDescription> (72#40). The parameter <FaultHistoryNumber> (72#41) selects one of the last eight error description strings. The newest error description string is shown if the <FaultHistoryNumber> (72#41) is set to zero and the oldest saved error description string is shown if the <FaultHistoryNumber> (72#41) is set equal to the parameter <NumberOfErrors> (64#35…51).

8.1.8.1 Object 72#40: Fault reaction description

The parameter <FaultReactionDescription> (72#40) contains the fault reaction description string depending on the <FaultHistoryNumber> (72#41) including the fault time in minutes since production of the servo valve. The format is "A/B @ M min STRING". A = displayed fault number, B = count of faults, M = fault time, STRING = description.

FaultReaction							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#40	FaultReactionDescription	0x2832#0	STRING	ro	-	None	None

8.1.8.2 Object 72#41: Fault history number

The parameter <FaultHistoryNumber> (72#41) selects the fault description shown in the parameter <FaultReactionDescription> (72#40).

FaultReaction							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#41	FaultHistoryNumber	0x2833#0	UINT8	rw	N	0…7	None

8.1.9 Diagnostic message

Every time a configured error occurs on the servo valve, it sends a diagnostic message with error register, error code, fault code and timestamp to the Profibus DP master. The emergency diagnostic message will also be sent if all errors has disappeared. In this case the fault code 0x00 (Error reset or no error) will be sent.

➪ Chapter "8.1.4 Error codes depending on fault codes", page 262
➪ Chapter "8.1.6 Error register", page 265
➪ Chapter "2.10 Diagnostic (DP-V0)", page 27

8.1.10 Fault disappears

If all faults has disappeared the <ErrorRegister> (64#53) and the <FaultStatus> (72#26…29) are set to zero. To confirm that no faults are present, the error code 0x00 (Error reset or no error) will be sent via an emergency message to the field bus master. If the device state machine (DSM) is in the state 'FAULT_HOLD', 'FAULT_DISABLED' or 'NOT_READY', the DSM must be set to 'ACTIVE' again. This can be done by the #ControlWord# or the enable signal (digital input 0).

➪ Chapter "8.1.11 Fault acknowledgment", page 268

8.1.11 Fault acknowledgment

Depending on the configured fault reaction, the servo valve sends out an emergency message and changes the device state machine to the corresponding fault state.

⇨ Chapter "8.1.3 Fault reaction type", page 261

In order to get out of the fault state the fault must be acknowledged. This can be achieved by

- Sending the #ControlWord# to the servo valve with the bit 3 (fault reset) is set.
- Toggling the enable signal (digital input 0).

⇨ Chapter "5.1 Local mode", page 39
⇨ Chapter "5.2 Device state machine (DSM)", page 42
⇨ Chapter "6.8.2 Object 0#220…221: Digital output configuration", page 94

If the fault is not fixed or other faults are still present, the servo valve will fall back into the DSM state defined by the fault reaction type <FaultReactionType> (72#102…219).

8.2 Internal errors

The following parameters store information which could assist to debug software malfunctions. The user may be asked to pass these values to our service personal in order to identify software malfunctions.

8.2.1 Object 72#13…17: Internal error code

This object contains the first five occurred internal error codes since the firmware reset.

ErrorHandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#13…17	InternalErrorCode	0x2822#1…5	UINT32	ro	-	UINT32	0

8.2.2 Object 72#18…22: Internal error time

The time stamp of the occurrence of the last five error codes were saved in this array. The time stamps were stored in minutes since power on.

ErrorHandler							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#18…22	InternalErrorTime	0x2823#1…5	UINT32	ro	-	UINT32	0

8.3 Abort SDO Transfer Protocol

The Service Data Object (SDO) is used to transmit / receive parameter values to / from the valve. If the SDO upload or download is not successful, the servo valve will send an Abort SDO Transfer Protocol. The error description is coded in the data bytes.

SDO Abort Code	Description
0x05000000	General SDO protocol error detected.
0x05030000	Toggle bit not alternated.
0x05040000	SDO protocol timeout.
0x05040001	Client/server command specifier not valid or unknown.
0x05040002	Invalid block size (block mode only).
0x05040003	Invalid sequence (block mode only).
0x05040004	CRC error (block mode only).
0x05040005	Out of memory.
0x06010000	Unsupported access to an object.
0x06010001	Attempt to read a write only object.
0x06010002	Attempt to write a read only object.
0x06020000	Object does not exist in the object dictionary.
0x06040041	Object cannot be mapped PDO.
0x06040042	The number and length of the objects to be mapped would exceed PDO length.
0x06040043	General parameter incompatibility reason.
0x06040047	General internal incompatibility in the device.
0x06060000	Access failed due to hardware error.
0x06070010	Data type / length of service data does not match.
0x06070012	Data type does not match / length of service data too high.
0x06070013	Data type does not match / length of service data too low.
0x06090011	Sub index doesn't exist.
0x06090030	Invalid value for parameter (download only).
0x06090031	Value of parameter written too high (download only).
0x06090032	Value of parameter written too low (download only).
0x06090036	Maximum value is less than minimum value.
0x08000000	General error.
0x08000020	Data cannot be transferred or stored to the application.
0x08000021	Data cannot be transferred or stored to the application because of local control.
0x08000022	Data cannot be transferred or stored to the application because of the present device state.
0x08000023	Object dictionary dynamic generation fails or no object dictionary is present.
0x08000024	No data available.

Table 99: SDO Abort Codes

8.4 Troubleshooting

8.4.1 Fault code descriptions

8.4.1.1 Not implemented fault

Error Code:	Various
Error Name:	Not implemented fault
Description:	Fault is not implemented until now. This fault is a placeholder and the implementation is planned for further version.
Severity of error:	None
Action:	None

A fault reaction may be configured, but won't have any impact on the valve. The fault never will be thrown.

8.4.1.2 0x02: Error during startup

Error Code:	0x02
Error Name:	Error during startup
Description:	Internal error during startup
Severity of error:	Major
Action:	Send valve back to factory service

8.4.1.3 0x03: Error DSP program download

Error Code:	0x03
Error Name:	Error DSP program download
Description:	Internal error during startup
Severity of error:	Major
Action:	Send valve back to factory service

8.4.1.4 0x04: Error DSP realtime data transmission

Error Code:	0x04
Error Name:	Error DSP realtime data transmission
Description:	Internal communication error during runtime
Severity of error:	Major
Action:	Send valve back to factory service

8.4.1.5 0x05: Power supply voltage too low

Error Code:	0x05
Error Name:	Power supply voltage too low
Description:	Power supply voltage exceeds lower limit of 18 V
Severity of error:	Minor
Action:	Check power supply

8.4.1.6 0x06: Power supply voltage too high

Error Code:	0x06
Error Name:	Power supply voltage too high
Description:	Power supply voltage exceeds upper limit of 32 V
Severity of error:	Minor
Action:	Check power supply

8.4.1.7 0x07: Internal supply voltage too low

Error Code:	0x07
Error Name:	Internal supply voltage too low
Description:	Internal power supply for the processor is too low
Severity of error:	Major
Action:	Send valve back to factory service

8.4.1.8 0x08: Internal supply voltage too high

Error Code:	0x08
Error Name:	Internal supply voltage too high
Description:	Internal power supply for the processor is too high
Severity of error:	Major
Action:	Send valve back to factory service

8.4.1.9 0x0D: Electronics temperature too low (< –20 °C)

Error Code:	0x0D
Error Name:	Electronics temperature too low (< –20 °C)
Description:	Temperature of the electronic is less than –20 °C
Severity of error:	Minor
Action:	• Check environmental temperature • Consider additional protection cover or heating

Respect temperature limit (–20 °C to +80 °C).

8.4.1.10 0x0E: Electronics temperature too high (> 85 °C)

Error Code:	0x0E
Error Name:	Electronics temperature too high (> 85 °C)
Description:	Temperature of the electronic is greater than 85° C
Severity of error:	Minor
Action:	• Check environmental temperature • Consider additional protection cover or cooling

Respect temperature limit (–20 °C to +80 °C).

8.4.1.11 0x0F: Electronics temperature exceeded (> 105 °C)

Error Code:	0x0F
Error Name:	Electronics temperature exceeded (> 105 °C)
Description:	Temperature of the electronic is greater than 105 °C
Severity of error:	Medium
Action:	• Check environmental temperature • Check for external heat sources • Consider additional protection cover or cooling

Long-term impact of high temperature onto the electronics reduce lifetime significantly.

8.4.1.12 0x11: Pilot/single stage LVDT cable break

Error Code:	0x11
Error Name:	Pilot/single stage LVDT cable break
Description:	Cable break on the internal LVDT detected
Severity of error:	Major
Action:	Send valve back to factory for service

In order to make a customer configured fault reaction working, a specific factory setting of the parameter "lvdfautyp" is necessary.

8.4.1.13 0x14: Main stage LVDT cable break

Error Code:	0x14
Error Name:	Main stage LVDT cable break
Description:	Cable break on the external LVDT detected
Severity of error:	Major
Action:	• Check cable connection between pilot valve and main stage (for example, loosen connector). If the error still occurs after restarting the valve, send the valve back to factory for service.

8.4.1.14 0x1C: Analog input 2 supply cable break/short circuit

Error Code:	0x1C
Error Name:	Analog input 2 supply cable break/short circuit
Description:	Supply voltage of analog input 2 (X5) is monitored
Severity of error:	Minor
Action:	• Check cable and cable connection • Check input configuration and sensor adjustments

8.4.1.15 0x1D: Analog input 3 supply cable break/short circuit

Error Code:	0x1D
Error Name:	Analog input 3 supply cable break/short circuit
Description:	Supply voltage of analog input 3 (X6) is monitored
Severity of error:	Minor
Action:	• Check cable and cable connection • Check input configuration and sensor adjustments

8.4.1.16 0x1E: Analog input 4 supply cable break/short circuit

Error Code:	0x1E
Error Name:	Analog input 4 supply cable break/short circuit
Description:	Supply voltage of analog input 4 (X7) is monitored
Severity of error:	Minor
Action:	• Check cable and cable connection • Check input configuration and sensor adjustments

8.4.1.17 0x1F…0x23: Analog input 0…4 current too low (4…20 mA) / ADC overflow (voltage)

Error Code:	0x1F…0x23
Error Name:	Analog input 0…4 current too low (4…20 mA) / ADC overflow (voltage)
Description:	Measured current is below the adjusted limit (anamonlow)
Severity of error:	Minor
Action:	• Check cable and cable connection • Check input configuration and sensor adjustments • Check parameter setting (anamonlow)

8.4.1.18 0x29: Encoder channel a cable break

Error Code:	0x29
Error Name:	Encoder channel a cable break
Description:	Cable break is detected on encoder channel a / SSI data line
Severity of error:	Minor
Action:	• Check cable and cable connection • Check PIN assignment • Check encoder/SSI power supply • Check functionality

8.4.1.19 0x2A: Encoder channel b cable break

Error Code: 0x2A

Error Name: Encoder channel b cable break

Description: Cable break is detected on encoder channel b

Severity of error: Minor

Action:
- Check cable and cable connection
- Check PIN assignment
- Check encoder/SSI power supply
- Check functionality

If a SSI sensor is used, this wire is not connected to the sensor. Hence, the cable break fault on channel b will always be active.

8.4.1.20 0x2B: Encoder channel z cable break

Error Code: 0x2B

Error Name: Encoder channel z cable break

Description: Cable break is detected on encoder channel z / SSI data line

Severity of error: Minor

Action:
- Check cable and cable connection
- Check PIN assignment
- Check encoder/SSI power supply
- Check functionality

8.4.1.21 0x2C: SSI error

Error Code: 0x2C

Error Name: SSI error

Description: SSI error is detected

Severity of error: Minor

Action:
- Check cable and cable connection
- Check PIN assignment
- Check encoder/SSI power supply
- Check sensor configuration (frequency and bit size)

8.4.1.22 0x30: Internal nonvolatile memory initialization error

Error Code: 0x30

Error Name: Internal nonvolatile memory initialization error

Description: An internal error during EEPROM initialization / database was detected

Severity of error: Major

Action:
- Restart the valve

If the error still occurs after restarting the valve, send the valve back to factory for service.

8.4.1.23 0x32: Software coding

Error Code:	0x32
Error Name:	Software coding
Description:	An internal software error (deadlock, illegal code operation) caused an restart of the valve
Severity of error:	Major
Action:	Send valve back to factory for service.

8.4.1.24 0x33: Software reset (watchdog) occurred

Error Code:	0x33
Error Name:	Software reset (watchdog) occurred
Description:	-
Severity of error:	Major
Action:	-

8.4.1.25 0x34: Interrupt time exceeded

Error Code:	0x34
Error Name:	Interrupt time exceeded
Description:	The internal interrupt task time has exceeded. The number of tasks and the time to calculate them exceeds the time limitation of the so-called interrupt. Not all interrupt tasks were calculated.
Severity of error:	Major
Action:	• Check the task time • Reduce the number of tasks • Reduce interface usage • Change control mode If the error occurs often or on a daily basis, even after reevaluation of all tasks, send the valve back to factory for service.

Do not store your application in state 'HOLD' or 'ACTIVE'.

8.4.1.26 0x35: Task time exceeded

Error Code:	0x35
Error Name:	Task time exceeded
Description:	The internal task time has exceeded. The number of tasks and the time to calculate them exceeds the general time limitation. Not all tasks were calculated.
Severity of error:	Medium
Action:	• Check the task time • Reduce the number of tasks (for example data logger, event handler) • Reduce interface usage • Change control mode If the error occurs often or on a daily basis, even after reevaluation of all tasks, send the valve back to factory for service.

Do not store your application in state 'HOLD' or 'ACTIVE'.

8.4.1.27 0x36: Parameter initialization error

Error Code:	0x36
Error Name:	Parameter initialization error
Description:	Internal error during initialization of RAM parameter settings occurred
Severity of error:	Major
Action:	• Restart the valve If the error still occurs after restarting the valve, send the valve back to factory for service.

8.4.1.28 0x37: Node identifier data memory corrupted

Error Code:	0x37
Error Name:	Node identifier data memory corrupted
Description:	Internal error during initialization of extended parameter settings occurred
Severity of error:	Major
Action:	• Restart the valve If the error still occurs after restarting the valve, send the valve back to factory for service.

8.4.1.29 0x38: User data memory corrupted

Error Code:	0x38
Error Name:	User data memory corrupted
Description:	Internal error during initialization of customer parameter settings occurred
Severity of error:	Major
Action:	• Restart the valve If the error still occurs after restarting the valve, send the valve back to factory for service.

8.4.1.30 0x3A: Factory data memory corrupted

Error Code: 0x3A

Error Name: Factory data memory corrupted

Description: Internal error during initialization of factory parameter settings occurred

Severity of error: Major

Action:
- Restart the valve

If the error still occurs after restarting the valve, send the valve back to factory for service.

8.4.1.31 0x3C: Diagnosis data memory corrupted

Error Code: 0x3C

Error Name: Diagnosis data memory corrupted

Description: Internal error during initialization of diagnose parameter settings occurred

Severity of error: Minor

Action:
- Restart the valve

If the error still occurs after restarting the valve, send the valve back to factory for service.

8.4.1.32 0x3D: Position control monitoring

Error Code: 0x3D

Error Name: Position control monitoring

Description: An axis position control monitoring fault is detected. The current axis position control deviation exceeds the adjusted limits.

Severity of error: Application specific

Action:
- Check adjusted deviation limits
- Check position
- Check sensor adjustments and settings
- Check interface configuration

This fault reaction is available for Axis Control Valves (ACV) only.

In order to make a customer configured fault reaction working, the axis position monitoring must be switched on.
⇨ Chapter "7.15.3 Axis position control deviation monitoring", page 227

8.4.1.33 0x3E: Velocity control monitoring

Error Code:	0x3E
Error Name:	Velocity control monitoring
Description:	An axis velocity control monitoring fault is detected. The current axis velocity deviation exceeds the adjusted limits.
Severity of error:	Application specific
Action:	• Check adjusted deviation limits • Check sensor adjustments and settings • Check interface configuration

This fault reaction is available for Axis Control Valves (ACV) only.

In order to make a customer configured fault reaction working, the axis velocity monitoring must be switched on.
⇨ Chapter "7.15.4 Axis velocity control deviation monitoring", page 229

8.4.1.34 0x41: Pressure control monitoring

Error Code:	0x41
Error Name:	Pressure control monitoring
Description:	A pressure control monitoring fault is detected. The current pressure deviation exceeds the adjusted limits.
Severity of error:	Application specific
Action:	• Check adjusted deviation limits • Check pressure • Check sensor adjustments and settings • Check interface configuration

In order to make a customer configured fault reaction working, the pressure monitoring must be switched on.
⇨ Chapter "7.15.2 Pressure control deviation monitoring", page 225

8.4.1.35 0x42: Current control monitoring

Error Code:	0x42
Error Name:	Current control monitoring
Description:	A current control monitoring fault is detected. Most likely due to a faulty hardware.
Severity of error:	Major
Action:	• Restart the valve If the error still occurs after restarting the valve, send the valve back to factory for service.

8.4.1.36 0x43: Spool position control monitoring

Error Code: 0x43

Error Name: Spool position control monitoring

Description: A spool position control monitoring fault is detected. The actual spool position control deviation exceeds the adjusted limits (for example, due to particles or pollution).

Severity of error: Application specific

Action:
- Check adjusted deviation limits
- Flush valve with clean oil at low pressure
- Switch to open loop operation and move spool end stop to end stop
- Shake the valve at low pressure with sweeping sine and/or square signal

For Pilot operated valves:
- Check pressure levels

For Pilot operated valve with external pilot pressure:
- Check external pilot pressure level
- Check external pilot pressure availability

In order to make a customer configured fault reaction working, the position monitoring must be switched on.
On two stage valves, the main stage is monitored.

⇨ Chapter "7.15.1 Spool position control deviation monitoring", page 223

8.4.1.37 0x45: Event handler exception

Error Code: 0x45

Error Name: Event handler exception

Description: A general event handler fault is detected.

Severity of error: Minor

Action:
- Check the event handler expression for illegal operations
- Clear all event handler expressions
- Restart the valve

If the error still occurs after restarting the valve, send the valve back to factory for service.

8.4.1.38 0x4A…0x4D: Local CAN RPDO1…RPDO4 time out

Error Code: 0x4A…0x4D

Error Name: Local CAN RPDO1…RPDO4 time out

Description: A timeout on Local CAN Receive PDO 1…4 occurred.

