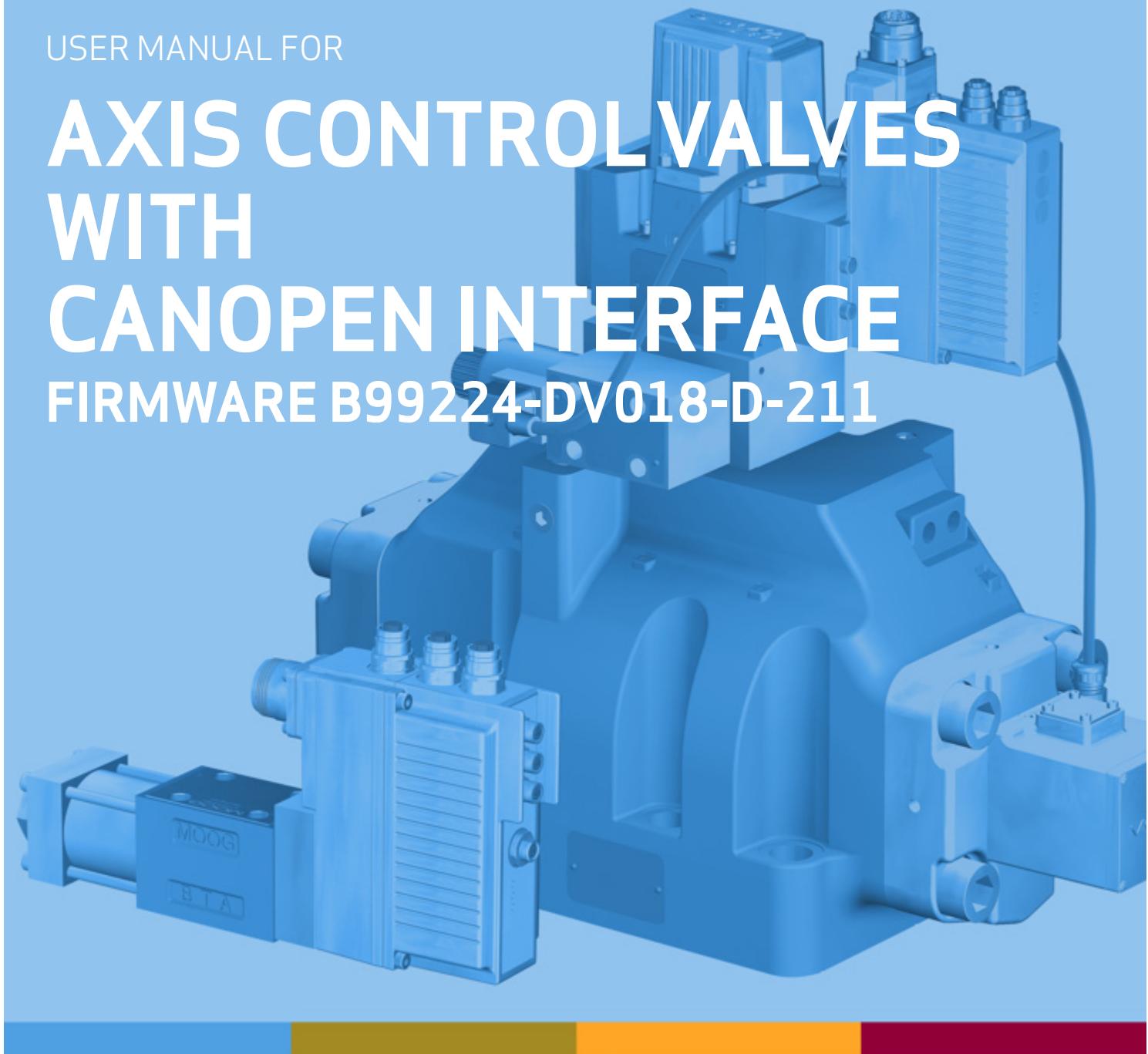USER MANUAL FOR

AXIS CONTROL VALVES WITH CANOPEN INTERFACE

FIRMWARE B99224-DV018-D-211

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

WHAT MOVES YOUR WORLD

MOOG

Copyright

© 2018 Moog GmbH
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 CANopen field bus interface of the Moog servo valves.

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

All parameters follow the common CANopen communication profile CiA 301 / CiA 301-1 / CiA 305 and the device specific CANopen profile "Device profile fluid power technology proportional valves and hydrostatic transmissions", CiA 408 released by the CAN in Automation (CiA) organisation.

This manual is part of the set of documentation available for the servo valve.

This document is not a replacement for the CANopen standards as listed in the references.

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 CANopen, 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

1.5.1 CAN field bus

The CANopen field bus interface provides a connection to the servo valves using standard CAN frames according to ISO 11898-1…3 and ISO 11898-5.

CiA 301	CANopen application layer and communication profile
CiA 303-1	Cabling and connector pin assignment
CiA 305	Layer setting services (LSS) and protocols
ISO 11898-1	Road vehicles -- Controller area network (CAN) -- Part 1: Data link layer and physical
ISO 11898-2	Road vehicles -- Controller area network (CAN) -- Part 2: High-speed medium access unit
ISO 11898-3	Road vehicles -- Controller area network (CAN) -- Part 3: Low-speed, fault-tolerant, medium-dependent interface
ISO 11898-5	Road vehicles -- Controller area network (CAN) -- Part 5: High-speed medium access unit with low-power mode

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

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

Single stage servo valve

Dual stage servo valve
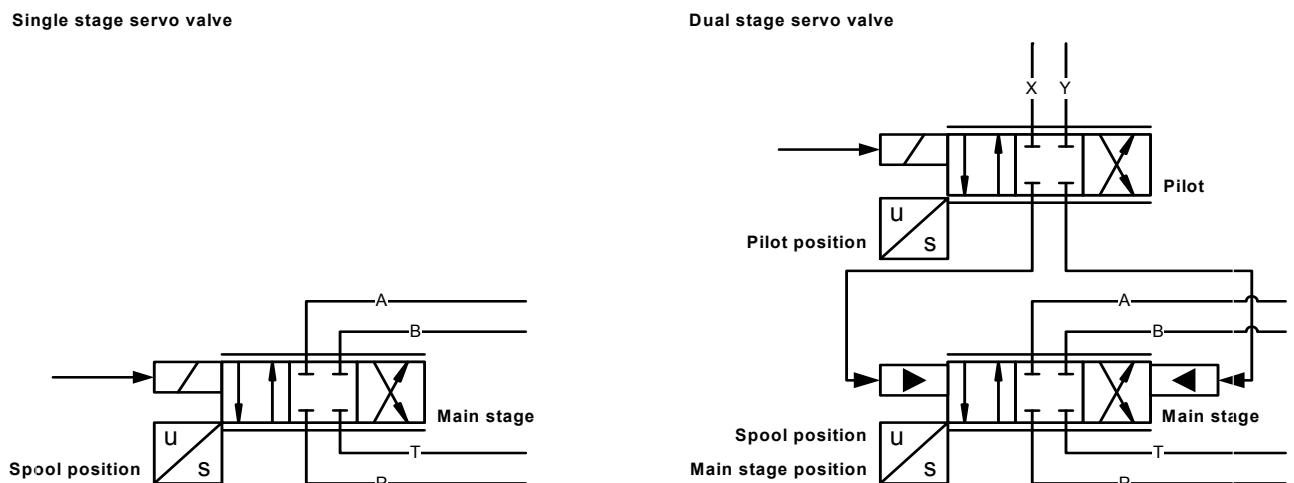
Figure 2: Servo valve position and stage names

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
PPT1	Proportional first order lag element
ro	Read only
rw	Read write
RxPDO	Receive Process Data Object

Table 1: Abbreviations (part 1 of 2)

Abbreviation	Explanation
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 CANopen

2.1 Introduction

CANopen is a communication protocol and device profile specification for embedded systems used in automation. In terms of the Open Systems Interconnection (OSI) model, CANopen implements the above layers and the network layer. The CANopen standard consists of an addressing scheme, several communication protocols and an application layer defined by specific device profiles. The communication protocols have support for network management, device monitoring and communication between nodes. The lower level protocol implementing the data link and physical layers is usually Controller Area Network (CAN). The basic CANopen application and communication profiles are given in the CiA 301 specification released by CAN in Automation (CiA). The device profiles for different devices are built on top of this basic profile.

2.2 Device profiles

The German Engineering Federation (VDMA), together with the manufacturers of hydraulic devices, have developed the "profile for fluid power technology". This profile defines common functionality and parameters for the communication of hydraulic components via field bus in a standardized format across manufacturers. This profile is implemented in all Moog servo valves with field bus interface. The CiA organization transformed the bus-independent device profile from the VDMA to the CANopen specific device profile CiA 408 "Device Profile Fluid Power Technology - proportional valves and hydrostatic transmissions". The device profiles describe the application parameters and the functional behavior of the devices including the device class-specific state machines. For many device classes, field bus technology already offers reliable device profiles for example for generic I/O-modules (CiA 401), drives and motion control (CiA 402) or for fluid power technology, proportional valves and hydrostatic transmissions (CiA 408). Users should be familiar with the associated profile.

2.3 CANopen slave reference model

The architecture of the CANopen stack with Physical Layer (Phy), Data Link Layer (DL) and Application Layer (AL) was taken from the ISO Reference Model (ISO/IEC standard 7498-1:1994). Layers three to six of this 7-layer reference model were not implemented, as these layers are intended for exchanging and sending telegrams. In a real-time field bus system, such functionalities are not required. The CANopen communication concept can be described similar to the ISO/OSI reference model.
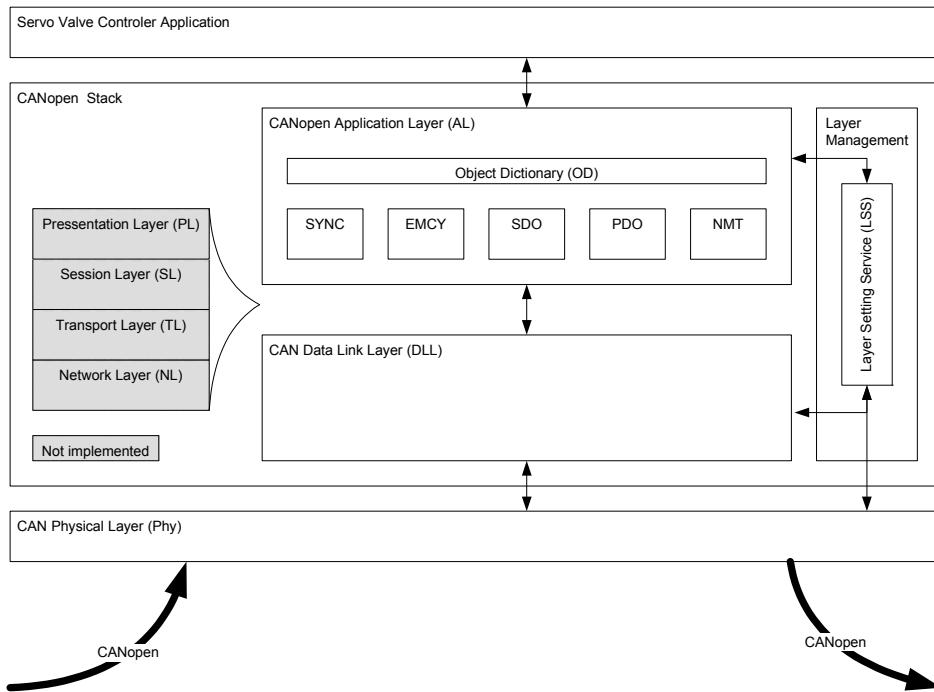
Figure 3: CANopen slave reference model