⇨ Chapter "6.9 Local CAN", page 95

Severity of error: Minor

Action:
- Check cable and cable connection
- Check Local CAN communication interface settings
- Check Local CAN RPDO timeout settings

8.4.1.39 0x4E…0x51: Local CAN RPDO1…RPDO4 data

Error Code: 0x4E…0x51

Error Name: Local CAN RPDO1…RPDO4 time out

Description: A problem with the parametrization on Local CAN Receive PDO 1…4 was detected.

➩ Chapter "6.9 Local CAN", page 95

Severity of error: Minor

Action:
- Check cable and cable connection
- Check Local CAN communication interface settings

8.4.1.40 0x52…0x55: Local CAN TPDO1…4 time out

Error Code: 0x52…0x55

Error Name: Local CAN TPDO1…4 time out

Description: A timeout on Local CAN Transmit PDO 1…4 occurred.

➩ Chapter "6.9 Local CAN", page 95

Severity of error: Minor

Action:
- Check cable and cable connection
- Check Local CAN communication interface settings
- Check Local CAN TPDO timeout settings

8.4.1.41 0x56…0x59: Local CAN TPDO1…4 data

Error Code: 0x56…0x59

Error Name: Local CAN TPDO1…4 time out

Description: A problem with the parametrization on Local CAN Transmit PDO 1…4 was detected.

➩ Chapter "6.9 Local CAN", page 95

Severity of error: Minor

Action:
- Check cable and cable connection
- Check Local CAN communication interface settings

8.4.1.42 0x5B: CAN general fault

Error Code: 0x5B

Error Name: CAN general fault

Description: A general problem in the CAN fieldbus initialization or communication was detected.

A problem in the Network State Machine occurred.

Severity of error: Application specific

Action:
- Check cable and cable connection
- Check CAN state machine
- Check CAN communication interface settings
- Restore all communication parameters to factory settings

➩ Chapter "9.2 Restoring parameters to factory settings", page 287

If the error still occurs after restarting the valve, send the valve back to factory for service.

8.4.1.43 0x5E…0x61: CAN RPDO1…4 time out

Error Code:	0x5E…0x61
Error Name:	CAN RPDO1…4 time out
Description:	A timeout on CAN Receive PDO 1…4 occurred.
Severity of error:	Application specific
Action:	• Check cable and cable connection • Check CAN communication interface settings • Check CAN TPDO timeout settings

8.4.1.44 0x62…0x65: CAN RPDO1…4 data

Error Code:	0x62…0x65
Error Name:	CAN RPDO1…4 data
Description:	A problem with the parametrization on CAN Receive PDO 1…4 was detected.
Severity of error:	Application specific
Action:	• Check cable and cable connection • Check CAN communication interface settings

8.4.1.45 0x66…0x69: CAN TPDO1…4 time out

Error Code:	0x66…0x69
Error Name:	CAN TPDO1…4 time out
Description:	A timeout on CAN Transmit PDO 1…4 occurred.
Severity of error:	Application specific
Action:	• Check cable and cable connection • Check CAN communication interface settings • Check CAN TPDO timeout settings

8.4.1.46 0x6A…0x6D: CAN TPDO1…4 data

Error Code:	0x6A…0x6D
Error Name:	CAN TPDO1…4 data
Description:	A problem with the parametrization on CAN Transmit PDO 1…4 was detected.
Severity of error:	Application specific
Action:	• Check cable and cable connection • Check CAN communication interface settings

8.4.1.47 0x6E: CAN life guard error or heartbeat error

Error Code: 0x6E

Error Name: CAN life guard error or heartbeat error

Description: Periodical monitoring of the device with the Node Guarding protocol timed out. This happens when the guarding request has not been received by the device within the configured GuardTime and lifeTimeFactor.

Severity of error: Minor

Action:
- Check cable and cable connection
- Check CAN communication interface settings
- Check guarding settings
- Check NMT
- Check guard request from PLC (master)

Only guarding errors can be detected. Heartbeat error detection is not implemented yet.

8.4.1.48 0x6F: CAN SYNC producer time out

Error Code: 0x6F

Error Name: CAN SYNC producer time out

Description: A synchronization problem on CAN occurred. The synchronization signal could not be received within timeout.

Severity of error: Minor

Action:
- Check cable and cable connection
- Check CAN communication interface settings
- Check corresponding Receive PDO and Transmission type

8.4.1.49 0x70: CAN SYNC producer time out

Error Code: 0x70

Error Name: CAN SYNC consumer time out

Description: A synchronization problem on CAN occurred.

Severity of error: Minor

Action:
- Check cable and cable connection
- Check CAN communication interface settings
- Check corresponding Receive PDO and Transmission type

8.4.1.50 0x71: EtherCAT communication fault

Error Code: 0x71

Error Name: EtherCAT communication fault

Description: An internal error related to EtherCAT has occurred (e.g., internal EEPROM error, internal EtherCAT logic could not be accessed, etc.).

Severity of error: Major

Action:
- Restart the valve

If the error still occurs after restarting the valve, send the valve back to factory for service.

8.4.1.51 0x72: EtherCAT RPDO time out

Error Code: 0x72

Error Name: EtherCAT RPDO time out

Description: A timeout on EtherCAT Receive PDO occurred.

Severity of error: Application specific

Action:
- Check cable and cable connection
- Check EtherCAT communication interface settings

8.4.1.52 0x73: EtherCAT RPDO data

Error Code: 0x73

Error Name: EtherCAT RPDO data

Description: A problem with the parametrization on EtherCAT Receive PDO was detected.

Severity of error: Application specific

Action:
- Check cable and cable connection
- Check EtherCAT communication interface settings

8.4.1.53 0x74: EtherCAT TPDO time out

Error Code: 0x74

Error Name: EtherCAT TPDO time out

Description: A timeout on EtherCAT Transmit PDO occurred.

Severity of error: Application specific

Action:
- Check cable and cable connection
- Check EtherCAT communication interface settings

8.4.1.54 0x75: EtherCAT TPDO data

Error Code: 0x75

Error Name: EtherCAT TPDO data

Description: A problem with the parametrization on EtherCAT Transmit PDO was detected.

Severity of error: Application specific

Action:
- Check cable and cable connection
- Check EtherCAT communication interface settings

8.4.1.55 0x76: PROFIBUS general fault

Error Code: 0x76

Error Name: PROFIBUS general fault

Description: A problem with the Profibus communication occurred.

Severity of error: Application specific

Action:
- Check cable and cable connection
- Check Profibus communication interface settings

8.4.1.56 0x77: I2C_general_fault

Error Code: 0x77

Error Name: I2C_general_fault

Description: An error while reading the setting of the DIP switches occurred.

Severity of error: Major

Action:
- Check the DIP switches for correct positioning
- Restart the valve

If the error still occurs after restarting the valve, send the valve back to factory for service.

This fault reaction is available for valves with DIP switches only.

9 Storing / restoring parameters

The electronics of the servo valve provide a non-volatile memory which allows storing parameters. The current values of all parameters declared as non-volatile (persistence = "Y") can be stored in a non-volatile memory on the servo valve. Three storing / restoring operations are possible:

- Parameters can be stored in the non-volatile memory.
  ⇨ Chapter "9.1 Storing parameters", page 286
- Parameters are automatically restored while power on.
  The stored parameters are automatically reloaded during a power on cycle.
- Factory settings can be restored
  ⇨ Chapter "9.2 Restoring parameters to factory settings", page 287

Parameters are stored and restored in accordance with the procedure described in the Device Profile Fluid Power.

The following table describes the behavior of the savable and volatile parameters when performing a store parameters, servo valve bootup or restore parameters operation.

Operation	Non-volatile parameters (persistence = "Y")	Volatile parameters (persistence = "N")	
		Default value defined for the parameter	Default value = "None"
Bootup servo valve	Parameter values will be loaded.	Factory default values will be loaded.	No values will be loaded.
Store parameters	Parameter values will be saved.	No values will be saved.	No values will be saved.
Restoring factory settings	Factory values of the parameters will be loaded.	Factory default values will be loaded.	No values will be loaded.

Table 100: Behavior of saveable and volatile parameters

The write access to any parameter will not affect its default value.

9.1 Storing parameters

The current values of all parameters declared as non-volatile (persistence = "Y") can be stored in a non-volatile memory on the servo valve.

9.1.1 Object 0#51 / 0#212…214: Store parameters

Storing is proceeded by writing the signature 0x65766173 ("save") to one of the following parameters.

Byteorder for "save": byte[4] = 0x73 = 's' ; byte[5] = 0x61 = 'a' ; byte[6] = 0x76 = 'v' ; byte[7] = 0x65 = 'e'.

StoreParameters							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#51	SaveAllParameters	0x1010#1	UINT32	rw	N	UINT32	None
0#212	SaveCommunicationParameters	0x1010#2	UINT32	rw	N	UINT32	None
0#213	SaveApplicationParameters	0x1010#3	UINT32	rw	N	UINT32	None
0#214	SaveManufacturerDefinedParameters	0x1010#4	UINT32	rw	N	UINT32	None

Value description

Parameter	Description
<SaveAllParameters>	Saves all non-volatile parameters in the servo valve's non-volatile memory.
<SaveCommunicationParameters>	Saves all non-volatile communication parameters (index range 0x1000…0x1FFF) in the servo valve's non-volatile memory.
<SaveApplicationParameters>	Saves all non-volatile application parameters (index range 0x6000…0x9FFF) in the servo valve's non-volatile memory.
<SaveManufacturerDefinedParameters>	Saves all non-volatile manufacturer-defined parameters (index range 0x2000…0x5FFF) in the servo valve's non-volatile memory.

Table 101: Possible values of parameter 0#51 / 0#212…214

9.2 Restoring parameters to factory settings

The restore command sets the values of the non-volatile parameters to factory settings.

9.2.1 Object 0#52 / 0#215…217: Restore default parameters

The factory settings can be restored by writing the signature 0x64616F6C ("load") to one the following parameters.

StoreParameters							
Slot # Index	Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#52	RestoreAllDefaultParameters	0x1011#1	UINT32	rw	N	UINT32	None
0#215	RestoreCommunicationDefaultParameters	0x1011#2	UINT32	rw	N	UINT32	None
0#216	RestoreApplicationDefaultParameters	0x1011#3	UINT32	rw	N	UINT32	None
0#217	RestoreManufacturerDefinedDefaultParameters	0x1011#4	UINT32	rw	N	UINT32	None

Value description

Parameter	**Description**
<RestoreAllDefaultParameters>	Restores the factory settings for all parameters in the servo valve.
<RestoreCommunicationDefaultParameters>	Restores all communication parameters (index range 0x1000…0x1FFF) in the servo valve.
<RestoreApplicationDefaultParameters>	Restores all application parameters (index range 0x6000…0x9FFF) in the servo valve.
<RestoreManufacturerDefinedDefaultParameters>	Restores all manufacturer-defined parameters (index range 0x2000…0x5FFF) in the servo valve.

Table 102: Possible values of parameter 0#52 / 0#215…217

After restoring the factory parameters the valve will generate a new start to get the factory setting values effective.

⇨ Chapter "5.2 Device state machine (DSM)", page 42

The following table shows the necessary state changes to activate the restored values.

Parameter	**Set the device state machine to 'INIT' #ControlWord# MHD = 000**	**Set the field bus network state machine to 'INIT'**
<RestoreAllDefaultParameters>	X	X
<RestoreCommunicationDefaultParameters>		X
<RestoreApplicationDefaultParameters>	X	
<RestoreManufacturerDefinedDefaultParameters>	X	

Table 103: State changes needed to activate the restored values

10 Object dictionary

ⓘ The listed default values only provide the firmware preset values and not necessarily the configuration of the delivered servo valve.

ⓘ The specifications CiA 408 and CiA 301 are in accordance to the Device Profile Fluid Power.

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#20	N	devvennam	CiA 408	Device, VendorName VendorName	0x6057#0	CHAR[64]	ro	-	-	MOOG
0#21	Y	provenide	Moog DCV	Profibus, IdentNumber IdentNumber	0x4460#0	UINT16	ro	-	UINT16	0x7F4
0#22	N	devver	CiA 408	Device, DeviceVersion DeviceVersion	0x6050#0	CHAR[64]	ro	-	-	-
0#24	N	mansfwver	CiA 301	Device, ManufacturerSoftwareVersion ManufacturerSoftwareVersion	0x100A#0	CHAR[64]	ro	-	-	B99225-DV018-D-211a
0#26	N	manhdwver	CiA 301	Device, ManufacturerHardwareVersion ManufacturerHardwareVersion	0x1009#0	CHAR[64]	ro	-	-	-
0#28	N	sernum	CiA 408	Device, SerialNumber SerialNumber	0x6052#0	CHAR[64]	ro	-	-	-
0#30	N	devmdldsc	CiA 408	Device, ModelDescription ModelDescription	0x6054#0	CHAR[64]	ro	-	-	-
0#32	Y	devcodnum	CiA 408	Device, CodeNumber CodeNumber	0x6051#0	UINT16	rw	Y	UINT16	-
0#33	N	devdsc	CiA 408	Device, Description Description	0x6053#0	CHAR[64]	rw	Y	-	-
0#36	Y	deverrcod	Moog DCV	Profibus, ErrorCode ErrorCode	0x4461#0	UINT16	ro	-	UINT16	-
0#37	Y	ctlwrd	CiA 408	Device, ControlWord ControlWord	0x6040#0	UINT16	rw	N	UINT16	-
0#38	Y	stswrd	CiA 408	Device, StatusWord StatusWord	0x6041#0	UINT16	ro	-	UINT16	-
0#39	Y	devmod	CiA 408	Device, DeviceMode DeviceMode	0x6042#0	INT8	rw	N	1…4	<DeviceModeDe-fault> (0x4042)
0#40	Y	ctlmod	CiA 408	Device, ControlMode ControlMode	0x6043#0	INT8	rw	N	-1…14	<ControlModeDe-fault> (0x4043)

Table 104: Object dictionary (part 1 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#41	Y	locmod	CiA 408	Device, Local Local	0x604F#0	INT8	rw	Y	-128…1	-
0#42	N	prom2stpm	Moog DCV	Profibus, Master2SlaveTelegramProjectMSB Master2SlaveTelegramProjectMSB	0x4453#0	UINT32	rw	Y	UINT32	-
0#43	N	prom2stpl	Moog DCV	Profibus, Master2SlaveTelegramProjectLSB Master2SlaveTelegramProjectLSB	0x4452#0	UINT32	rw	Y	UINT32	-
0#44	N	pros2mtpm	Moog DCV	Profibus, Slave2MasterTelegramProjectMSB Slave2MasterTelegramProjectMSB	0x4451#0	UINT32	rw	Y	UINT32	-
0#45	N	pros2mtpl	Moog DCV	Profibus, Slave2MasterTelegramProjectLSB Slave2MasterTelegramProjectLSB	0x4450#0	UINT32	rw	Y	UINT32	-
0#46	N	protelsel	Moog DCV	Profibus, TelegramSelection TelegramSelection	0x444F#0	UINT8	rw	Y	UINT8	3
0#50	Y	devcap	CiA 408	Device, Capability Capability	0x605F#0	UINT32	ro	-	16777216…1061134336	1061093376
0#51	N	stopar[0]	CiA 301	Device, StoreParameters SaveAllParameters	0x1010#1	UINT32	rw	N	UINT32	1
0#52	N	rstpar[0]	CiA 301	Device, RestoreDefaultParameters RestoreAllDefaultParameters	0x1011#1	UINT32	rw	N	UINT32	1
0#202	N	pwrdly	Moog DCV	Device, PowerOnDelay PowerOnDelay	0x200F#0	UINT8	rw	Y	0…10	-
0#203	N	prspar	Moog DCV	ValvePressureControl, PrsSetpointParameter PrsSetpointParameter	0x3310#0	UINT32	rw	Y	UINT32	0x63800110
0#204	N	splpar	Moog DCV	ValvePositionControl, SplSetpointParameter SplSetpointParameter	0x3320#0	UINT32	rw	Y	UINT32	0x63000110
0#205	N	ctllocdef	Moog DCV	Device, LocalControlWordDefault LocalControlWordDefault	0x403F#0	UINT16	rw	Y	UINT16	0x107
0#206	Y	ctlloc	Moog DCV	Device, LocalControlWord LocalControlWord	0x4040#0	UINT16	rw	N	UINT16	<LocalControl-WordDefault> (0x403F)
0#207	N	devmoddef	Moog DCV	Device, DeviceModeDefault DeviceModeDefault	0x4042#0	INT8	rw	Y	1…2	1
0#208	N	ctlmoddef	Moog DCV	Device, ControlModeDefault ControlModeDefault	0x4043#0	INT8	rw	Y	-1…14	2
0#209	N	faisaftyp	Moog DCV	ValveDigitalOutput, DigitalOutput1Type DigitalOutput1Type	0x2420#0	INT8	ro	-	0…4	0
0#210	N	faisafupp	Moog DCV	ValveFailSafeWindowMonitoring, UpperLimit UpperLimit	0x2421#0	INT16	ro	-	<LowerLimit> (0x2422)…32767	16384