CANopen is based on the CAN data link layer and high-speed transceiver as specified in ISO 11898. CANopen specifies in CiA 303-1 the bit-timing and recommends connectors and their pin-assignments. CANopen represents a standardized application layer and communication profile as defined in CiA 301. The CiA 305 specifies the layer setting services (LSS). These protocols are used to inquire or to change the settings of the physical layer, data link layer and application layer on a device.

Layer	Description	References CAN / CANopen
Layer 7	Application layer	CiA 301 (CANopen application layer and communication profile) CiA 305 (CANopen LSS)
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	ISO 11898-1 (CAN) CiA 305 (CANopen LSS)
Layer 1	Physical layer	ISO 11898-1/2/3/5 (CAN) CiA 303-1 (CANopen Additional Specification)

Table 2: CANopen slave reference model

2.4 CANopen objects

A CANopen object is a set of CANopen parameters with the same index and object name. It consists of one or more parameters and their values. Objects are grouped in thematic blocks.

2.4.1 Parameter value

A parameter value is a real value stored in the servo valve with the attributes of the parameter explained in the next chapter.

2.4.2 Parameter and their attributes

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

Block name							
Index	Sub-index	Parameter name	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:

Index	Sub-index	PDO-mapping	Short name	Specification	Block object and parameter name	Data type	Access	Persis-tence	Value range	Default

Column name	Meaning
Block name	Describes the family of the object. If the object does not belong to a block, the object name is taken as block name.
Object name	Defined name of the object.
Index	16 bit index that addresses the entry in the object dictionary. In case of a simple variable this references the value of this variable directly. In case of records and arrays, the index addresses the whole data structure. Then the 8 bit sub-index allows access to individual elements in the structure.
Sub-index	If the object is defined as a record or array, the sub-index defines an element in the structure.
Parameter name	Defined name of the parameter.
Data type	Data type of the parameter. INTn — Integer with n bits FLOAT32 — Floating point with 32 bit char — Character (ASC II) STRING — String of characters UINTn — Unsigned integer with n bits DOMAIN — Application specific data block
Access	Access permission for the parameter. rw — Read and write allowed wo — Write only ro — Read only
Persistence	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.
Value range	Allowed value range for the object.
Default	**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 model specific during production of the servo valve. These parameters no longer contain the firmware default preset values. ⇨ Chapter "9 Storing / restoring parameters", page 295
Specification	Related (field bus) standard defining the parameter. Possible entries: CiA 301 — Parameters correspond to CiA 301 (CANopen). CiA 408 — Parameters correspond to CiA 408. Moog DCV — Moog defined parameters for digital control valves.
PDO mapping	If set to "Y", the parameter can be mapped into a PDO. If set to "N", the parameter cannot be mapped into a PDO.
Short name	Unique short name.

Table 3: Field bus independent attributes

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

2.4.3 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 4: 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 5: Prefix representation

2.5 CANopen object dictionary (OD)

CANopen devices have an object dictionary, which is used for configuration and non-realtime communication with the device. It is essentially a grouping of objects accessible via the network. Each object within the object dictionary is addressed using a 16 bit index and an 8 bit sub-index. So an object can contain 256 parameters which are addressed by the sub-index. The object dictionary is structured in several index ranges. The classification of the object dictionary is defined in the CiA 301.

Index	Object	Reference
0x0000	Not used	
0x0001…0x001F	Data types	CiA 301
0x0020…0x003F	Complex data types (not used)	CiA 301
0x0040…0x005F	Manufacturer-specific complex data types (not used)	
0x0060…0x025F	Device profile specific data types (not used)	CiA 408
0x0260…0x03FF	Reserved for further use	
0x0400…0x0FFF	Reserved for further use	
0x1000…0x1FFF	Communication profile area	CiA 301 / IEC 61158-5-12
0x2000…0x5FFF	Manufacturer-specific area	Moog DCV

Table 6: Structure of the CANopen object dictionary (OD)

Index	Object	Reference
0x6000…0x67FF	Standardized profile area 1st logical device	CiA 408
0x6800…0x9FFF	Standardized profile area 2nd…8th logical device (not used)	CiA 301
0xA000…0xAFFF	Standardized network variable area (not used)	
0xB000…0xBFFF	Standardized system variable area (not used)	
0xC000…0xFFFF	Reserved for further use	

Table 6: Structure of the CANopen object dictionary (OD)

2.6 CAN data link layer

The CAN data frame can carry up to 8 bytes data. The header has an 11 bit communication object identifier (COB-ID) used to address a CAN service. The lowest COB-ID has the highest transmission priority. 29-bit COB-IDs are not supported.
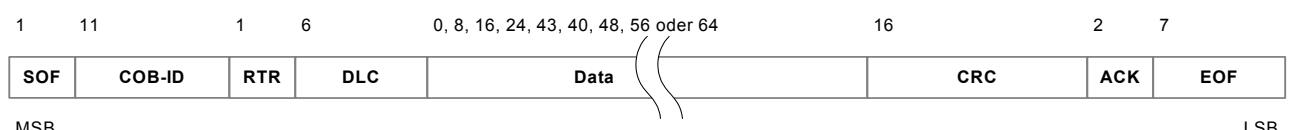
Figure 4: CAN protocol

SOF	Start of frame
COB-ID	Communication object identifier, 11-bit
RTR	Remote transmission request
DLC	Data length code
Data	CANopen protocols: SDO, PDO, NMT, EMCY …
CRC	Cyclic redundancy check
ACK	Acknowledgement
EOF	End of frame

2.7 CAN bit rate and Node-ID configuration

The bit rate of the CAN bus communication can be changed to achieve a maximal transmission rate by a given length of the cable. Each node in the network has a unique Node-ID which must be configured before initial operation.