Table 104: Object dictionary (part 2 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
0#211	N	faisaflow	Moog DCV	ValveFailSafeWindowMonitoring, LowerLimit LowerLimit	0x2422#0	INT16	ro	-	-32768…<UpperLimit> (0x2421)	-16384
0#212	N	stopar[1]	CiA 301	Device, StoreParameters SaveCommunicationParameters	0x1010#2	UINT32	rw	N	UINT32	1
0#213	N	stopar[2]	CiA 301	Device, StoreParameters SaveApplicationParameters	0x1010#3	UINT32	rw	N	UINT32	1
0#214	N	stopar[3]	CiA 301	Device, StoreParameters SaveManufacturerDefinedParameters	0x1010#4	UINT32	rw	N	UINT32	1
0#215	N	rstpar[1]	CiA 301	Device, RestoreDefaultParameters RestoreCommunicationDefaultParameters	0x1011#2	UINT32	rw	N	UINT32	1
0#216	N	rstpar[2]	CiA 301	Device, RestoreDefaultParameters RestoreApplicationDefaultParameters	0x1011#3	UINT32	rw	N	UINT32	1
0#217	N	rstpar[3]	CiA 301	Device, RestoreDefaultParameters RestoreManufacturerDefinedDefaultParameters	0x1011#4	UINT32	rw	N	UINT32	1
0#218	Y	digout[0]	Moog DCV	ValveDigitalOutput, DigitalOutputSetpoint DigitalOutputSetpoint0	0x5E42#1	UINT8	rw	N	0…1	-
0#219	Y	digout[1]	Moog DCV	ValveDigitalOutput, DigitalOutputSetpoint DigitalOutputSetpoint1	0x5E42#2	UINT8	rw	N	0…1	-
0#220	N	digouttyp[0]	Moog DCV	ValveDigitalOutput, DigitalOutputConfiguration DigitalOutputConfiguration0	0x5E41#1	UINT8	ro	-	0…4	3
0#221	N	digouttyp[1]	Moog DCV	ValveDigitalOutput, DigitalOutputConfiguration DigitalOutputConfiguration1	0x5E41#2	UINT8	ro	-	0…4	0
0#222	N	sfwenasig	Moog DCV	EnableSignal, SoftwareEnableSignal SoftwareEnableSignal	0x5E43#0	UINT8	ro	-	0…1	0
0#223	Y	digoutmon[0]	Moog DCV	ValveDigitalOutput, ValveDigitalOutput DigitalOutputValue_0	0x5E44#1	UINT8	ro	-	UINT8	-
0#224	Y	digoutmon[1]	Moog DCV	ValveDigitalOutput, ValveDigitalOutput DigitalOutputValue_1	0x5E44#2	UINT8	ro	-	UINT8	-
1#20	N	drvtrdnum	CiA 408	Drive_ActualValueConditioning, InterfaceNo InterfaceNo	0x6201#0	UINT8	rw	N	1…<MaxInterfaceNo> (0x6200)	1
1#21	N	drvtrdmax	CiA 408	Drive_ActualValueConditioning, MaxInterfaceNo MaxInterfaceNo	0x6200#0	UINT8	ro	-	UINT8	8
1#22	N	drvtrdtyp	CiA 408	Drive_ActualValueConditioning, Type Type	0x6202#0	INT8	rw	N	-61…69	-
1#23	N	drvtrdprsmin	CiA 408	Drive_ActualValueConditioning, MinimumPressure MinimumPressure	0x6220#1	INT32	rw	N	-32768…32767	-16384
1#25	N	drvtrdprssigmin	CiA 408	Drive_ActualValueConditioning, MinimumSignal MinimumSignal	0x6224#1	INT32	rw	N	-32768…32767	-16384

Table 104: Object dictionary (part 3 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#26	N	drvtrdprsmax	CiA 408	Drive_ActualValueConditioning, MaximumPressure MaximumPressure	0x6221#1	INT32	rw	N	-32768…32767	16384
1#28	N	drvtrdprssigmax	CiA 408	Drive_ActualValueConditioning, MaximumSignal MaximumSignal	0x6225#1	INT32	rw	N	-32768…32767	16384
1#29	N	drvtrdprsare	CiA 408	Drive_ActualValueConditioning, PressureArea PressureArea	0x6222#1	INT32	rw	N	-32768…32767	-
1#32	N	drvtrdprsofs	CiA 408	Drive_ActualValueConditioning, PressureOffset PressureOffset	0x6223#1	INT32	rw	N	-32768…32767	-
1#41	N	drvtrdres	CiA 408	Drive_ActualValueConditioning, Resolution Resolution	0x6240#1	INT32	rw	N	INT32	1
1#44	N	drvtrdofs	CiA 408	Drive_ActualValueConditioning, PositionOffset PositionOffset	0x6241#1	INT32	rw	N	INT32	-
1#47	N	drvtrdzroshf	CiA 408	Drive_ActualValueConditioning, ZeroShift ZeroShift	0x6242#1	UINT8	rw	N	UINT8	-
1#50	N	drvtrdminref	CiA 408	Drive_ActualValueConditioning, MinimumReference MinimumReference	0x6230#1	INT32	rw	N	INT32	-1
1#53	N	drvtrdmaxref	CiA 408	Drive_ActualValueConditioning, MaximumReference MaximumReference	0x6231#1	INT32	rw	N	INT32	1
1#56	N	drvtrdltc	CiA 408	Drive_ActualValueConditioning, T1 T1	0x6232#1	UINT32	rw	N	UINT32	-
1#59	N	drvtrdminitf	CiA 408	Drive_ActualValueConditioning, MinimumInterface MinimumInterface	0x6233#1	INT32	rw	N	INT32	-1
1#62	N	drvtrdmaxitf	CiA 408	Drive_ActualValueConditioning, MaximumInterface MaximumInterface	0x6234#1	INT32	rw	N	INT32	1
1#65	N	drvtrdc	CiA 408	Drive_ActualValueConditioning, StartStopEncoderTime StartStopEncoderTime	0x6244#1	INT32	rw	N	0…2147483647	-
1#68	N	drvtrdstastotyp	CiA 408	Drive_ActualValueConditioning, StartStopEncoderType StartStopEncoderType	0x6245#1	INT8	rw	N	0…127	1
1#69	N	drvtrdbitsiz	CiA 408	Drive_ActualValueConditioning, BitSize BitSize	0x6243#1	UINT8	rw	N	0…32	24
1#83	Y	drvtrdval	CiA 408	Drive_ActualValueConditioning, ActualValue ActualValue	0x6204#1	INT32	ro	-	INT32	-
1#86	N	drvtrdsgn	CiA 408	Drive_ActualValueConditioning, Sign Sign	0x6203#0	INT8	rw	N	-1…1	1
1#87	Y	drvtrditfval[0]	CiA 408	Drive_ActValCond, ActualValue ActualValue	0x6210#1	INT32	ro	-	INT32	-
1#88	Y	drvtrditfval[1]	CiA 408	Drive_ActValCond, ActualValue ActualValue	0x6211#1	INT32	ro	-	INT32	-

Table 104: Object dictionary (part 4 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
1#89	Y	drvtrditfval[2]	CiA 408	Drive_ActValCond, ActualValue ActualValue	0x6212#1	INT32	ro	-	INT32	-
1#90	Y	drvtrditfval[3]	CiA 408	Drive_ActValCond, ActualValue ActualValue	0x6213#1	INT32	ro	-	INT32	-
1#91	Y	drvtrditfval[4]	CiA 408	Drive_ActValCond, ActualValue ActualValue	0x6214#1	INT32	ro	-	INT32	-
1#92	Y	drvtrditfval[5]	CiA 408	Drive_ActValCond, ActualValue ActualValue	0x6215#1	INT32	ro	-	INT32	-
1#93	Y	drvtrditfval[6]	CiA 408	Drive_ActValCond, ActualValue ActualValue	0x6216#1	INT32	ro	-	INT32	-
1#94	Y	drvtrditfval[7]	CiA 408	Drive_ActValCond, ActualValue ActualValue	0x6217#1	INT32	ro	-	INT32	-
1#95	Y	prsref	Moog DCV	ValvePressureControl_DemandValueGenerator, PrsReferenceValue PrsReferenceValue	0x231C#1	INT16	rw	N	0…32767	400
1#96	N	prsrefuni	Moog DCV	ValvePressureControl_DemandValueGenerator, PrsReferenceValue Unit	0x231C#2	UINT8	ro	-	UINT8	0x4E
1#97	N	prsrefprf	Moog DCV	ValvePressureControl_DemandValueGenerator, PrsReferenceValue Prefix	0x231C#3	INT8	ro	-	INT8	0
1#201	N	drvtrdpar	CiA 408	Drive_ActualValueConditioning, TransducerPort TransducerPort	0x4032#0	UINT32	rw	N	UINT32	-
1#202…207	N	drvtrdfltcof[0…5]	CiA 408	Drive_ActualValueConditioning, FilterCoefficient FilterCoefficient	0x4031#1…6	FLOAT32	rw	N	FLOAT32	-
1#209	N	drvtrdbdr	Moog DCV	Drive_ActualValueConditioning, BitRate BitRate	0x561F#0	UINT8	rw	Y	0…7	2
1#210	N	drvtrdmstslv	Moog DCV	Drive_ActualValueConditioning, MasterSlave MasterSlave	0x5620#0	UINT8	rw	N	1…1	1
1#211	N	drvtrdgenmin	CiA 408	_Drive_ActualValueConditioning, GeneralInputMinimum GeneralInputMinimum	0x6228#1	INT32	rw	N	INT32	-
1#212	N	drvtrdgenmax	CiA 408	_Drive_ActualValueConditioning, GeneralInputMaximum GeneralInputMaximum	0x6229#1	INT32	rw	N	INT32	16384
1#213	N	drvtrdgenofs	CiA 408	_Drive_ActualValueConditioning, GeneralInputOffset GeneralInputOffset	0x622B#1	INT32	rw	N	INT32	-
1#214	N	drvtrdgensigmin	CiA 408	_Drive_ActualValueConditioning, GeneralInputMinimumSignal GeneralInputMinimumSignal	0x622C#1	INT32	rw	N	INT32	-
1#215	N	drvtrdgensigmax	CiA 408	_Drive_ActualValueConditioning, GeneralInputMaximumSignal GeneralInputMaximumSignal	0x622D#1	INT32	rw	N	INT32	16384
12#20	N	positfref[0]	CiA 408	DrivePositionControl, InterfaceReference InterfaceReference	0x6602#1	UINT8	rw	Y	0…8	1

Table 104: Object dictionary (part 5 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#21	Y	posset	CiA 408	DrivePositionControl, Setpoint Setpoint	0x6600#1	INT32	rw	N	INT32	-
12#22	N	posuni	CiA 408	DrivePositionControl, Setpoint Position_Unit	0x6600#2	UINT8	ro	-	UINT8	0
12#23	N	posprf	CiA 408	DrivePositionControl, Setpoint Position_Prefix	0x6600#3	INT8	ro	-	INT8	0
12#24	Y	posdem	CiA 408	DrivePositionControl_DemandValueGenerator, DemandValue DemandValue	0x6610#1	INT32	ro	-	INT32	-
12#25	N	posuni	CiA 408	DrivePositionControl_DemandValueGenerator, DemandValue Position_Unit	0x6610#2	UINT8	ro	-	UINT8	0
12#26	N	posprf	CiA 408	DrivePositionControl_DemandValueGenerator, DemandValue Position_Prefix	0x6610#3	INT8	ro	-	INT8	0
12#27	N	posminref	Moog DCV	DrivePositionControl, MinimumReference MinimumReference	0x4230#1	INT32	rw	Y	INT32	-16384
12#30	N	posmaxref	Moog DCV	DrivePositionControl, MaximumReference MaximumReference	0x4231#1	INT32	rw	Y	INT32	16384
12#31	N	posminitf	Moog DCV	DrivePositionControl, MinimumInterface MinimumInterface	0x5509#1	INT32	rw	Y	INT32	-16384
12#32	N	posmaxitf	Moog DCV	DrivePositionControl, MaximumInterface MaximumInterface	0x5510#1	INT32	rw	Y	INT32	16384
12#33	Y	possethld	CiA 408	DrivePositionControl, DemandValueGenerator_HoldSetpoint DemandValueGenerator_HoldSetpoint	0x6614#1	INT32	rw	Y	INT32	-
12#34	N	posuni	CiA 408	DrivePositionControl, DemandValueGenerator_HoldSetpoint Position_Unit	0x6614#2	UINT8	ro	-	UINT8	0
12#35	N	posprf	CiA 408	DrivePositionControl, DemandValueGenerator_HoldSetpoint Position_Prefix	0x6614#3	INT8	ro	-	INT8	0
12#100	Y	posval[0]	CiA 408	DrivePositionControl, ActualValues ActualValue#1	0x6601#1	INT32	ro	-	INT32	-
12#103	Y	posctldvn	CiA 408	DrivePositionControl, dX dX	0x6650#1	INT32	ro	-	INT32	-
12#140	Y	posmontyp	CiA 408	DrivePositionControl, ControlMonitoring_Type ControlMonitoring_Type	0x6651#0	INT8	rw	Y	0…1	-
12#141	Y	posmonupp	CiA 408	DrivePositionControl, ControlMonitoring_UpperThreshold ControlMonitoring_UpperThreshold	0x6654#0	INT32	rw	Y	INT32	512
12#144	Y	posmonlow	CiA 408	DrivePositionControl, ControlMonitoring_LowerThreshold ControlMonitoring_LowerThreshold	0x6655#0	INT32	rw	Y	INT32	-512
12#147	Y	posmontim	CiA 408	DrivePositionControl, ControlMonitoring_Time ControlMonitoring_Time	0x6652#0	UINT16	rw	Y	UINT16	-

Table 104: Object dictionary (part 6 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#205	N	pospar	Moog DCV	DrivePositionControl, DemandParameter DemandParameter	0x3330#0	UINT32	rw	Y	UINT32	0x66000120
12#206	Y	pospt1	Moog DCV	DrivePositionControl, KpT1TimeConstant KpT1TimeConstant	0x5508#1	FLOAT32	rw	Y	0.00…+inf	-
12#207	Y	posfbp	Moog DCV	DrivePositionControl, FilterLowPassFilter FilterLowPassFilter	0x550C#1	FLOAT32	rw	Y	0.00…+inf	-
12#208	Y	posdt1	Moog DCV	DrivePositionControl, T1 T1	0x5503#1	FLOAT32	rw	Y	0.00…+inf	-
12#209	Y	posctldiv	Moog DCV	DrivePositionControl, ControlTimeDivider ControlTimeDivider	0x553F#1	UINT8	rw	Y	2…255	10
12#210	Y	posign[0]	Moog DCV	DrivePositionControl, DrivePositionControl Outer_I_Gain	0x5504#1	FLOAT32	rw	Y	FLOAT32	-
12#211	Y	posign[1]	Moog DCV	DrivePositionControl, DrivePositionControl Inner_I_Gain	0x5504#2	FLOAT32	rw	Y	FLOAT32	-
12#212	Y	posval[1]	CiA 408	DrivePositionControl, ActualValues ActualValue#2	0x6601#2	INT32	ro	-	INT32	-
12#213	Y	posval[2]	CiA 408	DrivePositionControl, ActualValues ActualValue#3	0x6601#3	INT32	ro	-	INT32	-
12#214	Y	posval[3]	CiA 408	DrivePositionControl, ActualValues ActualValue#4	0x6601#4	INT32	ro	-	INT32	-
12#215	N	positfref[1]	CiA 408	DrivePositionControl, InterfaceReference InterfaceReference	0x6602#2	UINT8	rw	Y	0…8	0
12#216	N	positfref[2]	CiA 408	DrivePositionControl, InterfaceReference InterfaceReference	0x6602#3	UINT8	rw	Y	0…8	0
12#217	N	positfref[3]	CiA 408	DrivePositionControl, InterfaceReference InterfaceReference	0x6602#4	UINT8	rw	Y	0…8	0
12#218	Y	posveldem	Moog DCV	DrivePositionControl, VelocityDemandValue VelocityDemandValue	0x561E#0	INT32	ro	-	INT32	-
12#220	Y	posdemflt	Moog DCV	DrivePositionControl, PositionDemandFilter PositionDemandFilter	0x582A#0	FLOAT32	rw	Y	0.00…+inf	-
12#221	Y	posdemfltout	Moog DCV	DrivePositionControl, PositionDemandFilterOutput PositionDemandFilterOutput	0x582B#0	FLOAT32	ro	-	FLOAT32	-
12#222	N	pospgnout[0]	Moog DCV	DrivePositionControl, DrivePositionControl KpOutput	0x552A#1	FLOAT32	ro	-	FLOAT32	-
12#223	N	pospgnout[1]	Moog DCV	DrivePositionControl, DrivePositionControl KpT1Output	0x552A#2	FLOAT32	ro	-	FLOAT32	-
12#224	Y	sncpgn	Moog DCV	DrivePositionControl, SynchronisationGain SynchronisationGain	0x5829#0	FLOAT32	rw	Y	FLOAT32	-

Table 104: Object dictionary (part 7 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#225	N	pospgn[0]	Moog DCV	DrivePositionControl, DrivePositionControl Kp	0x5501#1	FLOAT32	rw	Y	FLOAT32	-
12#226	N	pospgn[1]	Moog DCV	DrivePositionControl, DrivePositionControl KpT1	0x5501#2	FLOAT32	rw	Y	FLOAT32	-
12#227	Y	posdgn	Moog DCV	DrivePositionControl, Td Td	0x5502#1	FLOAT32	rw	Y	FLOAT32	-
12#228	Y	posicrout	Moog DCV	DrivePositionControl, Integrator_OuterControlRange Integrator_OuterControlRange	0x5505#1	FLOAT32	rw	Y	0.00…2147483648.00	-
12#229	Y	posicrin	Moog DCV	DrivePositionControl, Integrator_InnerControlRange Integrator_InnerControlRange	0x5511#1	FLOAT32	rw	Y	0.00…2147483648.00	-
12#230	N	posdgnout	Moog DCV	DrivePositionControl, KdOutput KdOutput	0x552B#1	FLOAT32	ro	-	FLOAT32	-
12#231	N	posignout	Moog DCV	DrivePositionControl, KiOutput KiOutput	0x552C#1	FLOAT32	ro	-	FLOAT32	-
12#232	N	posffvout	Moog DCV	DrivePositionControl, FeedForwardVelocityOutput FeedForwardVelocityOutput	0x552E#1	FLOAT32	ro	-	FLOAT32	-
12#233	N	posffaout	Moog DCV	DrivePositionControl, FeedForwardAccelerationOutput FeedForwardAccelerationOutput	0x552F#1	FLOAT32	ro	-	FLOAT32	-
12#234	N	posfbvout	Moog DCV	DrivePositionControl, VelocityStateFeedbackOutput VelocityStateFeedbackOutput	0x5530#1	FLOAT32	ro	-	FLOAT32	-
12#235	N	posfbaout	Moog DCV	DrivePositionControl, AccelerationStateFeedbackOutput AccelerationStateFeedbackOutput	0x5531#1	FLOAT32	ro	-	FLOAT32	-
12#236	Y	posffv	Moog DCV	DrivePositionControl, VelocityFeedforwardProportionalGain VelocityFeedforwardProportionalGain	0x5506#1	FLOAT32	rw	Y	FLOAT32	-
12#237	Y	posffa	Moog DCV	DrivePositionControl, AccelerationFeedforwardProportionalGain AccelerationFeedforwardProportionalGain	0x5507#1	FLOAT32	rw	Y	FLOAT32	-
12#238	Y	posfba	Moog DCV	DrivePositionControl, StateFeedbackAccelerationProportionalGain StateFeedbackAccelerationProportionalGain	0x550A#1	FLOAT32	rw	Y	FLOAT32	-
12#239	Y	posfbv	Moog DCV	DrivePositionControl, StateFeedbackVelocityProportionalGain StateFeedbackVelocityProportionalGain	0x550B#1	FLOAT32	rw	Y	FLOAT32	-
12#240	Y	posctlout	Moog DCV	DrivePositionControl, PositionControllerOutput PositionControllerOutput	0x550F#1	FLOAT32	ro	-	FLOAT32	-
12#241	Y	posposgn	Moog DCV	DrivePositionControl, PositiveOverallGain PositiveOverallGain	0x550D#1	FLOAT32	rw	Y	FLOAT32	1.00
12#242	Y	posneggn	Moog DCV	DrivePositionControl, NegativeOverallGain NegativeOverallGain	0x550E#1	FLOAT32	rw	Y	FLOAT32	1.00
12#244	Y	posaccdem	Moog DCV	DrivePositionControl, AccelerationDemandvalue AccelerationDemandvalue	0x5616#1	INT32	ro	-	INT32	-