The Node-ID and the bit rate can be changed using:

- Layer Setting Service (LSS)
- Service Data Objects (SDO)

The following standardized CANopen bit rates and maximum cable lengths can be configured:

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

Bit rates	Maximum cable length
1000 kBit/s	< 25 m
800 kBit/s	< 50 m
500 kBit/s	< 100 m
250 kBit/s	< 250 m
125 kBit/s	< 500 m
50 kBit/s	< 1000 m
20 kBit/s	< 2500 m

Table 7: CANopen bit rates

2.7.1 CAN bit rate and Node-ID configuration using LSS services

The digital servo valve has a Layer Setting Service (LSS) slave implementation according to CiA 305. With this service the bit rate and the Node-ID of the digital servo valve can be configured by a LSS master. The Layer Setting Services (LSS) use the COB-IDs 0x7E4 and 0x7E5.

The LSS service provides two methods to change the bit rate of the servo valve:

1. If a point to point connection has been established between the LSS master and the servo valve, the bit rate and the Node-ID is set in dialog mode.

2. If more than one slave nodes are connected to the network at the same time, the bit rate and Node-ID can be changed with the LSS service by selecting this particular slave device by his unique identification object (0x1018). This node identification object is worldwide unique and will be programmed during production. Information about the identification object address is available from the name plate of the device or can be inquired by using SDO or the boot-up message.

The LSS service itself stores the new settings permanently in the servo valve. It is not necessary to initiate a manual Store command as needed for other configuration parameters of the servo valve.

The identification object (Vendor-ID, Product Code, Revision Number, Serial Number) is printed on the name plate of the servo valve.

For more details about the LSS service see CiA 305.
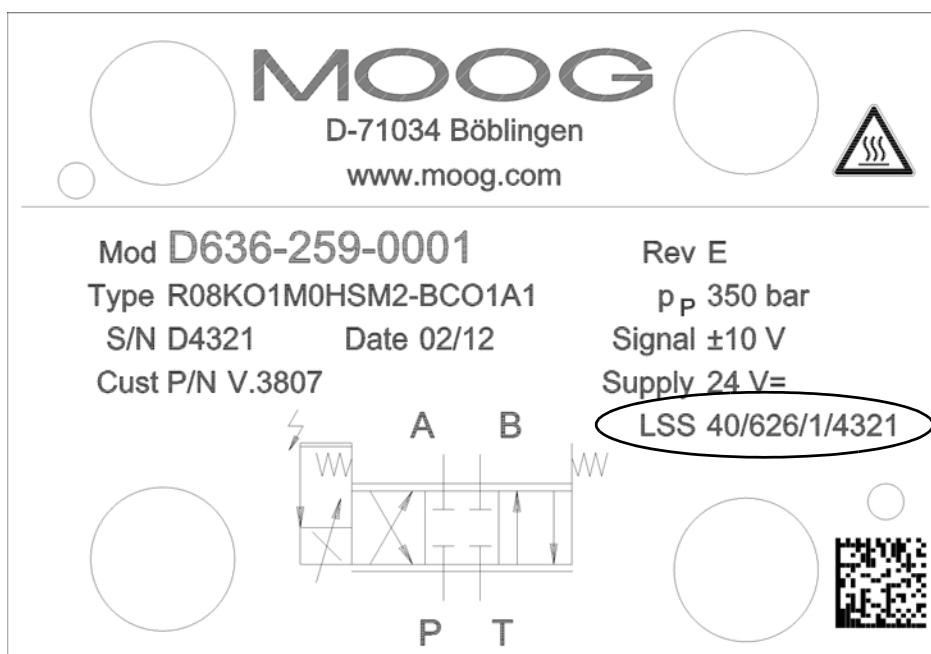
Figure 5: Name plate of the device with identification object address

2.7.2 CAN bit rate and Node-ID configuration using SDO protocol

In case a network master does not support the LSS services, the bit rate and the Node-ID in the servo valve can be configured by the SDO protocol. Therefore two SDO objects were introduced.

When changing these settings, they will get active not before either a power-up or a NMT-reset communication are initiated. That makes sure that the CAN communication will not get interrupted by changing these basic settings.

Different to all other configuration parameters, writing these parameters to the servo valve, stores them permanently. It is not necessary to initiate a manual Store command as needed for all other configuration parameters of the servo valve.

2.7.2.1 Object 0x3002: Module identifier (Node-ID)

This parameter represents the CAN Node-ID of the servo valve. To activate the new Node-ID either a power-up or a NMT-reset communication has to be initiated.

⇨ NMT protocol CiA 203
⇨ Chapter "2.9.6 Reset communication command (COB-ID:0, CS:130)", page 35

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3002	0	ModuleIdentifier	UINT8	rw	Y	1…127	127

2.7.2.2 Object 0x3003: Bit rate

The bit rate will be configured in bits per second. To activate the new bit rate either a power-up or a NMT-reset communication has to be initiated.