Table 104: Object dictionary (part 8 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
12#245	N	poshldtgt	Moog DCV	DrivePositionControl, HoldSetpointTarget HoldSetpointTarget	0x585C#0	INT8	rw	Y	0…2	-
12#246	Y	posignlim	Moog DCV	DrivePositionControl, IntegratorLimit IntegratorLimit	0x5513#1	FLOAT32	rw	Y	0.00…+inf	16384.00
12#247	Y	poslimwin[0]	Moog DCV	DrivePositionControl, DrivePositionControl PositionWindow_0	0x570C#1	INT32	rw	Y	0…2147483647	-
12#248	Y	poslimwin[1]	Moog DCV	DrivePositionControl, DrivePositionControl PositionWindow_1	0x570C#2	INT32	rw	Y	0…2147483647	-
12#249	N	sncnumaxs	Moog DCV	DrivePositionControl, NumberOfAxisToSyncronize NumberOfAxisToSyncronize	0x5830#0	INT8	rw	Y	1…4	3
13#20	N	velitfref	Moog DCV	DriveVelocityControl, InterfaceReference InterfaceReference	0x5542#1	UINT8	rw	Y	1…<MaxInterfaceNo> (0x6200)	1
13#21	Y	velset	CiA 408	DriveVelocityControl, Setpoint Setpoint	0x6500#1	INT32	rw	N	-10000000…10000000	-
13#22	N	veluni	CiA 408	DriveVelocityControl, Setpoint Velocity_Unit	0x6500#2	UINT8	ro	-	UINT8	0
13#23	Y	velsethld	Moog DCV	DrivePositionControl_DemandValueGenerator, HoldSetpoint HoldSetpoint	0x5527#1	INT32	rw	Y	INT32	-
13#24	Y	veldem	Moog DCV	DriveVelocityControl, VelocityDemandValue VelocityDemandValue	0x5615#1	INT32	ro	-	INT32	-
13#25	N	velprf	CiA 408	DriveVelocityControl, Setpoint Velocity_Prefix	0x6500#3	INT8	ro	-	INT8	0
13#27	N	velminref	Moog DCV	DriveVelocityControl, MinimumReference MinimumReference	0x5523#1	INT32	rw	Y	INT32	-16384
13#30	N	velmaxref	Moog DCV	DriveVelocityControl, MaximumReference MaximumReference	0x5524#1	INT32	rw	Y	INT32	16384
13#31	N	velminitf	Moog DCV	DriveVelocityControl, MinimumInterface MinimumInterface	0x5525#1	INT32	rw	Y	INT32	-16384
13#32	N	velmaxitf	Moog DCV	DriveVelocityControl, MaximumInterface MaximumInterface	0x5526#1	INT32	rw	Y	INT32	16384
13#33	N	veluni	Moog DCV	DriveVelocityControl, VelocityDemandValue Velocity_Unit	0x5615#2	UINT8	ro	-	UINT8	0
13#34	N	velprf	Moog DCV	DriveVelocityControl, VelocityDemandValue Velocity_Prefix	0x5615#3	INT8	ro	-	INT8	0
13#35	N	veluni	Moog DCV	DrivePositionControl_DemandValueGenerator, HoldSetpoint Velocity_Unit	0x5527#2	UINT8	ro	-	UINT8	0
13#36	N	velprf	Moog DCV	DrivePositionControl_DemandValueGenerator, HoldSetpoint Velocity_Prefix	0x5527#3	INT8	ro	-	INT8	0

Table 104: Object dictionary (part 9 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#100	Y	velval	CiA 408	DriveVelocityControl, ActualValue ActualValue	0x6501#0	INT32	ro	-	INT32	-
13#103	Y	velctldvn	CiA 408	DriveVelocityControl, dV dV	0x6550#1	INT32	ro	-	INT32	-
13#112	Y	velmontyp	CiA 408	DriveVelocityControl, ControlMonitoring_Type ControlMonitoring_Type	0x6551#0	INT8	rw	Y	0…1	-
13#113	Y	velmonupp	CiA 408	DriveVelocityControl, ControlMonitoring_UpperThreshold ControlMonitoring_UpperThreshold	0x6554#0	INT32	rw	Y	INT32	512
13#116	Y	velmonlow	CiA 408	DriveVelocityControl, ControlMonitoring_LowerThreshold ControlMonitoring_LowerThreshold	0x6555#0	INT32	rw	Y	INT32	-512
13#119	Y	velmontim	CiA 408	DriveVelocityControl, ControlMonitoring_Time ControlMonitoring_Time	0x6552#0	UINT16	rw	Y	UINT16	-
13#205	N	velpar	Moog DCV	DriveVelocityControl, DemandParameter DemandParameter	0x5520#0	UINT32	rw	Y	UINT32	0x65000120
13#206	Y	velign[0]	Moog DCV	DriveVelocityControl, DriveVelocityControl IntegratorGain_Low	0x5522#1	FLOAT32	rw	Y	FLOAT32	-
13#207	Y	velign[1]	Moog DCV	DriveVelocityControl, DriveVelocityControl IntegratorGain_High	0x5522#2	FLOAT32	rw	Y	FLOAT32	-
13#208	Y	vellim	Moog DCV	DrivePositionControl, VelocityLimit VelocityLimit	0x570E#0	INT32	rw	Y	INT32	-
13#209	Y	vellimwin	Moog DCV	DrivePositionControl, VelocityLimitWindow VelocityLimitWindow	0x570D#0	INT32	rw	Y	0…2147483647	-
13#210	N	velcalmod	Moog DCV	DriveVelocityControl, VelocityCalculationMode VelocityCalculationMode	0x5545#1	UINT8	rw	Y	UINT8	-
13#211	N	velposfltout	Moog DCV	DriveVelocityControl, PositionSignalFilterOutput PositionSignalFilterOutput	0x5552#1	FLOAT32	ro	-	FLOAT32	-
13#212	N	velquelen	Moog DCV	DriveVelocityControl, VelocitySignalQueueLength VelocitySignalQueueLength	0x5550#1	UINT8	rw	Y	1…20	1
13#213	N	velres	Moog DCV	DriveVelocityControl, VelocityResolution VelocityResolution	0x5544#1	FLOAT32	rw	Y	FLOAT32	1.00
13#214	Y	velctldiv	Moog DCV	DriveVelocityControl, ControlTimeDivider ControlTimeDivider	0x5540#1	UINT8	rw	Y	1…255	2
13#215	N	velvelflt	Moog DCV	DriveVelocityControl, VelocitySignalFilterTimeConstant VelocitySignalFilterTimeConstant	0x5548#1	FLOAT32	rw	Y	0.00…+inf	-
13#216	Y	vellimupp	CiA 408	DriveVelocityControl, VelocityUpperDemandLimit VelocityUpperDemandLimit	0x6520#1	INT32	rw	Y	INT32	16384
13#217	Y	vellimlow	CiA 408	DriveVelocityControl, VelocityLowerDemandLimit VelocityLowerDemandLimit	0x6521#1	INT32	rw	Y	INT32	-16384

Table 104: Object dictionary (part 10 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
13#218	N	veldemrmp	Moog DCV	DriveVelocityControl, VelocityDemandRamp VelocityDemandRamp	0x5553#1	INT32	rw	Y	INT32	-
13#219	Y	veldemflt	Moog DCV	DriveVelocityControl, VelocityDemandFilter VelocityDemandFilter	0x5533#1	FLOAT32	rw	Y	0.00…+inf	-
13#220	Y	veldemfltout	Moog DCV	DriveVelocityControl, VelocityDemandFilterOutput VelocityDemandFilterOutput	0x553B#1	FLOAT32	ro	-	FLOAT32	-
13#221	Y	velpgnout	Moog DCV	DriveVelocityControl, VelocityProportionalGainOut VelocityProportionalGainOut	0x5537#1	FLOAT32	ro	-	FLOAT32	-
13#222	N	velpgn	Moog DCV	DriveVelocityControl, Kp Kp	0x5514#1	FLOAT32	rw	Y	FLOAT32	-
13#223	Y	velicrout	Moog DCV	DriveVelocityControl_Integrator, OuterControlRange OuterControlRange	0x5516#1	FLOAT32	rw	Y	0.00…2147483648.00	-
13#224	Y	velicrin	Moog DCV	DriveVelocityControl_Integrator, InnerControlRange InnerControlRange	0x5517#1	FLOAT32	rw	Y	0.00…2147483648.00	-
13#225	Y	velignout	Moog DCV	DriveVelocityControl, VelocityIntegratorOutput VelocityIntegratorOutput	0x553C#1	FLOAT32	ro	-	FLOAT32	-
13#226	N	velvelfltout	Moog DCV	DriveVelocityControl, VelocitySignalFilterOutput VelocitySignalFilterOutput	0x5549#1	FLOAT32	ro	-	FLOAT32	-
13#227	N	accval	Moog DCV	DriveVelocityControl, AccelerationSignal AccelerationSignal	0x5551#1	FLOAT32	ro	-	FLOAT32	-
13#228	Y	velfbapgn	Moog DCV	DriveVelocityControl, StateFeedbackAccelerationProportionalGain StateFeedbackAccelerationProportionalGain	0x5529#1	FLOAT32	rw	Y	FLOAT32	-
13#229	Y	velfbapgnout	Moog DCV	DriveVelocityControl, VelocityFeedbackAccelerationGain VelocityFeedbackAccelerationGain	0x5539#1	FLOAT32	ro	-	FLOAT32	-
13#230	N	velffvout	Moog DCV	DriveVelocityControl, VelocityFeedForwardOut VelocityFeedForwardOut	0x5546#1	FLOAT32	ro	-	FLOAT32	-
13#231	N	velffv	Moog DCV	DriveVelocityControl, VelocityFeedForwardGain VelocityFeedForwardGain	0x5547#1	FLOAT32	rw	Y	FLOAT32	-
13#232	Y	velposgn	Moog DCV	DriveVelocityControl, VelocityPositiveOverallGain VelocityPositiveOverallGain	0x5534#1	FLOAT32	rw	Y	FLOAT32	-
13#233	Y	velneggn	Moog DCV	DriveVelocityControl, VelocityNegativeOverallGain VelocityNegativeOverallGain	0x5535#1	FLOAT32	rw	Y	FLOAT32	-
13#234	Y	velctlout	Moog DCV	DriveVelocityControl, VelocityControllerOutput VelocityControllerOutput	0x5519#1	FLOAT32	ro	-	FLOAT32	-
13#236	Y	velignlim	Moog DCV	DriveVelocityControl_Integrator, IntegratorLimit IntegratorLimit	0x5518#1	FLOAT32	rw	Y	0.00…+inf	16384.00
21#21	Y	splset	CiA 408	ValvePositionControl, SplSetpoint SplSetpoint	0x6300#1	INT16	rw	N	INT16	-

Table 104: Object dictionary (part 11 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#22	N	spluni	CiA 408	ValvePositionControl, SplSetpoint Unit	0x6300#2	UINT8	ro	-	UINT8	0
21#23	N	splprf	CiA 408	ValvePositionControl, SplSetpoint Prefix	0x6300#3	INT8	ro	-	INT8	0
21#24	Y	spldem	CiA 408	ValvePositionControl_DemandValueGeneratorSplDemand, SplDe-mandValue SplDemandValue	0x6310#1	INT16	ro	-	INT16	-
21#25	N	spluni	CiA 408	ValvePositionControl_DemandValueGeneratorSplDemand, SplDe-mandValue Unit	0x6310#2	UINT8	ro	-	UINT8	0
21#26	N	splprf	CiA 408	ValvePositionControl_DemandValueGeneratorSplDemand, SplDe-mandValue Prefix	0x6310#3	INT8	ro	-	INT8	0
21#27	Y	spldemref	CiA 408	ValvePositionControl_DemandValueGenerator, SplReferenceValue SplReferenceValue	0x6311#1	INT16	ro	-	INT16	16384
21#28	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator, SplReferenceValue Unit	0x6311#2	UINT8	ro	-	UINT8	0
21#29	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator, SplReferenceValue Prefix	0x6311#3	INT8	ro	-	INT8	0
21#30	Y	splsethld	CiA 408	ValvePositionControl_DemandValueGeneratorSplHoldSetPoint, SplHoldSetPoint SplHoldSetPoint	0x6314#1	INT16	rw	Y	INT16	-
21#31	N	spluni	CiA 408	ValvePositionControl_DemandValueGeneratorSplHoldSetPoint, SplHoldSetPoint Unit	0x6314#2	UINT8	ro	-	UINT8	0
21#32	N	splprf	CiA 408	ValvePositionControl_DemandValueGeneratorSplHoldSetPoint, SplHoldSetPoint Prefix	0x6314#3	INT8	ro	-	INT8	0
21#33	Y	spllimupp	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, UpperLimit UpperLimit	0x6320#1	INT16	rw	Y	<LowerLimit> (0x6321)…32767	16384
21#34	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, UpperLimit Unit	0x6320#2	UINT8	ro	-	UINT8	0
21#35	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, UpperLimit Prefix	0x6320#3	INT8	ro	-	INT8	0
21#36	Y	spllimlow	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, LowerLimit LowerLimit	0x6321#1	INT16	rw	Y	-32768…<UpperLimit> (0x6320)	-16384
21#37	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, LowerLimit Unit	0x6321#2	UINT8	ro	-	UINT8	0

Table 104: Object dictionary (part 12 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#38	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, LowerLimit Prefix	0x6321#3	INT8	ro	-	INT8	0
21#39	Y	spldemfct	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Factor Factor	0x6322#0	UINT32	rw	Y	UINT32	0x10001
21#40	Y	spldemofs	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Offset Offset	0x6323#1	INT16	rw	Y	INT16	-
21#41	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Offset Unit	0x6323#2	UINT8	ro	-	UINT8	0
21#42	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Offset Prefix	0x6323#3	INT8	ro	-	INT8	0
21#43	Y	splrmptyp	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Type Type	0x6330#0	INT8	rw	Y	0…3	-
21#44	Y	splrmpacl	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration-Time AccelerationTime	0x6331#1	UINT16	rw	Y	UINT16	-
21#45	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration-Time Unit	0x6331#2	UINT8	ro	-	UINT8	3
21#46	Y	splrmpaclprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration-Time AccelerationTime_Prefix	0x6331#3	INT8	rw	Y	-4…0	-3
21#47	Y	splrmpaclneg	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration-TimeNegative AccelerationTimeNegative	0x6333#1	UINT16	rw	Y	UINT16	-
21#48	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration-TimeNegative Unit	0x6333#2	UINT8	ro	-	UINT8	3
21#49	Y	splrmpaclnegprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration-TimeNegative AccelerationTimeNegative_Prefix	0x6333#3	INT8	rw	Y	-4…0	-3
21#50	Y	splrmpaclpos	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration-TimePositive AccelerationTimePositive	0x6332#1	UINT16	rw	Y	UINT16	-
21#51	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration-TimePositive Unit	0x6332#2	UINT8	ro	-	UINT8	3
21#52	Y	splrmpaclposprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration-TimePositive AccelerationTimePositive_Prefix	0x6332#3	INT8	rw	Y	-4…0	-3

Table 104: Object dictionary (part 13 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#53	Y	splrmpdcl	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration-Time DecelerationTime	0x6334#1	UINT16	rw	Y	UINT16	-
21#54	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration-Time Unit	0x6334#2	UINT8	ro	-	UINT8	3
21#55	Y	splrmpdclprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration-Time DecelerationTime_Prefix	0x6334#3	INT8	rw	Y	-4…0	-3
21#56	Y	splrmpdclneg	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration-TimeNegative DecelerationTimeNegative	0x6336#1	UINT16	rw	Y	UINT16	-
21#57	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration-TimeNegative Unit	0x6336#2	UINT8	ro	-	UINT8	3
21#58	Y	splrmpdclnegprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration-TimeNegative DecelerationTimeNegative_Prefix	0x6336#3	INT8	rw	Y	-4…0	-3
21#59	Y	splrmpdclpos	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration-TimePositive DecelerationTimePositive	0x6335#1	UINT16	rw	Y	UINT16	-
21#60	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration-TimePositive Unit	0x6335#2	UINT8	ro	-	UINT8	3
21#61	Y	splrmpdclposprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration-TimePositive DecelerationTimePositive_Prefix	0x6335#3	INT8	rw	Y	-4…0	-3
21#86	Y	spldirtyp	CiA 408	ValvePositionControl_DemandValueGenerator, DirectionalDepen-dentGain_Type DirectionalDependentGain_Type	0x6340#0	INT8	rw	Y	0…1	-
21#87	Y	spldirfct	CiA 408	ValvePositionControl_DemandValueGenerator, DirectionalDepen-dentGain_Factor DirectionalDependentGain_Factor	0x6341#0	UINT32	rw	Y	UINT32	0x10001
21#96	Y	splchrtyp	CiA 408	ValvePositionControl_DemandValueGenerator, CharacteristicCom-pensation_Type CharacteristicCompensation_Type	0x6346#0	INT8	rw	Y	-1…0	-
21#106	Y	spldbdtyp	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompen-sationType DeadbandCompensationType	0x6342#0	INT8	rw	Y	0…2	-