⇨ NMT protocol CiA 203
⇨ Chapter "2.9.6 Reset communication command (COB-ID:0, CS:130)", page 35

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3003	0	Bitrate	UINT32	rw	Y	1…1000000	500000

Value description

The following standardized CANopen bit rates can be configured:

Bit rate	Parameter value
1000 kBit/s	1000000
800 kBit/s	800000
500 kBit/s	500000
250 kBit/s	250000
125 kBit/s	125000
50 kBit/s	50000
20 kBit/s	20000

Table 8: CANopen bit rates

2.8 CANopen communication protocols

CANopen communication protocols are classified as follows:

- The real-time data are transferred with the process data object (PDO) protocol.
- The configuration parameters are transferred with the service data object (SDO) protocol.
- Special protocols provide application-specific network synchronization (SYNC) protocol, time stamping and emergency message (EMCY) protocol.
- The network management (NMT) protocol provides services for network initialization, error control and network status control.

Protocol	COB-ID	Description	SDO objects used for protocol configuration	Reference	Chapter
NMT	0x000	Network Management (NMT) protocol (Broadcast)		CiA 301	⇨ Chapter "2.9.1 Network management (NMT) protocol (COB-ID: 0x000, 0x700)", page 34
NMT	0x000+Node-ID	Network Management (NMT) protocol		CiA 301	
SYNC	0x080	Synchronization (Broadcast) protocol	0x1005	CiA 301	⇨ Chapter "2.8.1 Synchronization (SYNC) protocol (COB-ID: 0x080)", page 16
EMCY	0x080+Node-ID	Emergency protocol	0x1014	CiA 301	⇨ Chapter "2.8.2 Emergency (EMCY) protocol (COB-ID: 0x080+Node-ID)", page 17
TxPDO	0x180+Node-ID	1st Transmit PDO protocol	0x1800, 0x1A00	CiA 301	⇨ Chapter "2.8.3 Process data object (PDO) protocol", page 18
RxPDO	0x200+Node-ID	1st Receive PDO protocol	0x1400, 0x1600	CiA 301	
TxPDO	0x280+Node-ID	2nd Transmit PDO protocol	0x1801, 0x1A01	CiA 301	
RxPDO	0x300+Node-ID	2nd Receive PDO protocol	0x1401, 0x1601	CiA 301	
TxPDO	0x380+Node-ID	3rd Transmit PDO protocol	0x1802, 0x1A02	CiA 301	
RxPDO	0x400+Node-ID	3rd Receive PDO protocol	0x1402, 0x1602	CiA 301	
TxPDO	0x480+Node-ID	4th Transmit PDO protocol	0x1803, 0x1A03	CiA 301	
RxPDO	0x500+Node-ID	4th Receive PDO protocol	0x1403, 0x1603	CiA 301	
SDO	0x580+Node-ID	Transmit SDO protocol	0x1200	CiA 301	⇨ Chapter "2.8.4 Service data object (SDO) protocol (COB-ID: 0x580, 0x600)", page 31
SDO	0x600+Node-ID	Receive SDO protocol	0x1200	CiA 301	
NMT	0x700+Node-ID	Network Management (NMT) error control protocol (Bootup, Node Guarding)	0x100C, 0x100D, 0x1016, 0x1017	CiA 301	⇨ Chapter "2.9.1 Network management (NMT) protocol (COB-ID: 0x000, 0x700)", page 34
LSS	0x7E4	Transmit Layer Setting Services (LSS) protocol	0x1018	CiA 305	⇨ Chapter "2.7.1 CAN bit rate and Node-ID configuration using LSS services", page 14
LSS	0x7E5	Receive Layer Setting Services (LSS) protocol			

Table 9: CANopen communication objects

2.8.1 Synchronization (SYNC) protocol (COB-ID: 0x080)

The SYNC protocol is a network wide system trigger generated by one CANopen device in the network. The SYNC protocol has a very high priority and has no data in order to guarantee a minimum of jitter. The SYNC protocol is sent by a sync producer and can trigger PDO transmissions in the sync consumer nodes when activated in the corresponding PDO transmission types.

⇨ Chapter "2.8.3.1.1 Object 0x1400: 1st RxPDO protocol configuration", page 19
⇨ Chapter "2.8.3.4.1 Object 0x1800: 1st TxPDO protocol configuration", page 25
⇨ Chapter "2.8.3.6 Object 0x3011: TxPDO trigger", page 30

2.8.1.1 Object 0x1005: SYNC protocol COB-ID configuration

This object defines the COB-ID of the sync object itself.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1005	0	CobIdSyncMessage	UINT32	rw	Y	1…0x000007FF| 0x40000000	0x0080

Value description

<CobIdSyncMessage>				
Bit	**31**	**30**	**29…11**	**10…0**
Description	Reserved	0: SYNC client 1: SYNC server (not implemented)	Reserved	11 bit COB-ID

Table 10: Possible values of parameter <CobIdSyncMessage> (0x1005)

2.8.2 Emergency (EMCY) protocol (COB-ID: 0x080+Node-ID)

The emergency protocol is a high priority message triggered by an error event in the device. The error codes sent with the emergency message are device profile independent and are specified in the CANopen communication profile CiA 301. They are described in the chapter Diagnostics.