Table 104: Object dictionary (part 14 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#107	Y	spldbdsida	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_ASide DeadbandCompensation_ASide	0x6343#1	INT16	rw	Y	0…16384	-
21#108	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_ASide Unit	0x6343#2	UINT8	ro	-	UINT8	0
21#109	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_ASide Prefix	0x6343#3	INT8	ro	-	INT8	0
21#110	Y	spldbdsidb	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_BSide DeadbandCompensation_BSide	0x6344#1	INT16	rw	Y	0…16384	-
21#111	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_BSide Unit	0x6344#2	UINT8	ro	-	UINT8	0
21#112	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_BSide Prefix	0x6344#3	INT8	ro	-	INT8	0
21#113	Y	spldbdtrs	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_Threshold DeadbandCompensation_Threshold	0x6345#1	INT16	rw	Y	0…16383	-
21#114	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_Threshold Unit	0x6345#2	UINT8	ro	-	UINT8	0
21#115	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_Threshold Prefix	0x6345#3	INT8	ro	-	INT8	0
21#128	Y	splzrocor	CiA 408	ValvePositionControl_DemandValueGenerator_ZeroCorrection, Offset Offset	0x6324#1	INT16	rw	Y	INT16	-
21#129	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_ZeroCorrection, Offset Unit	0x6324#2	UINT8	ro	-	UINT8	0
21#130	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_ZeroCorrection, Offset Prefix	0x6324#3	INT8	ro	-	INT8	0
21#144	Y	splval	CiA 408	ValvePositionControl, SplActualValue SplActualValue	0x6301#1	INT16	ro	-	INT16	-
21#145	N	spluni	CiA 408	ValvePositionControl, SplActualValue Unit	0x6301#2	UINT8	ro	-	UINT8	0

Table 104: Object dictionary (part 15 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#146	N	splprf	CiA 408	ValvePositionControl, SplActualValue Prefix	0x6301#3	INT8	ro	-	INT8	0
21#147	Y	splctldvn	CiA 408	ValvePositionControl, SplControlDeviation SplControlDeviation	0x6350#1	INT16	ro	-	INT16	-
21#148	N	spluni	CiA 408	ValvePositionControl, SplControlDeviation Unit	0x6350#2	UINT8	ro	-	UINT8	0
21#149	N	splprf	CiA 408	ValvePositionControl, SplControlDeviation Prefix	0x6350#3	INT8	ro	-	INT8	0
21#150	Y	splmontyp	CiA 408	ValvePositionControl_ControlMonitoring, Type Type	0x6351#0	INT8	rw	Y	0…1	-
21#151	Y	splmonupp	CiA 408	ValvePositionControl_ControlMonitoring, UpperThreshold UpperThreshold	0x6354#1	INT16	rw	Y	INT16	512
21#152	N	spluni	CiA 408	ValvePositionControl_ControlMonitoring, UpperThreshold Unit	0x6354#2	UINT8	ro	-	UINT8	0
21#153	N	splprf	CiA 408	ValvePositionControl_ControlMonitoring, UpperThreshold Prefix	0x6354#3	INT8	ro	-	INT8	0
21#154	Y	splmonlow	CiA 408	ValvePositionControl_ControlMonitoring, LowerThreshold LowerThreshold	0x6355#1	INT16	rw	Y	INT16	-512
21#155	N	spluni	CiA 408	ValvePositionControl_ControlMonitoring, LowerThreshold Unit	0x6355#2	UINT8	ro	-	UINT8	0
21#156	N	splprf	CiA 408	ValvePositionControl_ControlMonitoring, LowerThreshold Prefix	0x6355#3	INT8	ro	-	INT8	0
21#157	Y	splmontim	CiA 408	ValvePositionControl_ControlMonitoring, DelayTime DelayTime	0x6352#1	UINT16	rw	Y	UINT16	30
21#158	N	timuni	CiA 408	ValvePositionControl_ControlMonitoring, DelayTime Unit	0x6352#2	UINT8	ro	-	UINT8	3
21#159	N	timprf	CiA 408	ValvePositionControl_ControlMonitoring, DelayTime Prefix	0x6352#3	INT8	ro	-	INT8	-3
21#160	Y	cmpsplcusogn	Moog DCV	ValvePositionControl, CustomerOverallGain CustomerOverallGain	0x241F#0	FLOAT32	rw	Y	0.00…2.00	1.00
21#235	Y	spldemplt	Moog DCV	ValvePositionControl, DemandValvePilot DemandValvePilot	0x3300#0	INT16	ro	-	INT16	-
21#236	Y	splvalplt	Moog DCV	ValvePositionControl, ActualValvePilot ActualValvePilot	0x3301#0	INT16	ro	-	INT16	-
21#237	Y	stgposout	Moog DCV	ValveMainStageControl, ControllerOutput ControllerOutput	0x2158#0	INT16	ro	-	INT16	-
21#238	N	stgitfnum	Moog DCV	MainStageControl, MainStageInterface MainStageInterface	0x2149#0	UINT8	rw	Y	0…8	-

Table 104: Object dictionary (part 16 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
21#239	Y	stgsplval	Moog DCV	MainStageControl, MainStagePositionActualValue MainStagePositionActualValue	0x215B#1	INT16	ro	-	INT16	-
21#240	N	stgposcusogn	Moog DCV	MainStageControl, MainStageCustomerOverallGain MainStageCustomerOverallGain	0x215C#0	FLOAT32	rw	Y	0.00…2.00	1.00
22#21	Y	prsset	CiA 408	ValvePressureControl, PrsSetpoint PrsSetpoint	0x6380#1	INT16	rw	N	INT16	-
22#22	N	prsuni	CiA 408	ValvePressureControl, PrsSetpoint Unit	0x6380#2	UINT8	ro	-	UINT8	0
22#23	N	prsprf	CiA 408	ValvePressureControl, PrsSetpoint Prefix	0x6380#3	INT8	ro	-	INT8	0
22#24	Y	prsdem	CiA 408	ValvePressureControl_DemandValueGenerator, PrsDemandValue PrsDemandValue	0x6390#1	INT16	ro	-	INT16	-
22#25	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator, PrsDemandValue Unit	0x6390#2	UINT8	ro	-	UINT8	0
22#26	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator, PrsDemandValue Prefix	0x6390#3	INT8	ro	-	INT8	0
22#27	Y	prsdemref	CiA 408	ValvePressureControl_DemandValueGenerator, PrsReferenceValue PrsReferenceValue	0x6391#1	INT16	ro	-	INT16	16384
22#28	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator, PrsReferenceValue Unit	0x6391#2	UINT8	ro	-	UINT8	0
22#29	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator, PrsReferenceValue Prefix	0x6391#3	INT8	ro	-	INT8	0
22#30	Y	prssethld	CiA 408	ValvePressureControl_DemandValueGenerator, PrsHoldSetpoint PrsHoldSetpoint	0x6394#1	INT16	rw	Y	INT16	-
22#31	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator, PrsHoldSetpoint Unit	0x6394#2	UINT8	ro	-	UINT8	0
22#32	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator, PrsHoldSetpoint Prefix	0x6394#3	INT8	ro	-	INT8	0
22#33	Y	prslimupp	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, UpperLimit UpperLimit	0x63A0#1	INT16	rw	Y	<LowerLimit> (0x63A1)…32767	16384
22#34	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, UpperLimit Unit	0x63A0#2	UINT8	ro	-	UINT8	0
22#35	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, UpperLimit Prefix	0x63A0#3	INT8	ro	-	INT8	0
22#36	Y	prslimlow	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, LowerLimit LowerLimit	0x63A1#1	INT16	rw	Y	-32768…<UpperLimit> (0x63A0)	-16384
22#37	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, LowerLimit Unit	0x63A1#2	UINT8	ro	-	UINT8	0

Table 104: Object dictionary (part 17 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
22#38	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, LowerLimit Prefix	0x63A1#3	INT8	ro	-	INT8	0
22#39	Y	prsdemfct	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Factor Factor	0x63A2#0	UINT32	rw	Y	UINT32	0x10001
22#40	Y	prsdemofs	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Offset Offset	0x63A3#1	INT16	rw	Y	INT16	-
22#41	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Offset Unit	0x63A3#2	UINT8	ro	-	UINT8	0
22#42	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Offset Prefix	0x63A3#3	INT8	ro	-	INT8	0
22#43	Y	prsrmptyp	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Type Type	0x63B0#0	INT8	rw	Y	0…3	-
22#44	Y	prsrmpacl	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Accelera- tionTime AccelerationTime	0x63B1#1	UINT16	rw	Y	UINT16	-
22#45	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Accelera- tionTime Unit	0x63B1#2	UINT8	ro	-	UINT8	3
22#46	Y	prsrmpaclprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Accelera- tionTime AccelerationTime_Prefix	0x63B1#3	INT8	rw	Y	-4…0	-3
22#47	Y	prsrmpaclneg	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Accelera- tionTimeNegative AccelerationTimeNegative	0x63B3#1	UINT16	rw	Y	UINT16	-
22#48	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Accelera- tionTimeNegative Unit	0x63B3#2	UINT8	ro	-	UINT8	3
22#49	Y	prsrmpaclnegprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Accelera- tionTimeNegative AccelerationTimeNegative_Prefix	0x63B3#3	INT8	rw	Y	-4…0	-3
22#50	Y	prsrmpaclpos	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Accelera- tionTimePositive AccelerationTimePositive	0x63B2#1	UINT16	rw	Y	UINT16	-
22#51	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Accelera- tionTimePositive Unit	0x63B2#2	UINT8	ro	-	UINT8	3
22#52	Y	prsrmpaclposprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Accelera- tionTimePositive AccelerationTimePositive_Prefix	0x63B2#3	INT8	rw	Y	-4…0	-3

Table 104: Object dictionary (part 18 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
22#53	Y	prsrmpdcl	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTime DecelerationTime	0x63B4#1	UINT16	rw	Y	UINT16	-
22#54	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTime Unit	0x63B4#2	UINT8	ro	-	UINT8	3
22#55	Y	prsrmpdclprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTime DecelerationTime_Prefix	0x63B4#3	INT8	rw	Y	-4…0	-3
22#56	Y	prsrmpdclneg	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimeNegative DecelerationTimeNegative	0x63B6#1	UINT16	rw	Y	UINT16	-
22#57	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimeNegative Unit	0x63B6#2	UINT8	ro	-	UINT8	3
22#58	Y	prsrmpdclnegprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimeNegative DecelerationTimeNegative_Prefix	0x63B6#3	INT8	rw	Y	-4…0	-3
22#59	Y	prsrmpdclpos	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimePositive DecelerationTimePositive	0x63B5#1	UINT16	rw	Y	UINT16	-
22#60	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimePositive Unit	0x63B5#2	UINT8	ro	-	UINT8	3
22#61	Y	prsrmpdclposprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimePositive DecelerationTimePositive_Prefix	0x63B5#3	INT8	rw	Y	-4…0	-3
22#144	Y	prsval	CiA 408	ValvePressureControl, PrsActualValue PrsActualValue	0x6381#1	INT16	ro	-	INT16	-
22#145	N	prsuni	CiA 408	ValvePressureControl, PrsActualValue Unit	0x6381#2	UINT8	ro	-	UINT8	0
22#146	N	prsprf	CiA 408	ValvePressureControl, PrsActualValue Prefix	0x6381#3	INT8	ro	-	INT8	0
22#147	Y	prsctldvn	CiA 408	ValvePressureControl, PrsControlDeviation PrsControlDeviation	0x63D0#1	INT16	ro	-	INT16	-
22#148	N	prsuni	CiA 408	ValvePressureControl, PrsControlDeviation Unit	0x63D0#2	UINT8	ro	-	UINT8	0
22#149	N	prsprf	CiA 408	ValvePressureControl, PrsControlDeviation Prefix	0x63D0#3	INT8	ro	-	INT8	0

Table 104: Object dictionary (part 19 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
22#150	Y	prsmontyp	CiA 408	ValvePressureControl_ControlMonitoring, Type Type	0x63D1#0	INT8	rw	Y	0…1	-
22#151	Y	prsmonupp	CiA 408	ValvePressureControl_ControlMonitoring, UpperThreshold UpperThreshold	0x63D4#1	INT16	rw	Y	INT16	512
22#152	N	prsuni	CiA 408	ValvePressureControl_ControlMonitoring, UpperThreshold Unit	0x63D4#2	UINT8	ro	-	UINT8	0
22#153	N	prsprf	CiA 408	ValvePressureControl_ControlMonitoring, UpperThreshold Prefix	0x63D4#3	INT8	ro	-	INT8	0
22#154	Y	prsmonlow	CiA 408	ValvePressureControl_ControlMonitoring, LowerThreshold LowerThreshold	0x63D5#1	INT16	rw	Y	INT16	-512
22#155	N	prsuni	CiA 408	ValvePressureControl_ControlMonitoring, LowerThreshold Unit	0x63D5#2	UINT8	ro	-	UINT8	0
22#156	N	prsprf	CiA 408	ValvePressureControl_ControlMonitoring, LowerThreshold Prefix	0x63D5#3	INT8	ro	-	INT8	0
22#157	Y	prsmontim	CiA 408	ValvePressureControl_ControlMonitoring, DelayTime DelayTime	0x63D2#1	UINT16	rw	Y	UINT16	30
22#158	N	timuni	CiA 408	ValvePressureControl_ControlMonitoring, DelayTime Unit	0x63D2#2	UINT8	ro	-	UINT8	3
22#159	N	timprf	CiA 408	ValvePressureControl_ControlMonitoring, DelayTime Prefix	0x63D2#3	INT8	ro	-	INT8	-3
64#0	N	promodide	Moog DCV	Profibus, ModuleIdentifier ModuleIdentifier	0x4446#0	UINT8	rw	Y	1…126	125
64#1	N	probdr	Moog DCV	Profibus, Bitrate Bitrate	0x4447#0	UINT32	ro	-	0…12000000	-
64#2	N	procfglen	Moog DCV	Profibus, TelegramConfgurationLength TelegramConfgurationLength	0x444D#0	UINT8	ro	-	UINT8	-
64#3…9	N	procfgio[0…6]	Moog DCV	Profibus, TelegramConfigurationByte TelegramConfigurationByte_1	0x444E#1…7	UINT8	ro	-	UINT8	-
64#10	N	proprmchn	Moog DCV	Profibus, ParameterChannelActive ParameterChannelActive	0x4454#0	UINT8	rw	Y	0…1	-
64#11	Y	vpcsts	Moog DCV	Profibus, VPC3+b_Status VPC3+b_Status	0x4448#0	UINT16	ro	-	UINT16	-
64#12	N	prodpv1mod	Moog DCV	Profibus, DPV1Status DPV1Status	0x444B#0	UINT8	ro	-	0…1	-
64#13	N	ideobj[0]	CiA 301	Device, IdentityObject VendorId	0x1018#1	UINT32	ro	-	0x28…0x28	0x28
64#14	N	ideobj[1]	CiA 301	Device, IdentityObject ProductCode	0x1018#2	UINT32	ro	-	0…0xFFFFFFFF	0

Table 104: Object dictionary (part 20 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
64#15	N	ideobj[2]	CiA 301	Device, IdentityObject RevisionNumber	0x1018#3	UINT32	ro	-	0…0xFFFFFFFF	0
64#16	N	ideobj[3]	CiA 301	Device, IdentityObject SerialNumber	0x1018#4	UINT32	ro	-	0…0xFFFFFFFF	0
64#17…32	N	prosigpar[0…15]	Moog DCV	Profibus, SignalParameterSelection SignalParameterSelection	0x4456#1…16	UINT32	rw	Y	UINT32	-
64#33	N	pronodide	Moog DCV	Profibus, ActualNodeIdentifier ActualNodeIdentifier	0x4449#0	UINT8	ro	-	1…126	125
64#34	N	progct	Moog DCV	Profibus, LastGlobalControlTelegram LastGlobalControlTelegram	0x4457#0	UINT8	ro	-	UINT8	-
64#35	N	preerrnum	CiA 301	Device, NumberOfErrors NumberOfErrors	0x1003#0	UINT32	rw	N	UINT32	0
64#36…51	N	preerrfld[0…15]	CiA 301	Device, StandardErrorField StandardErrorField	0x1003#1…16	UINT32	ro	-	UINT32	-
64#52	N	manstsreg	CiA 301	Device, ManufacturerStatusRegister ManufacturerStatusRegister	0x1002#0	UINT32	ro	-	UINT32	-
64#53	N	errreg	CiA 301	Device, ErrorRegister ErrorRegister	0x1001#0	UINT8	ro	-	UINT8	-
66#1…16	Y	cmpprstyp[0…15]	Moog DCV	ValvePressureControl, PressureControllerType PressureControllerType	0x2300#1…16	UINT8	rw	Y	UINT8	-
66#17…32	N	cmpprsitf[0…15]	Moog DCV	ValvePressureControl, ActiveTransducerInterfaceAreaA ActiveTransducerInterfaceAreaA	0x230D#1…16	INT8	rw	Y	1…8	1
66#33…48	N	cmpprsitfb[0…15]	Moog DCV	ValvePressureControl, ActiveTransducerInterfaceAreaB ActiveTransducerInterfaceAreaB	0x230F#1…16	INT8	rw	Y	0…8	-
66#49…64	N	cmpprsprs[0…15]	Moog DCV	ValvePressureControl, SystemPressure SystemPressure	0x2301#1…16	INT16	rw	Y	INT16	-
66#65…80	N	cmpprspbr[0…15]	Moog DCV	ValvePressureControl, ReferencePressure ReferencePressure	0x2302#1…16	INT16	rw	Y	INT16	-
66#81…96	Y	cmpprschy[0…15]	Moog DCV	ValvePressureControl, HydraulicCapacity HydraulicCapacity	0x230C#1…16	FLOAT32	rw	Y	0.00…+inf	-
66#97…112	Y	cmpprsrmp[0…15]	Moog DCV	ValvePressureControl, RampSlope RampSlope	0x2303#1…16	UINT16	rw	Y	UINT16	-
65#113…128	Y	cmpprsfb- spgn[0…15]	Moog DCV	PressureControl, SpoolPositionFeedBackGain SpoolPositionFeedBackGain	0x2324#1…16	FLOAT32	rw	Y	0.00…+inf	-
65#145…160	Y	cmpprsfbsp- gn_2[0…15]	Moog DCV	PressureControl, SpoolPositionFeedBackGain_2 SpoolPositionFeedBackGain_2	0x5858#1…16	FLOAT32	rw	Y	0.00…+inf	-
66#113…128	Y	cmpprspgn[0…15]	Moog DCV	ValvePressureControl, ProportionalGain ProportionalGain	0x2304#1…16	FLOAT32	rw	Y	0.00…+inf	-

Table 104: Object dictionary (part 21 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
66#129…144	Y	cmpprsptm[0…15]	Moog DCV	ValvePressureControl, ProportionalGainTimeConstant ProportionalGainTimeConstant	0x230E#1…16	FLOAT32	rw	Y	0.00…+inf	-
66#145…160	Y	cmpprsign[0…15]	Moog DCV	ValvePressureControl, IntegratorGain IntegratorGain	0x2305#1…16	FLOAT32	rw	Y	0.00…+inf	-
66#161…176	Y	cmpprsigf[0…15]	Moog DCV	ValvePressureControl, IntegratorFactor IntegratorFactor	0x2306#1…16	FLOAT32	rw	Y	0.00…+inf	0.10
66#177…192	Y	cmpprsicr[0…15]	Moog DCV	ValvePressureControl, IntegratorControlRange IntegratorControlRange	0x2307#1…16	INT16	rw	Y	0…32767	163
66#193…208	Y	cmpprsdgn[0…15]	Moog DCV	ValvePressureControl, DifferentiatorGain DifferentiatorGain	0x2308#1…16	FLOAT32	rw	Y	FLOAT32	-
66#209…224	N	cmppr-signswitrs[0…15]	Moog DCV	ValvePressureControl, IntegratorGainSwitchThreshold IntegratorGainSwitchThreshold	0x5857#1…16	UINT32	rw	Y	UINT32	5000
66#225…240	Y	cmpprsdtm[0…15]	Moog DCV	ValvePressureControl, DifferentiatorT1 DifferentiatorT1	0x2309#1…16	FLOAT32	rw	Y	0.00…+inf	-
66#241	Y	cmpprsint	Moog DCV	ValvePressureControl, KiOutput KiOutput	0x2310#0	FLOAT32	ro	-	FLOAT32	-
66#242	Y	cmpprspro	Moog DCV	ValvePressureControl, KpT1Output KpT1Output	0x2311#0	FLOAT32	ro	-	FLOAT32	-
66#243	Y	cmpprsdt1	Moog DCV	ValvePressureControl, KdOutput KdOutput	0x2312#0	FLOAT32	ro	-	FLOAT32	-
66#244	Y	cmpprsdir	Moog DCV	ValvePressureControl, DirectionalDependentGain DirectionalDependentGain	0x2313#0	FLOAT32	rw	Y	0.00…+inf	1.00
66#246	Y	prssetnum	Moog DCV	ValvePressureControl, ActiveParameterSetNumber ActiveParameterSetNumber	0x2350#0	UINT8	rw	Y	1…16	1
66#247	Y	cmpprsout	Moog DCV	ValvePressureControl, ControllerOutput ControllerOutput	0x2418#0	INT16	ro	-	INT16	-
66#248	N	cmpfrcpst	Moog DCV	ValvePressureControl, CylinderPistonDiameter CylinderPistonDiameter	0x585F#0	FLOAT32	rw	Y	0.00…+inf	1000000.00
66#249	N	cmpfrcroda	Moog DCV	ValvePressureControl, CylinderRodDiameterA CylinderRodDiameterA	0x585D#0	FLOAT32	rw	Y	0.00…<CylinderPistonDi-ameter> (0x585F)	-
66#250	N	cmpfrcrodb	Moog DCV	ValvePressureControl, CylinderRodDiameterB CylinderRodDiameterB	0x585E#0	FLOAT32	rw	Y	0.00…<CylinderPistonDi-ameter> (0x585F)	-
67#1	N	cmpprsflt	Moog DCV	ValvePressureControl, ActualPressureFilterCutoffFrequency ActualPressureFilterCutoffFrequency	0x23F2#0	FLOAT32	rw	Y	0.00…3333.33	-
67#2	N	prsfltord	Moog DCV	ValvePressureControl, ActualPressureFilterOrder ActualPressureFilterOrder	0x23F3#0	UINT8	rw	Y	0…3	1
67#3	N	prsfltbi[0]	Moog DCV	ValvePressureControl, PressureValueFilterBCoeff PressureValueFilterBCoeff	0x23F0#1	FLOAT32	rw	Y	FLOAT32	-

Table 104: Object dictionary (part 22 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
67#4	N	prsfltbi[1]	Moog DCV	ValvePressureControl, PressureValueFilterBCoeff PressureValueFilterBCoeff	0x23F0#2	FLOAT32	rw	Y	FLOAT32	-
67#5	N	prsfltbi[2]	Moog DCV	ValvePressureControl, PressureValueFilterBCoeff PressureValueFilterBCoeff	0x23F0#3	FLOAT32	rw	Y	FLOAT32	-
67#6	N	prsfltbi[3]	Moog DCV	ValvePressureControl, PressureValueFilterBCoeff PressureValueFilterBCoeff	0x23F0#4	FLOAT32	rw	Y	FLOAT32	-
67#7	N	prsfltai[0]	Moog DCV	ValvePressureControl, PressureValueFilterACoeff PressureValueFilterACoeff	0x23F1#1	FLOAT32	rw	Y	FLOAT32	-
67#8	N	prsfltai[1]	Moog DCV	ValvePressureControl, PressureValueFilterACoeff PressureValueFilterACoeff	0x23F1#2	FLOAT32	rw	Y	FLOAT32	-
67#9	N	prsfltai[2]	Moog DCV	ValvePressureControl, PressureValueFilterACoeff PressureValueFilterACoeff	0x23F1#3	FLOAT32	rw	Y	FLOAT32	-
67#10	N	prsfltai[3]	Moog DCV	ValvePressureControl, PressureValueFilterACoeff PressureValueFilterACoeff	0x23F1#4	FLOAT32	rw	Y	FLOAT32	-
67#11	Y	prsintprlvar[0]	Moog DCV	ValvePressureControl, ValvePressureControl IntegratorPreloadValue_0	0x5860#1	INT16	ro	-	INT16	-
67#12	Y	prsintprlvar[1]	Moog DCV	ValvePressureControl, ValvePressureControl IntegratorPreloadValue_1	0x5860#2	INT16	ro	-	INT16	-
67#13	Y	cmpprsdt1_2	Moog DCV	ValvePressureControl, kdFeedbackOutput kdFeedbackOutput	0x5862#0	FLOAT32	ro	-	FLOAT32	-
67#17…32	Y	cmpprsiup[0…15]	Moog DCV	ValvePressureControl, IntegratorUpperOutputLimit IntegratorUpperOutputLimit	0x231A#1…16	INT16	rw	Y	<IntegratorLowerOutput-Limit> (0x231B[n])…32767	16384
67#33…48	Y	cmpprsilo[0…15]	Moog DCV	ValvePressureControl, IntegratorLowerOutputLimit IntegratorLowerOutputLimit	0x231B#1…16	INT16	rw	Y	-32768…<IntegratorUp-perOutputLimit> (0x231A[n])	-16384
67#49…64	Y	cmpprsupp[0…15]	Moog DCV	ValvePressureControl, UpperOutputLimit UpperOutputLimit	0x230A#1…16	INT16	rw	Y	<LowerOutputLimit> (0x230B[n])…32767	16384
67#65…80	Y	cmpprslow[0…15]	Moog DCV	ValvePressureControl, LowerOutputLimit LowerOutputLimit	0x230B#1…16	INT16	rw	Y	-32768…<UpperOutput-Limit> (0x230A[n])	-16384
67#81…96	Y	prsintfbapgn[0…15]	Moog DCV	ValvePressureControl, IntegralProportionalPartPGain IntegralProportionalPartPGain	0x5861#1…16	FLOAT32	rw	Y	FLOAT32	-
67#97…112	Y	cmpprs-dgn_2[0…15]	Moog DCV	ValvePressureControl, DifferentiatorGain_2 DifferentiatorGain_2	0x5863#1…16	FLOAT32	rw	Y	FLOAT32	-
67#113…128	Y	cmpprsdt-m_2[0…15]	Moog DCV	ValvePressureControl, DifferentiatorT1_2 DifferentiatorT1_2	0x5864#1…16	FLOAT32	rw	Y	0.00…+inf	-
67#129…144	Y	prsoutupp[0…15]	Moog DCV	ValvePressureControl, UpperControllerOutputLimit UpperControllerOutputLimit	0x5865#1…16	INT16	rw	Y	<LowerOutputLimit> (0x230B[n])…32767	16384

Table 104: Object dictionary (part 23 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
67#145…160	Y	prsoutlow[0…15]	Moog DCV	ValvePressureControl, LowerControllerOutputLimit LowerControllerOutputLimit	0x5866#1…16	INT16	rw	Y	-32768…<UpperOutput-Limit> (0x230A[n])	-16384
67#161…176	Y	prsfwp[0…15]	Moog DCV	ValvePressureControl, FeedForwardGain FeedForwardGain	0x5867#1…16	FLOAT32	rw	Y	FLOAT32	-
67#177…192	Y	prsffwpar[0…15]	Moog DCV	ValvePressureControl, FeedForwardParameter FeedForwardParameter	0x5868#1…16	UINT32	rw	Y	UINT32	0x63800110
67#193…208	Y	prsffwofs[0…15]	Moog DCV	ValvePressureControl, FeedForwardOffset FeedForwardOffset	0x5870#1…16	INT16	rw	Y	INT16	-
67#209…224	Y	splprsmod[0…15]	Moog DCV	ValvePressureControl, pQSwitchingMode pQSwitchingMode	0x586C#1…16	UINT8	rw	Y	0…2	-
67#241	Y	prsintprlpgn	Moog DCV	ValvePressureControl, IntegratorPreloadGain IntegratorPreloadGain	0x5869#0	FLOAT32	rw	Y	FLOAT32	-
67#242	Y	prsintprlpar	Moog DCV	ValvePressureControl, IntegratrPreloadParameter IntegratrPreloadParameter	0x586A#0	UINT32	rw	Y	UINT32	0x63800110
67#243	Y	prsintprlmod	Moog DCV	ValvePressureControl, IntegratorPreloadMode IntegratorPreloadMode	0x586B#0	UINT8	rw	Y	0…2	-
67#244	Y	prsdemsgnmod	Moog DCV	ValvePressureControl, PressureDemandSignMode PressureDemandSignMode	0x586D#0	UINT8	rw	Y	0…1	-
67#245	Y	dirparset[0]	Moog DCV	ValvePressureControl, ValvePressureControl DirectionalParameterSet_0	0x586E#1	UINT8	rw	Y	1…16	1
67#246	Y	dirparset[1]	Moog DCV	ValvePressureControl, ValvePressureControl DirectionalParameterSet_1	0x586E#2	UINT8	rw	Y	1…16	1
67#247	Y	dirparsetmod	Moog DCV	ValvePressureControl, DirectionalDependantParameterSetMode DirectionalDependantParameterSetMode	0x586F#0	UINT8	rw	Y	0…1	-
69#1	N	snssup	Moog DCV	DrivePositionControl, SensorSupplyEnable SensorSupplyEnable	0x5619#0	UINT8	rw	Y	UINT8	-
69#2	N	refvel	Moog DCV	DrivePositionControl, ReferencingVelocity ReferencingVelocity	0x5614#0	INT32	rw	Y	1000…2147483647	1000
69#3	N	reffrc	Moog DCV	DrivePositionControl, ReferencingForce ReferencingForce	0x561A#0	INT16	rw	Y	INT16	-
69#4	Y	drvsts	Moog DCV	Drive, DriveStatus DriveStatus	0x561C#0	UINT32	rw	N	UINT32	-
69#5	N	zplstrg	Moog DCV	DrivePositionControl, ZPulseTrigger ZPulseTrigger	0x5617#0	UINT8	rw	Y	0…1	-
69#6	Y	zplsdet	Moog DCV	DrivePositionControl, ZPulseDetected ZPulseDetected	0x5611#0	UINT8	rw	N	0…1	-
69#7	N	refstp	Moog DCV	DrivePositionControl, ReferencingStop ReferencingStop	0x561B#0	FLOAT32	rw	Y	FLOAT32	-

Table 104: Object dictionary (part 24 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
69#8	Y	zplsena	Moog DCV	DrivePositionControl, ZPulseEnable ZPulseEnable	0x5610#0	UINT8	rw	N	0…1	-
69#9	Y	zplsclr	Moog DCV	DrivePositionControl, ZPulseClear ZPulseClear	0x5618#0	UINT8	rw	N	UINT8	-
69#10	N	encraw	Moog DCV	DrivePositionControl, EncoderValue EncoderValue	0x5613#0	INT32	rw	N	INT32	-
69#11	Y	prswin	Moog DCV	DrivePositionControl, PressureWindow PressureWindow	0x5710#0	INT16	rw	Y	0…32767	-
69#12	Y	prsmaxwin	Moog DCV	DrivePositionControl, MaximumPressureWindow MaximumPressureWindow	0x5711#0	INT16	rw	Y	0…32767	-
69#13	Y	drvsts16	Moog DCV	Drive, 16BitDriveStatus 16BitDriveStatus	0x561D#0	UINT16	rw	N	UINT16	-
69#14	Y	zplsset	Moog DCV	DrivePositionControl, ZPulseSet ZPulseSet	0x5612#0	INT32	rw	Y	INT32	-
69#15	N	encrawset	Moog DCV	DrivePositionControl, EncoderSetValue EncoderSetValue	0x5621#0	INT32	rw	N	INT32	-
70#1	N	cmpprsitfps	Moog DCV	ValveFlowControl, PressureControllerTransducerInterfacePS PressureControllerTransducerInterfacePS	0x2330#0	INT8	rw	Y	0…8	-
70#2	Y	flwset	Moog DCV	ValveFlowControl, FlowSetpoint FlowSetpoint	0x5300#1	INT16	rw	N	INT16	-
70#3	N	flwuni	Moog DCV	ValveFlowControl, FlowSetpoint Unit	0x5300#2	UINT8	ro	-	UINT8	0
70#4	N	flwprf	Moog DCV	ValveFlowControl, FlowSetpoint Prefix	0x5300#3	INT8	ro	-	INT8	0
70#5	Y	flwdem	Moog DCV	ValveFlowControl, FlowDemand FlowDemand	0x5213#0	INT16	ro	-	INT16	-
70#6	Y	flwsethld	Moog DCV	ValveFlowControl, FlowHoldSetPoint FlowHoldSetPoint	0x5314#1	INT16	rw	Y	INT16	-
70#7	N	flwuni	Moog DCV	ValveFlowControl, FlowHoldSetPoint Unit	0x5314#2	UINT8	ro	-	UINT8	0
70#8	N	flwprf	Moog DCV	ValveFlowControl, FlowHoldSetPoint Prefix	0x5314#3	INT8	ro	-	INT8	0
70#10	Y	flwbrncor	Moog DCV	ValveFlowControl, BernoulliCorrection BernoulliCorrection	0x5200#1	FLOAT32	ro	-	FLOAT32	-
70#12	N	flwpar	Moog DCV	ValveFlowControl, SetpointParameter SetpointParameter	0x5202#0	UINT32	ro	-	UINT32	0x53000110
70#13	N	prspsnom	Moog DCV	ValveFlowControl, NominalSupplyPressure NominalSupplyPressure	0x5203#0	INT16	rw	Y	1…32767	400

Table 104: Object dictionary (part 25 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name, Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
70#14	N	prsflwtim[0]	Moog DCV	ValveFlowControl, ValveFlowControl AntiOscillationWaitTime_0	0x5204#1	FLOAT32	rw	Y	0.00…+inf	30.00
70#15	N	prsflwtim[1]	Moog DCV	ValveFlowControl, ValveFlowControl AntiOscillationWaitTime_1	0x5204#2	FLOAT32	rw	Y	0.00…+inf	3.00
70#16	N	flwmod	Moog DCV	ValveFlowControl, FlowControlMode FlowControlMode	0x5205#0	UINT8	rw	Y	UINT8	-
70#17	N	flwprspeatrs	Moog DCV	ValveFlowControl, PressurePeakThreshold PressurePeakThreshold	0x520D#0	INT16	rw	Y	0…32767	20000
70#18	Y	flwvlvdir	Moog DCV	ValveFlowControl, ValveDirection ValveDirection	0x520E#0	INT8	rw	Y	-1…1	1
70#19	N	flwctlout	Moog DCV	ValveFlowControl, ControllerOutput ControllerOutput	0x520F#0	INT16	ro	-	INT16	-
70#20	Y	flwprsfdbflt	Moog DCV	ValveFlowControl, PressureFeedbackFilterPT1TimeConstant PressureFeedbackFilterPT1TimeConstant	0x5210#0	FLOAT32	rw	Y	0.00…+inf	-
70#21	Y	flwctlsts	Moog DCV	ValveFlowControl, FlowActualStatus FlowActualStatus	0x5212#0	UINT8	ro	-	UINT8	-
70#22	N	vlvqan	Moog DCV	ValveFlowControl, RatedValveFlowASide RatedValveFlowASide	0x5215#0	FLOAT32	rw	Y	0.01…+inf	1.00
70#23	N	vlvqbn	Moog DCV	ValveFlowControl, RatedValveFlowBSide RatedValveFlowBSide	0x5216#0	FLOAT32	rw	Y	0.01…+inf	1.00
70#24	N	cmpprsitfpt	Moog DCV	ValveFlowControl, PressureControllerTransducerInterfacePT PressureControllerTransducerInterfacePT	0x5217#0	INT8	rw	Y	0…8	-
70#25	N	flwalpha	Moog DCV	ValveFlowControl, Alpha Alpha	0x5219#0	FLOAT32	ro	-	FLOAT32	-
70#26	N	flwbeta	Moog DCV	ValveFlowControl, Beta Beta	0x521A#0	FLOAT32	ro	-	FLOAT32	-
70#27	N	flwsetsca	Moog DCV	ValveFlowControl, SetpointScaling SetpointScaling	0x521B#0	FLOAT32	rw	N	-1.00…1.00	1.00
70#28	N	prsflwctlout	Moog DCV	ValveFlowControl, P_Flow_ControllerOutput P_Flow_ControllerOutput	0x521C#0	INT16	ro	-	INT16	-
70#29	N	prssensmax	Moog DCV	ValveFlowControl, PressureSensorMaximum PressureSensorMaximum	0x521D#0	INT16	ro	-	INT16	-
70#30	N	vlvdpn	Moog DCV	ValveFlowControl, RatedValvePressureDrop RatedValvePressureDrop	0x521E#0	INT16	rw	Y	1…32767	400
70#31	N	flwqamax	Moog DCV	ValveFlowControl, MaximalFlowQAMax MaximalFlowQAMax	0x521F#0	FLOAT32	ro	-	FLOAT32	-
70#32	N	flwfrcfhi	Moog DCV	ValveFlowControl, ForceHightPassCutOffFreq ForceHightPassCutOffFreq	0x5220#0	FLOAT32	rw	Y	FLOAT32	20.00