⇨ Chapter "8 Diagnostics", page 266

2.8.2.1 Object 0x1014: EMCY protocol COB-ID configuration

This object defines the COB-ID of the emergency object itself.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1014	0	CobIdEmergencyMessage	UINT32	rw	Y	1…0x000007FF	0x0080 +127

Value description

<CobIdEmergencyMessage>				
Bit	**31**	**30**	**29…11**	**10…0**
Description	0: EMCY exists	Reserved	Reserved	11 bit COB-ID

Table 11: Possible values of parameter <CobIdEmergencyMessage> (0x1014)

2.8.3 Process data object (PDO) protocol

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

Four TxPDOs and four RxPDOs are implemented:

- Receive process data object (RxPDO)
- Transmit process data object (TxPDO)

Figure 6: Process data object (PDO) mapping

2.8.3.1 RxPDO protocol configuration (COB-ID: 0x200, 0x300, 0x400, 0x500)

To enable receive process data object (RxPDO) transmission, the local application parameters must be mapped to the RxPDO and a transmission type must be selected for each RxPDO channel. Four RxPDO channels are available.

2.8.3.1.1 Object 0x1400: 1st RxPDO protocol configuration

First RxPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1400	1	RPdo1_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0200+127
0x1400	2	RPdo1_TransmissionType	UINT8	rw	Y	UINT8	255
0x1400	5	RPdo1_EventTimer	UINT16	rw	Y	UINT16	0

Value description

Parameter	Description
<RPdo1_CobIdUsedByPdo>	COB-ID of the 1st RxPDO.
<RPdo1_TransmissionType>	Different trigger can be configured: ⇨ Table 14, page 19 For local CAN this parameter must be set to 0xFF (asynchron).
<RPdo1_EventTimer>	This parameter defines the timeout in milliseconds for the RxPDO time-out monitoring. The event timer is used to recognize the expiration of the RxPDO. If the time elapsed and the PDO was not received within that period, a fault reaction can be initiated. If this parameter is set to 0, the receive timeout monitoring is turned off.

Table 12: Object 0x1400: 1st RxPDO configuration

<RPdo1_**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 13: Possible values of parameter <RPdo1_CobIdUsedByPdo> (0x1400, sub-index 1)

<RPdo1_**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 <1stRxPDO_TransmissionType> SYNC telegram.
241…253	Reserved
254…255	PDO will be processed immediately after reception.

Table 14: Possible values of parameter <RPdo1_TransmissionType> (0x1400, sub-index 2)

2.8.3.1.2 Object 0x1401: 2nd RxPDO protocol configuration

Second RxPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1401	1	RPdo2_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0300+127
0x1401	2	RPdo2_TransmissionType	UINT8	rw	Y	UINT8	255
0x1401	5	RPdo2_EventTimer	UINT16	rw	Y	UINT16	0

Value description
⇨ Table 12, page 19

2.8.3.1.3 Object 0x1402: 3rd RxPDO protocol configuration

Third RxPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis- tence	Value range	Default
0x1402	1	RPdo3_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0400+127
0x1402	2	RPdo3_TransmissionType	UINT8	rw	Y	UINT8	255
0x1402	5	RPdo3_EventTimer	UINT16	rw	Y	UINT16	0

Value description
⇨ Table 12, page 19

2.8.3.1.4 Object 0x1403: 4th RxPDO protocol configuration

Fourth RxPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis- tence	Value range	Default
0x1403	1	RPdo4_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0500+127
0x1403	2	RPdo4_TransmissionType	UINT8	rw	Y	UINT8	255
0x1403	5	RPdo4_EventTimer	UINT16	rw	Y	UINT16	0

Value description
⇨ Table 12, page 19

2.8.3.2 RxPDO mapping (COB-ID: 0x200, 0x300, 0x400, 0x500)

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> (0x6040) (sub-index 0x00)
  ⇨ Chapter "5.2 Device state machine (DSM)", page 46
- Spool position setpoint value <SplSetpoint> (0x6300) (sub-index 0x01)
  ⇨ Chapter "6.2.3 Spool position setpoint value path", page 59
- Pressure setpoint value <PrsSetpoint> (0x6380) (sub-index 0x01)
  ⇨ Chapter "6.2.4 Pressure setpoint value path", page 61
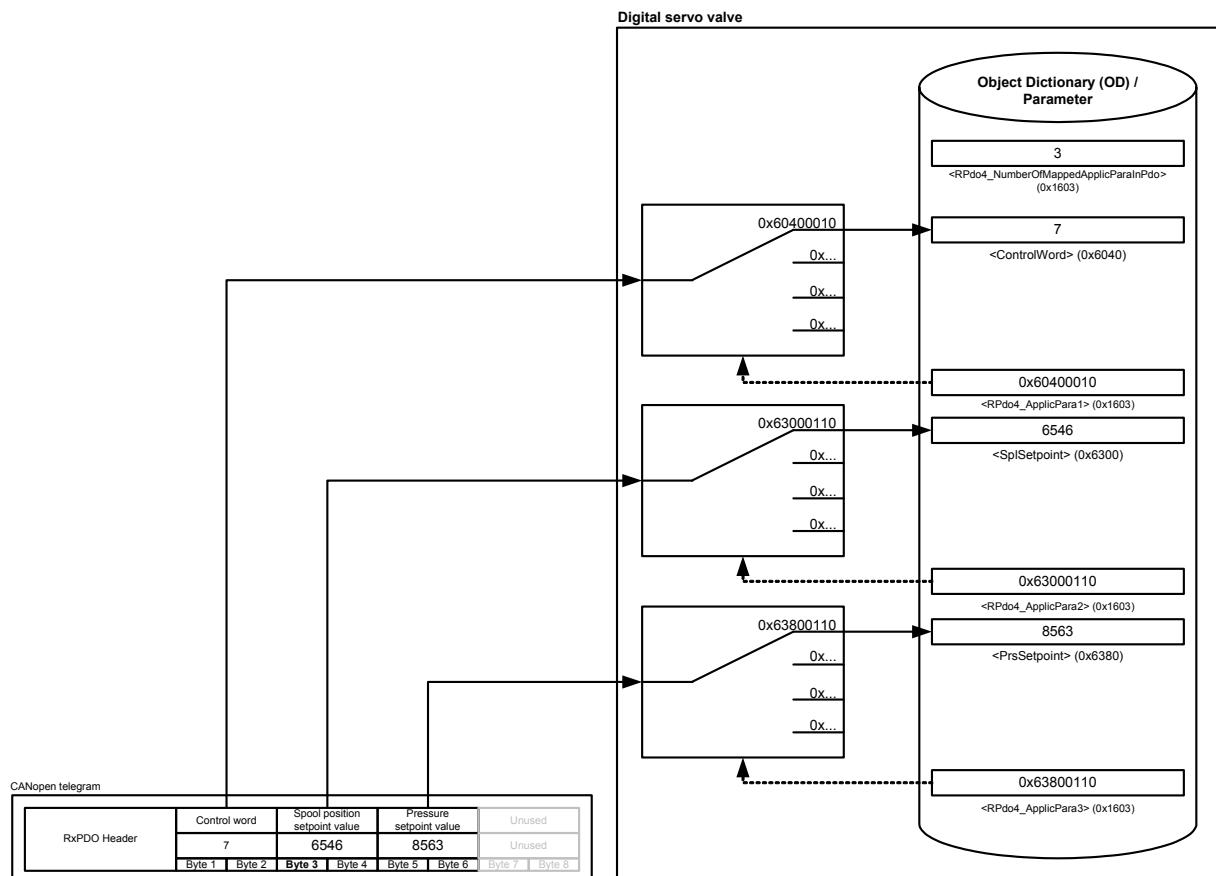
Figure 7: Receive process data object (RxPDO) mapping

The parameter <RPdo4_NumberOfMappedApplicParaInPdo> (0x1603) defines the number of mapped values for the fourth RxPDO. The second sub-index is a reference to the <ControlWord> (0x6040) (sub-index 0x00). The references to the parameters <SplSetpoint> (0x6300) (sub index 0x01) and <PrsSetpoint> (0x6380) (sub-index 0x01) are defined in the same manner.

2.8.3.2.1 Object 0x1600: 1st RxPDO mapping

With the parameter <RPdo1_NumberOfMappedApplicParaInPdo> (0x1600) the number of real-time application parameters to be received can be set. To map the application parameter itself, its index, sub-index and length must be combined to a 32 bit value and written to one of the eight possible positions (corresponding to the sub-indexes 1…8) within the PDO object mapping object.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1600	0	RPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	1
0x1600	1	RPdo1_ApplicPara1	UINT32	rw	Y	UINT32	0x60400010
0x1600	2	RPdo1_ApplicPara2	UINT32	rw	Y	UINT32	0
0x1600	3	RPdo1_ApplicPara3	UINT32	rw	Y	UINT32	0
0x1600	4	RPdo1_ApplicPara4	UINT32	rw	Y	UINT32	0
0x1600	5	RPdo1_ApplicPara5	UINT32	rw	Y	UINT32	0
0x1600	6	RPdo1_ApplicPara6	UINT32	rw	Y	UINT32	0
0x1600	7	RPdo1_ApplicPara7	UINT32	rw	Y	UINT32	0
0x1600	8	RPdo1_ApplicPara8	UINT32	rw	Y	UINT32	0

Value description

Parameter	Description
<RPdo1_NumberOfMappedApplicParaInPdo>	Numbers of mapped application parameters
<RPdo1_ApplicPara1>	Mapping of 1st application parameter
<RPdo1_ApplicPara2>	Mapping of 2nd application parameter
<RPdo1_ApplicPara3>	Mapping of 3rd application parameter
<RPdo1_ApplicPara4>	Mapping of 4th application parameter
<RPdo1_ApplicPara5>	Mapping of 5th application parameter
<RPdo1_ApplicPara6>	Mapping of 6th application parameter
<RPdo1_ApplicPara7>	Mapping of 7th application parameter
<RPdo1_ApplicPara8>	Mapping of 8th application parameter

Table 15: Object 0x1600: 1st RxPDO mapping

<RPdo1_ApplicPara1>…<RPdo1_ApplicPara8>			
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length 0x08 or 0x10 or 0x20
Example	0x60	0x40	0x00	0x10

Table 16: Value description of mapping parameter <RPdo1_ApplicPara1>…<RPdo1_ApplicPara8>

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

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

2.8.3.2.2 Object 0x1601: 2nd RxPDO mapping

Second RxPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1601	0	RPdo2_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	2
0x1601	1	RPdo2_ApplicPara1	UINT32	rw	Y	UINT32	0x60400010
0x1601	2	RPdo2_ApplicPara2	UINT32	rw	Y	UINT32	0x63000110
0x1601	3	RPdo2_ApplicPara3	UINT32	rw	Y	UINT32	0
0x1601	4	RPdo2_ApplicPara4	UINT32	rw	Y	UINT32	0
0x1601	5	RPdo2_ApplicPara5	UINT32	rw	Y	UINT32	0
0x1601	6	RPdo2_ApplicPara6	UINT32	rw	Y	UINT32	0
0x1601	7	RPdo2_ApplicPara7	UINT32	rw	Y	UINT32	0
0x1601	8	RPdo2_ApplicPara8	UINT32	rw	Y	UINT32	0

The mapping of this PDO is predefined as default mapping for a standard Q-control valve. It maps the <ControlWord> (0x6040) and the <SplSetpoint> (0x6300).