Table 104: Object dictionary (part 26 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
70#33	Y	flwfrcgn	Moog DCV	ValveFlowControl, ForceGain ForceGain	0x5221#0	FLOAT32	rw	Y	FLOAT32	1.00
70#34	N	flwfdbpar	Moog DCV	ValveFlowControl, flowFeedbackSource flowFeedbackSource	0x5222#0	UINT32	rw	Y	UINT32	0x52230010
70#35	N	flwfrc	Moog DCV	ValveFlowControl, ForceFeedback ForceFeedback	0x5223#0	INT16	ro	-	INT16	-
70#36	N	flwfdbgn	Moog DCV	ValveFlowControl, FlowFeedbackGain FlowFeedbackGain	0x5224#0	FLOAT32	rw	Y	FLOAT32	-
71#1	N	dlgctl	Moog DCV	DataLogger, Control Control	0x3180#0	UINT8	rw	N	0…1	-
71#2	N	dlgsts	Moog DCV	DataLogger, Status Status	0x3181#0	UINT8	ro	-	0…3	-
71#3	N	dlgdiv	Moog DCV	DataLogger, Divider Divider	0x3182#0	UINT16	rw	Y	1…65535	1
71#4	N	dlgsmp	Moog DCV	DataLogger, NumberOfSamples NumberOfSamples	0x3183#0	INT32	ro	-	-2147483648…2048	-
71#5	N	dlgena[0]	Moog DCV	DataLogger, EnableChannel EnableChannel1	0x3184#1	UINT8	rw	Y	UINT8	-
71#6	N	dlgena[1]	Moog DCV	DataLogger, EnableChannel EnableChannel2	0x3184#2	UINT8	rw	Y	UINT8	-
71#7	N	dlgena[2]	Moog DCV	DataLogger, EnableChannel EnableChannel3	0x3184#3	UINT8	rw	Y	UINT8	-
71#8	N	dlgena[3]	Moog DCV	DataLogger, EnableChannel EnableChannel4	0x3184#4	UINT8	rw	Y	UINT8	-
71#9	N	dlgpar[0]	Moog DCV	DataLogger, ChannelParameter ChannelParameter1	0x3185#1	UINT32	rw	Y	UINT32	0x63100110
71#10	N	dlgpar[1]	Moog DCV	DataLogger, ChannelParameter ChannelParameter2	0x3185#2	UINT32	rw	Y	UINT32	0x63010110
71#11	N	dlgpar[2]	Moog DCV	DataLogger, ChannelParameter ChannelParameter3	0x3185#3	UINT32	rw	Y	UINT32	0x63900110
71#12	N	dlgpar[3]	Moog DCV	DataLogger, ChannelParameter ChannelParameter4	0x3185#4	UINT32	rw	Y	UINT32	0x63810110
71#21	N	dlgofs	Moog DCV	DataLogger, SampleStartOffset SampleStartOffset	0x3187#0	UINT32	ro	-	UINT32	-
71#22	N	trgtyp	Moog DCV	DataLogger, TriggerType TriggerType	0x3188#0	UINT8	rw	Y	0…2	1
71#23	N	trgpar	Moog DCV	DataLogger, TriggerParameter TriggerParameter	0x3189#0	UINT32	rw	Y	UINT32	0x63100110

Table 104: Object dictionary (part 27 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#24	N	trgcpl	Moog DCV	DataLogger, TriggerCoupling TriggerCoupling	0x318A#0	UINT8	rw	Y	0…2	1
71#25	N	trgslp	Moog DCV	DataLogger, TriggerSlope TriggerSlope	0x318B#0	UINT8	rw	Y	1…3	1
71#26	N	trglvl	Moog DCV	DataLogger, TriggerLevelOrBitmask TriggerLevelOrBitmask	0x318C#0	INT32	rw	Y	INT32	-
71#27	N	trgpos	Moog DCV	DataLogger, TriggerPosition TriggerPosition	0x318D#0	INT32	rw	Y	INT32	-
71#28	N	trgtim	Moog DCV	DataLogger, TriggerTimeStamp TriggerTimeStamp	0x318E#0	UINT32	ro	-	UINT32	-
71#30	N	errflg	Moog DCV	FaultReaction, CustomerDefinedErrorFlag CustomerDefinedErrorFlag	0x2900#0	INT8	rw	N	0…127	-
71#31	N	ev0exp	Moog DCV	Eventhandler, EventExpression_1 EventExpression_1	0x2901#0	CHAR[192]	rw	Y	-	-
71#32	N	ev1exp	Moog DCV	Eventhandler, EventExpression_2 EventExpression_2	0x2902#0	CHAR[192]	rw	Y	-	-
71#33	N	ev2exp	Moog DCV	Eventhandler, EventExpression_3 EventExpression_3	0x2903#0	CHAR[192]	rw	Y	-	-
71#34	N	ev3exp	Moog DCV	Eventhandler, EventExpression_4 EventExpression_4	0x2904#0	CHAR[192]	rw	Y	-	-
71#35	N	ev4exp	Moog DCV	Eventhandler, EventExpression_5 EventExpression_5	0x2905#0	CHAR[192]	rw	Y	-	-
71#36	N	ev5exp	Moog DCV	Eventhandler, EventExpression_6 EventExpression_6	0x2906#0	CHAR[192]	rw	Y	-	-
71#37	N	ev6exp	Moog DCV	Eventhandler, EventExpression_7 EventExpression_7	0x2907#0	CHAR[192]	rw	Y	-	-
71#38	N	ev7exp	Moog DCV	Eventhandler, EventExpression_8 EventExpression_8	0x2908#0	CHAR[192]	rw	Y	-	-
71#39…46	Y	evtena[0…7]	Moog DCV	Eventhandler, EventEnable EventEnable_1	0x2909#1…8	UINT8	rw	Y	UINT8	-
71#48…55	Y	vars08[0…7]	Moog DCV	Eventhandler, Integer08 Integer08_1	0x290B#1…8	INT8	rw	N	INT8	-
71#56…63	Y	vars16[0…7]	Moog DCV	Eventhandler, Integer16 Integer16_1	0x290C#1…8	INT16	rw	N	INT16	-
71#64…71	Y	vars32[0…7]	Moog DCV	Eventhandler, Integer32 Integer32_1	0x290D#1…8	INT32	rw	N	INT32	-
71#72…79	Y	varu08[0…7]	Moog DCV	Eventhandler, Unsigned08 Unsigned08_1	0x290E#1…8	UINT8	rw	N	UINT8	-

Table 104: Object dictionary (part 28 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#80…87	Y	varu16[0…7]	Moog DCV	Eventhandler, Unsigned16 Unsigned16_1	0x290F#1…8	UINT16	rw	N	UINT16	-
71#88…95	Y	varu32[0…7]	Moog DCV	Eventhandler, Unsigned32 Unsigned32_1	0x2910#1…8	UINT32	rw	N	UINT32	-
71#104	Y	trjjmx	Moog DCV	DrivePositionControl, MaximumJerk MaximumJerk	0x5701#0	INT32	rw	Y	1…2147483647	10000000
71#105	Y	trjamx	Moog DCV	DrivePositionControl, MaximumAcceleration MaximumAcceleration	0x5702#0	INT32	rw	Y	1…2147483647	1000000
71#106	Y	trjvmx	Moog DCV	DrivePositionControl, MaximumVelocity MaximumVelocity	0x5703#0	INT32	rw	Y	1…2147483647	100000
71#107	Y	trjtyp	Moog DCV	DrivePositionControl, TrajectoryType TrajectoryType	0x5700#0	INT8	rw	Y	0…2	-
71#108	Y	trjcfgwng[0]	Moog DCV	DrivePositionControl, DrivePositionControl_Trajectory ConfigWarning_0	0x5705#1	UINT8	ro	-	UINT8	-
71#109	Y	trjcfgwng[1]	Moog DCV	DrivePositionControl, DrivePositionControl_Trajectory ConfigWarning_1	0x5705#2	UINT8	ro	-	UINT8	-
71#110	Y	trjcfgwng[2]	Moog DCV	DrivePositionControl, DrivePositionControl_Trajectory ConfigWarning_2	0x5705#3	UINT8	ro	-	UINT8	-
71#111	Y	dums08	CiA 301	DataType, DummyDataS08 DummyDataS08	0x0002#0	INT8	rw	N	INT8	-
71#112	Y	dums16	CiA 301	DataType, DummyDataS16 DummyDataS16	0x0003#0	INT16	rw	N	INT16	-
71#113	Y	dums32	CiA 301	DataType, DummyDataS32 DummyDataS32	0x0004#0	INT32	rw	N	INT32	-
71#114	Y	dumu08	CiA 301	DataType, DummyDataU08 DummyDataU08	0x0005#0	UINT8	rw	N	UINT8	-
71#115	Y	dumu16	CiA 301	DataType, DummyDataU16 DummyDataU16	0x0006#0	UINT16	rw	N	UINT16	-
71#116	Y	dumu32	CiA 301	DataType, DummyDataU32 DummyDataU32	0x0007#0	UINT32	rw	N	UINT32	-
71#117	Y	dumf32	CiA 301	DataType, DummyDataF32 DummyDataF32	0x0008#0	FLOAT32	rw	N	FLOAT32	-
71#118	Y	dumchr	CiA 301	DataType, DummyDataVisibleString DummyDataVisibleString	0x0009#0	CHAR[64]	rw	N	-	-
71#120	Y	fcntyp	Moog DCV	FunctionGenerator, FunctionGenType FunctionGenType	0x3100#0	INT8	rw	N	0…5	-
71#121	Y	fcndem	Moog DCV	FunctionGenerator, FunctionGenOutput FunctionGenOutput	0x3101#0	INT16	ro	-	INT16	-

Table 104: Object dictionary (part 29 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
71#122	Y	fcnsqr	Moog DCV	FunctionGenerator, FunctionGenSquareOutput FunctionGenSquareOutput	0x3102#0	INT16	ro	-	INT16	-
71#123	Y	fcntim	Moog DCV	FunctionGenerator, FunctionGenFrequency FunctionGenFrequency	0x3103#0	UINT16	rw	N	1…<LvdtFrequency> (0x3030)	10
71#124	Y	fcnmag	Moog DCV	FunctionGenerator, FunctionGenMagnitude FunctionGenMagnitude	0x3104#0	INT16	rw	N	0…32767	-
71#125	Y	fcnofs	Moog DCV	FunctionGenerator, FunctionGenOffset FunctionGenOffset	0x3105#0	INT16	rw	N	INT16	-
71#126	Y	fcnsgn	Moog DCV	FunctionGenerator, FunctionGenSign FunctionGenSign	0x3107#0	INT8	rw	N	-1…1	1
71#127	Y	fcnprf	Moog DCV	FunctionGenerator, FunctionGenFrequencyPrefix FunctionGenFrequencyPrefix	0x3108#0	INT8	rw	N	-4…0	-
72#1	N	devmdlurl	CiA 408	Device, ModelURL ModelURL	0x6055#0	CHAR[64]	ro	-	-	www.moog.com
72#2	Y	devprmcod	CiA 408	Device, ParameterSetCode ParameterSetCode	0x6056#0	UINT8	rw	Y	0…254	-
72#9	N	pcbtmpmaxcus	Moog DCV	Hardware_DiagnosticData, PcbMaxTemperatureCustomer PcbMaxTemperatureCustomer	0x2809#0	INT16	rw	Y	INT16	-
72#10	Y	cpusup	Moog DCV	Hardware_DiagnosticData, CpuSupplyVoltage CpuSupplyVoltage	0x2803#0	UINT16	ro	-	UINT16	-
72#11	Y	pwrsup	Moog DCV	Hardware_DiagnosticData, PowerSupplyVoltage PowerSupplyVoltage	0x2804#0	UINT16	ro	-	UINT16	-
72#12	Y	pcbtmp	Moog DCV	Hardware_DiagnosticData, PcbTemperature PcbTemperature	0x2805#0	INT16	ro	-	INT16	-
72#13…17	N	errval[0…4]	Moog DCV	ErrorHandler, InternalErrorCode InternalErrorCode	0x2822#1…5	UINT32	ro	-	UINT32	-
72#18…22	N	errtim[0…4]	Moog DCV	ErrorHandler, InternalErrorTime InternalErrorTime	0x2823#1…5	UINT32	ro	-	UINT32	-
72#24	N	oprtim[0]	Moog DCV	Hardware_DiagnosticData, Hardware_DiagnosticData PowerOnTime	0x280D#1	UINT32	ro	-	UINT32	-
72#25	N	oprtim[1]	Moog DCV	Hardware_DiagnosticData, Hardware_DiagnosticData OperatingTime	0x280D#2	UINT32	ro	-	UINT32	-
72#26	Y	fausts[0]	Moog DCV	FaultReaction, FaultStatus FaultStatus	0x2831#1	UINT32	ro	-	UINT32	-
72#27	Y	fausts[1]	Moog DCV	FaultReaction, FaultStatus FaultStatus	0x2831#2	UINT32	ro	-	UINT32	-
72#28	Y	fausts[2]	Moog DCV	FaultReaction, FaultStatus FaultStatus	0x2831#3	UINT32	ro	-	UINT32	-

Table 104: Object dictionary (part 30 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
72#29	Y	fausts[3]	Moog DCV	FaultReaction, FaultStatus FaultStatus	0x2831#4	UINT32	ro	-	UINT32	-
72#40	N	faudsc	Moog DCV	FaultReaction, FaultReactionDescription FaultReactionDescription	0x2832#0	CHAR[64]	ro	-	-	-
72#41	N	fauhis	Moog DCV	FaultReaction, FaultHistoryNumber FaultHistoryNumber	0x2833#0	UINT8	rw	N	0…7	-
72#42	Y	faustsret[0]	Moog DCV	FaultReaction, FaultRetainStatus FaultRetainStatus	0x2834#1	UINT32	rw	N	UINT32	-
72#43	Y	faustsret[1]	Moog DCV	FaultReaction, FaultRetainStatus FaultRetainStatus	0x2834#2	UINT32	rw	N	UINT32	-
72#44	Y	faustsret[2]	Moog DCV	FaultReaction, FaultRetainStatus FaultRetainStatus	0x2834#3	UINT32	rw	N	UINT32	-
72#45	Y	faustsret[3]	Moog DCV	FaultReaction, FaultRetainStatus FaultRetainStatus	0x2834#4	UINT32	rw	N	UINT32	-
72#101…220	N	faurea[0…119]	Moog DCV	FaultReaction, FaultReactionType FaultReactionType	0x2830#1…120	INT8	rw	Y	INT8	0
73#1	N	locmodide	Moog DCV	LocalCAN, LocalCAN_ModuleIdentifier LocalCAN_ModuleIdentifier	0x5B00#0	UINT8	rw	Y	1…127	0x7F
73#2	N	locbdr	Moog DCV	LocalCAN, LocalCAN_Bitrate LocalCAN_Bitrate	0x5B01#0	UINT32	rw	Y	0…1000000	500000
73#3	Y	locsrn	Moog DCV	LocalCAN, LocalCAN_StartRemoteNode LocalCAN_StartRemoteNode	0x5B02#0	UINT8	rw	N	UINT8	-
73#4	N	locrempar	Moog DCV	LocalCAN, LocalCAN_RemoteParameter LocalCAN_RemoteParameter	0x5B10#0	UINT32	rw	N	UINT32	-
73#5	N	locremadr	Moog DCV	LocalCAN, LocalCAN_RemoteParameterAdress LocalCAN_RemoteParameterAdress	0x5B11#0	UINT32	rw	N	UINT32	-
73#6	N	locremnod	Moog DCV	LocalCAN, LocalCAN_RemoteNodeId LocalCAN_RemoteNodeId	0x5B12#0	UINT8	rw	N	0…127	-
73#7	N	locremtrn	Moog DCV	LocalCAN, LocalCAN_RemoteTransmission LocalCAN_RemoteTransmission	0x5B13#0	INT8	rw	N	-1…2	-
73#8	N	locpdrcob[0]	Moog DCV	LocalCAN, RPdo LocalCANRPdo1_CobIdUsedByPdo	0x5400#1	UINT32	rw	Y	1…0x800007FF	<NodeID> (0x100B) + 0x0200
73#9	N	locpdrcob[1]	Moog DCV	LocalCAN, RPdo LocalCANRPdo2_CobIdUsedByPdo	0x5401#1	UINT32	rw	Y	1…0x800007FF	<NodeID> (0x100B) + 0x0300
73#10	N	locpdrcob[2]	Moog DCV	LocalCAN, RPdo LocalCANRPdo3_CobIdUsedByPdo	0x5402#1	UINT32	rw	Y	1…0x800007FF	<NodeID> (0x100B) + 0x0400
73#11	N	locpdrcob[3]	Moog DCV	LocalCAN, RPdo LocalCANRPdo4_CobIdUsedByPdo	0x5403#1	UINT32	rw	Y	1…0x800007FF	<NodeID> (0x100B) + 0x0500

Table 104: Object dictionary (part 31 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
73#12	N	locpdrtrn[0]	Moog DCV	LocalCAN, RPdo LocalCANRPdo1_TransmissionType	0x5400#2	UINT8	rw	Y	UINT8	255
73#13	N	locpdrtrn[1]	Moog DCV	LocalCAN, RPdo LocalCANRPdo2_TransmissionType	0x5401#2	UINT8	rw	Y	UINT8	255
73#14	N	locpdrtrn[2]	Moog DCV	LocalCAN, RPdo LocalCANRPdo3_TransmissionType	0x5402#2	UINT8	rw	Y	UINT8	255
73#15	N	locpdrtrn[3]	Moog DCV	LocalCAN, RPdo LocalCANRPdo4_TransmissionType	0x5403#2	UINT8	rw	Y	UINT8	255
73#16	N	locpdrtim[0]	Moog DCV	LocalCAN, RPdo LocalCANRPdo1_EventTimer	0x5400#5	UINT16	rw	Y	UINT16	-
73#17	N	locpdrtim[1]	Moog DCV	LocalCAN, RPdo LocalCANRPdo2_EventTimer	0x5401#5	UINT16	rw	Y	UINT16	-
73#18	N	locpdrtim[2]	Moog DCV	LocalCAN, RPdo LocalCANRPdo3_EventTimer	0x5402#5	UINT16	rw	Y	UINT16	-
73#19	N	locpdrtim[3]	Moog DCV	LocalCAN, RPdo LocalCANRPdo4_EventTimer	0x5403#5	UINT16	rw	Y	UINT16	-
73#20…27	N	locpdrmap[0…7]	Moog DCV	LocalCAN, RPdo_Mapping LocalRPdo1_ApplicPara1	0x5600#1…8	UINT32	rw	Y	UINT32	-
73#28…35	N	locpdrmap[8…15]	Moog DCV	LocalCAN, RPdo_Mapping LocalRPdo2_ApplicPara1	0x5601#1…8	UINT32	rw	Y	UINT32	-
73#36…43	N	locpdrmap[16…23]	Moog DCV	LocalCAN, RPdo_Mapping LocalRPdo3_ApplicPara1	0x5602#1…8	UINT32	rw	Y	UINT32	-
73#44…51	N	locpdrmap[24…31]	Moog DCV	LocalCAN, RPdo_Mapping LocalRPdo4_ApplicPara1	0x5603#1…8	UINT32	rw	Y	UINT32	-
73#54	N	locpdtcob[0]	Moog DCV	LocalCAN, LocalCANTPdo_CobIdUsedByPdo LocalCANTPdo1_CobIdUsedByPdo	0x5800#1	UINT32	rw	Y	1…0x800007FF	<NodeID> (0x100B) + 0x0180
73#55	N	locpdtcob[1]	Moog DCV	LocalCAN, LocalCANTPdo_CobIdUsedByPdo LocalCANTPdo2_CobIdUsedByPdo	0x5801#1	UINT32	rw	Y	1…0x800007FF	<NodeID> (0x100B) + 0x0280
73#56	N	locpdtcob[2]	Moog DCV	LocalCAN, LocalCANTPdo_CobIdUsedByPdo LocalCANTPdo3_CobIdUsedByPdo	0x5802#1	UINT32	rw	Y	1…0x800007FF	<NodeID> (0x100B) + 0x0380
73#57	N	locpdtcob[3]	Moog DCV	LocalCAN, LocalCANTPdo_CobIdUsedByPdo LocalCANTPdo4_CobIdUsedByPdo	0x5803#1	UINT32	rw	Y	1…0x800007FF	<NodeID> (0x100B) + 0x0480
73#58	N	locpdttrn[0]	Moog DCV	LocalCAN, LocalCANTPdo_TransmissionType LocalCANTPdo1_TransmissionType	0x5800#2	UINT8	rw	Y	UINT8	255
73#59	N	locpdttrn[1]	Moog DCV	LocalCAN, LocalCANTPdo_TransmissionType LocalCANTPdo2_TransmissionType	0x5801#2	UINT8	rw	Y	UINT8	255
73#60	N	locpdttrn[2]	Moog DCV	LocalCAN, LocalCANTPdo_TransmissionType LocalCANTPdo3_TransmissionType	0x5802#2	UINT8	rw	Y	UINT8	255

Table 104: Object dictionary (part 32 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
73#61	N	locpdttrn[3]	Moog DCV	LocalCAN, LocalCANTPdo_TransmissionType LocalCANTPdo4_TransmissionType	0x5803#2	UINT8	rw	Y	UINT8	255
73#62	N	locpdttrnman[0]	Moog DCV	LocalCAN, LocalCANTPdoManufacturerTransmissionType LocalCANTPdo1_ManufacturerTransmissionType	0x5A08#1	UINT8	rw	Y	UINT8	-
73#63	N	locpdttrnman[1]	Moog DCV	LocalCAN, LocalCANTPdoManufacturerTransmissionType LocalCANTPdo2_ManufacturerTransmissionType	0x5A08#2	UINT8	rw	Y	UINT8	-
73#64	N	locpdttrnman[2]	Moog DCV	LocalCAN, LocalCANTPdoManufacturerTransmissionType LocalCANTPdo3_ManufacturerTransmissionType	0x5A08#3	UINT8	rw	Y	UINT8	-
73#65	N	locpdttrnman[3]	Moog DCV	LocalCAN, LocalCANTPdoManufacturerTransmissionType LocalCANTPdo4_ManufacturerTransmissionType	0x5A08#4	UINT8	rw	Y	UINT8	-
73#66	N	locpdtinh[0]	Moog DCV	LocalCAN, LocalCANTPdo_InhibitTime LocalCANTPdo1_InhibitTime	0x5800#3	UINT16	rw	Y	UINT16	-
73#67	N	locpdtinh[1]	Moog DCV	LocalCAN, LocalCANTPdo_InhibitTime LocalCANTPdo2_InhibitTime	0x5801#3	UINT16	rw	Y	UINT16	-
73#68	N	locpdtinh[2]	Moog DCV	LocalCAN, LocalCANTPdo_InhibitTime LocalCANTPdo3_InhibitTime	0x5802#3	UINT16	rw	Y	UINT16	-
73#69	N	locpdtinh[3]	Moog DCV	LocalCAN, LocalCANTPdo_InhibitTime LocalCANTPdo4_InhibitTime	0x5803#3	UINT16	rw	Y	UINT16	-
73#70	N	locpdttim[0]	Moog DCV	LocalCAN, LocalCANTPdo_EventTimer LocalCANTPdo1_EventTimer	0x5800#5	UINT16	rw	Y	UINT16	-
73#71	N	locpdttim[1]	Moog DCV	LocalCAN, LocalCANTPdo_EventTimer LocalCANTPdo2_EventTimer	0x5801#5	UINT16	rw	Y	UINT16	-
73#72	N	locpdttim[2]	Moog DCV	LocalCAN, LocalCANTPdo_EventTimer LocalCANTPdo3_EventTimer	0x5802#5	UINT16	rw	Y	UINT16	-
73#73	N	locpdttim[3]	Moog DCV	LocalCAN, LocalCANTPdo_EventTimer LocalCANTPdo4_EventTimer	0x5803#5	UINT16	rw	Y	UINT16	-
73#74…81	N	locpdtmap[0…7]	Moog DCV	LocalCAN, TPdo_Mapping LocalTPdo1_ApplicPara1	0x5A00#1…8	UINT32	rw	Y	UINT32	-
73#82…89	N	locpdtmap[8…15]	Moog DCV	LocalCAN, TPdo_Mapping LocalTPdo2_ApplicPara1	0x5A01#1…8	UINT32	rw	Y	UINT32	-
73#90…97	N	locpdtmap[16…23]	Moog DCV	LocalCAN, TPdo_Mapping LocalTPdo3_ApplicPara1	0x5A02#1…8	UINT32	rw	Y	UINT32	-
73#98…105	N	locpdtmap[24…31]	Moog DCV	LocalCAN, TPdo_Mapping LocalTPdo4_ApplicPara1	0x5A03#1…8	UINT32	rw	Y	UINT32	-
73#106	N	locpdtmapnum	Moog DCV	LocalCAN, TPdoMapping_NumberOfMappedApplicParaInPdo TPdoMapping_NumberOfMappedApplicParaInPdo	0x5A00#0	UINT8	rw	Y	0…8	0
73#107	N	locpdtmapnum	Moog DCV	LocalCAN, TPdoMapping_NumberOfMappedApplicParaInPdo TPdoMapping_NumberOfMappedApplicParaInPdo	0x5A01#0	UINT8	rw	Y	0…8	0

Table 104: Object dictionary (part 33 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
73#108	N	locpdtmapnum	Moog DCV	LocalCAN, TPdoMapping_NumberOfMappedApplicParaInPdo TPdoMapping_NumberOfMappedApplicParaInPdo	0x5A02#0	UINT8	rw	Y	0…8	0
73#109	N	locpdtmapnum	Moog DCV	LocalCAN, TPdoMapping_NumberOfMappedApplicParaInPdo TPdoMapping_NumberOfMappedApplicParaInPdo	0x5A03#0	UINT8	rw	Y	0…8	0
73#110	N	locpdttrg	Moog DCV	LocalCAN, LocalCAN_TPdoTrigger LocalCAN_TPdoTrigger	0x5B03#0	UINT8	rw	N	0…4	-
73#114	N	locbuster	Moog DCV	LocalCAN, LocalCAN_TerminationResistor LocalCAN_TerminationResistor	0x5B14#0	UINT8	rw	Y	0…1	-
73#115	N	locpdrmapnum	Moog DCV	LocalCAN, LocalCANRPdo_NumberOfMappedApplicParaInPdo LocalCANRPdo_NumberOfMappedApplicParaInPdo	0x5600#0	UINT8	rw	Y	0…8	0
73#116	N	locpdrmapnum	Moog DCV	LocalCAN, LocalCANRPdo_NumberOfMappedApplicParaInPdo LocalCANRPdo_NumberOfMappedApplicParaInPdo	0x5601#0	UINT8	rw	Y	0…8	0
73#117	N	locpdrmapnum	Moog DCV	LocalCAN, LocalCANRPdo_NumberOfMappedApplicParaInPdo LocalCANRPdo_NumberOfMappedApplicParaInPdo	0x5602#0	UINT8	rw	Y	0…8	0
73#118	N	locpdrmapnum	Moog DCV	LocalCAN, LocalCANRPdo_NumberOfMappedApplicParaInPdo LocalCANRPdo_NumberOfMappedApplicParaInPdo	0x5603#0	UINT8	rw	Y	0…8	0
74#1	Y	an0val	Moog DCV	AnalogInput0, AnaInpActualValue0 AnaInpActualValue0	0x3204#0	INT16	ro	-	INT16	-
74#2	N	an0typ	Moog DCV	AnalogInput0, AnaInpType0 AnaInpType0	0x3200#0	INT8	rw	Y	0…12	1
74#3	Y	an1val	Moog DCV	AnalogInput1, AnaInpActualValue1 AnaInpActualValue1	0x320C#0	INT16	ro	-	INT16	-
74#4	N	an1typ	Moog DCV	AnalogInput1, AnaInpType1 AnaInpType1	0x3208#0	INT8	rw	Y	0…12	2
74#5	N	da0ref[0]	Moog DCV	AnalogOutput0, AnaOutScaCustomer0 AnaOutScaNumerator0	0x3244#1	INT16	rw	Y	INT16	16384
74#6	N	da0ref[1]	Moog DCV	AnalogOutput0, AnaOutScaCustomer0 AnaOutScaDenominator0	0x3244#2	INT16	rw	Y	INT16	16384
74#7	N	da0ref[2]	Moog DCV	AnalogOutput0, AnaOutScaCustomer0 AnaOutScaOffset0	0x3244#3	INT16	rw	Y	INT16	0
74#8	N	da1ref[0]	Moog DCV	AnalogOutput1, AnaOutScaCustomer1 AnaOutScaNumerator1	0x3265#1	INT16	rw	Y	INT16	16384
74#9	N	da1ref[1]	Moog DCV	AnalogOutput1, AnaOutScaCustomer1 AnaOutScaDenominator1	0x3265#2	INT16	rw	Y	INT16	16384
74#10	N	da1ref[2]	Moog DCV	AnalogOutput1, AnaOutScaCustomer1 AnaOutScaOffset1	0x3265#3	INT16	rw	Y	INT16	0
74#11	Y	da0val	Moog DCV	AnalogOutput0, AnaOutValue0 AnaOutValue0	0x3245#0	INT16	ro	-	INT16	-

Table 104: Object dictionary (part 34 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
74#12	Y	da1val	Moog DCV	AnalogOutput1, AnaOutValue1 AnaOutValue1	0x3266#0	INT16	ro	-	INT16	-
74#14	Y	prstrd	Moog DCV	PressureTransducer, ActualValue ActualValue	0x3404#0	INT16	ro	-	INT16	-
74#15	N	da0par	Moog DCV	AnalogOutput0, AnaOutMappingParameter0 AnaOutMappingParameter0	0x3240#0	UINT32	rw	Y	UINT32	0x63010110
74#16	N	da1par	Moog DCV	AnalogOutput1, AnaOutMappingParameter1 AnaOutMappingParameter1	0x3260#0	UINT32	rw	Y	UINT32	0x63810110
74#17	N	da0typ	Moog DCV	AnalogOutput0, AnaOutType0 AnaOutType0	0x3243#0	UINT8	rw	Y	0…1	0
74#18	N	da1typ	Moog DCV	AnalogOutput1, AnaOutType0 AnaOutType0	0x3263#0	UINT8	rw	Y	0…1	0
74#19	N	lvdcusofs	Moog DCV	Lvdt, CustomerScalingOffset CustomerScalingOffset	0x3506#0	INT16	rw	Y	-819…819	-
75#1	Y	an2val	Moog DCV	AnalogInput2, AnaInpActualValue2 AnaInpActualValue2	0x3214#0	INT16	ro	-	INT16	-
75#2	N	an2typ	Moog DCV	AnalogInput2, AnaInpType2 AnaInpType2	0x3210#0	INT8	rw	Y	0…12	2
75#3	N	an2mon	Moog DCV	AnalogInput2, AnaInMonitorCurrent2 AnaInMonitorCurrent2	0x3217#0	UINT8	rw	Y	0…1	-
75#4	Y	an3val	Moog DCV	AnalogInput3, AnaInpActualValue3 AnaInpActualValue3	0x321C#0	INT16	ro	-	INT16	-
75#5	N	an3typ	Moog DCV	AnalogInput3, AnaInpType3 AnaInpType3	0x3218#0	INT8	rw	Y	0…12	2
75#6	N	an3mon	Moog DCV	AnalogInput3, AnaInMonitorCurrent3 AnaInMonitorCurrent3	0x3228#0	UINT8	rw	Y	0…1	-
75#7	Y	an4val	Moog DCV	AnalogInput4, AnaInpActualValue4 AnaInpActualValue4	0x3224#0	INT16	ro	-	INT16	-
75#8	N	an4typ	Moog DCV	AnalogInput4, AnaInpType4 AnaInpType4	0x3220#0	INT8	rw	Y	0…12	2
75#9	N	an4mon	Moog DCV	AnalogInput4, AnaInMonitorCurrent4 AnaInMonitorCurrent4	0x3227#0	UINT8	rw	Y	0…1	-
75#10	Y	extlvdval	Moog DCV	ExternalLVDT, ExternalLVDTActualValue ExternalLVDTActualValue	0x3235#0	INT16	ro	-	INT16	-
75#11	N	extlvdref[0]	Moog DCV	ExternalLVDT, ExternalLvdtScaCustomer ExternalLvdtScaNumerator	0x3237#1	INT16	rw	Y	INT16	16384
75#12	N	extlvdref[1]	Moog DCV	ExternalLVDT, ExternalLvdtScaCustomer ExternalLvdtScaDenominator	0x3237#2	INT16	rw	Y	INT16	16384

Table 104: Object dictionary (part 35 of 36)

Slot#Index	Cyclic data	Short name	Specification	Block name, Object name Parameter name	CANopen SDO	Data type	Access	Persistence	Value range	Default
75#13	N	extlvdref[2]	Moog DCV	ExternalLVDT, ExternalLvdtScaCustomer ExternalLvdtOffset	0x3237#3	INT16	rw	Y	INT16	0
75#14	N	anamonlow	Moog DCV	AnalogInput, LowerCurrentBorder LowerCurrentBorder	0x3250#0	FLOAT32	rw	Y	2.20…20.00	3.00
75#15	N	anamontim	Moog DCV	AnalogInput, anInputMonitoringTime_in_ms anInputMonitoringTime_in_ms	0x3251#0	UINT16	rw	Y	0…60000	10
75#16	N	iopbkpldver	Moog DCV	Software_PiggyBack, ManufactutrerIOPiggybackVersion ManufactutrerIOPiggybackVersion	0x200A#0	UINT16	ro	-	UINT16	0
75#17	N	ssierrcnt	Moog DCV	ErrorHandler, SSIErrorCount SSIErrorCount	0x3252#0	UINT8	rw	Y	0…254	-
75#18	N	spgposmin	Moog DCV	Hardware_DiagnosticData, SpringPositionMinimum SpringPositionMinimum	0x3307#0	INT16	ro	-	INT16	-
75#19	N	spgposmax	Moog DCV	Hardware_DiagnosticData, SpringPositionMaximum SpringPositionMaximum	0x3308#0	INT16	ro	-	INT16	-

Table 104: Object dictionary (part 36 of 36)

TAKE A CLOSER LOOK.

Moog designs a range of products that complement the performance of those featured in this user manual. Visit our Web site for more information and the Moog facility nearest you.

Australia
+61 3 9561 6044
info.australia@moog.com

Brazil
+55 11 3572 0400
info.brazil@moog.com

Canada
+1 716 652 2000
info.canada@moog.com

China
+86 21 2893 1600
info.china@moog.com

Finland
+358 10 422 1840
info.finland@moog.com

France
+33 1 4560 7000
info.france@moog.com

Germany
+49 7031 622 0
info.germany@moog.com

Hong Kong
+852 2 635 3200
info.hongkong@moog.com

India
+91 80 4057 6666
info.india@moog.com

Ireland
+353 21 451 9000
info.ireland@moog.com

Italy
+39 0332 421 111
info.italy@moog.com

Japan
+81 46 355 3767
info.japan@moog.com

Korea
+82 31 764 6711
info.korea@moog.com

Luxembourg
+352 40 46 401
info.luxembourg@moog.com

The Netherlands
+31 252 462 000
info.thenetherlands@moog.com

Norway
+47 6494 1948
info.norway@moog.com

Russia
+7 8 31 713 1811
info.russia@moog.com

Singapore
+65 677 36238
info.singapore@moog.com

South Africa
+27 12 653 6768
info.southafrica@moog.com

Spain
+34 902 133 240
info.spain@moog.com

Turkey
+90 216 663 6020
info.turkey@moog.com

United Kingdom
+44 (0) 1684 858000
info.uk@moog.com

USA
+1 716 652 2000
info.usa@moog.com

www.moog.com/industrial

Moog Literature Name
AXIS CONTROL VALVES WITH PROFIBUS INTERFACE - Firmware
Rev. A, October 2018, Id. CDS45936-en

WHAT MOVES YOUR WORLD

MOOG