Value description
⇨ Table 15, page 22

2.8.3.2.3 Object 0x1602: 3rd RxPDO mapping

Third RxPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1602	0	RPdo3_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	2
0x1602	1	RPdo3_ApplicPara1	UINT32	rw	Y	UINT32	0x60400010
0x1602	2	RPdo3_ApplicPara2	UINT32	rw	Y	UINT32	0x63800110
0x1602	3	RPdo3_ApplicPara2	UINT32	rw	Y	UINT32	0
0x1602	4	RPdo3_ApplicPara4	UINT32	rw	Y	UINT32	0
0x1602	5	RPdo3_ApplicPara5	UINT32	rw	Y	UINT32	0
0x1602	6	RPdo3_ApplicPara6	UINT32	rw	Y	UINT32	0
0x1602	7	RPdo3_ApplicPara7	UINT32	rw	Y	UINT32	0
0x1602	8	RPdo3_ApplicPara8	UINT32	rw	Y	UINT32	0

The mapping of this PDO is predefined as default mapping for a standard p-control valve. It maps the <ControlWord> (0x6040) and the <PrsSetpoint> (0x6380).

Value description
⇨ Table 15, page 22

2.8.3.2.4 Object 0x1603: 4th RxPDO mapping

Fourth RxPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1603	0	RPdo4_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	3
0x1603	1	RPdo4_ApplicPara1	UINT32	rw	Y	UINT32	0x60400010
0x1603	2	RPdo4_ApplicPara2	UINT32	rw	Y	UINT32	0x63000110
0x1603	3	RPdo4_ApplicPara3	UINT32	rw	Y	UINT32	0x63800110
0x1603	4	RPdo4_ApplicPara4	UINT32	rw	Y	UINT32	0
0x1603	5	RPdo4_ApplicPara5	UINT32	rw	Y	UINT32	0
0x1603	6	RPdo4_ApplicPara6	UINT32	rw	Y	UINT32	0
0x1603	7	RPdo4_ApplicPara7	UINT32	rw	Y	UINT32	0
0x1603	8	RPdo4_ApplicPara8	UINT32	rw	Y	UINT32	0

The mapping of this PDO is predefined as default mapping for a standard p/Q-control valve. It maps the <ControlWord> (0x6040), the <SplSetpoint> (0x6300) and the <PrsSetpoint> (0x6380).

Value description

2.8.3.3 RxPDO counter

The received process data objects (RxPDOs) are counted using the object 0x3012 separately for every four PDO channels. On each power on the counters will be initialized with zero.

2.8.3.3.1 Object 0x3012: RxPDO counter

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3012	1	RPdo1_Counter	UINT32	rw	N	UINT32	0
0x3012	2	RPdo2_Counter	UINT32	rw	N	UINT32	0
0x3012	3	RPdo3_Counter	UINT32	rw	N	UINT32	0
0x3012	4	RPdo4_Counter	UINT32	rw	N	UINT32	0

2.8.3.4 TxPDO protocol configuration (COB-ID: 0x180, 0x280, 0x380, 0x480)

The transmit process data object (TxPDO) protocol must be configured and the transmit values must be mapped to the local parameters. Four transmit TxPDO channels are available. The transmission of the PDOs can be triggered by following events:

- Event timer elapsed.
- Synchronization (SYNC) telegram received.

Every event forces a PDO transmission. More than one event type can be active at the same time.

2.8.3.4.1 Object 0x1800: 1st TxPDO protocol configuration

First TxPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1800	1	TPdo1_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF|0x80000000	0x0180+127
0x1800	2	TPdo1_TransmissionType	UINT8	rw	Y	UINT8	255
0x1800	3	TPdo1_InhibitTime	UINT16	rw	Y	UINT16	0
0x1800	5	TPdo1_EventTimer	UINT16	rw	Y	UINT16	0

Value description

Parameter	Description
<TPdo1_CobIdUsedByPdo>	Defines the COB-ID for the 1st TxPDO.
<TPdo1_TransmissionType>	Defines the transmission behavior for the 1st TxPDO.
<TPdo1_InhibitTime>	Not implemented.
<TPdo1_EventTimer>	This parameter defines the event time in milliseconds. This time defines the cycle time of the TxPDO timer event. If the event time elapsed, a trigger to transmit a TxPDO is initiated and the event timer will be restarted.

Table 17: Object 0x1800: 1st TxPDO configuration

<TPdo1_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 18: Possible values of parameter <TPdo1_CobIdUsedByPdo> (0x1800 sub-index 1)

<TPdo1_TransmissionType>	
0	The process data for the TxPDO will be updated and send immediately after the next incoming SYNC telegram.
1…240	The process data for the TxPDO will be updated and send immediately every <TPdo1_TransmissionType> SYNC telegrams.
241…253	Reserved
254…255	The TxPDO will be send after the event time is elapses (when the event time is nonzero).

Table 19: Possible values of parameter <TPdo1_TransmissionType> (0x1400, sub-index 2)

ⓘ To initiate a single request only, the <TPdo1_EventTimer> (0x1800)…<TPdo4_EventTimer> (0x1803) must be set to 0. The transmission then can be triggered by the parameter <TPdoTrigger> (0x3011).
⇨ Chapter "2.8.3.6 Object 0x3011: TxPDO trigger", page 30

2.8.3.4.2 Object 0x1801: 2nd TxPDO configuration protocol

Second TxPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1801	1	TPdo2_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0280+127
0x1801	2	TPdo2_TransmissionType	UINT8	rw	Y	UINT8	255
0x1801	3	TPdo2_InhibitTime	UINT16	rw	Y	UINT16	0
0x1801	5	TPdo2_EventTimer	UINT16	rw	Y	UINT16	0

Value description

2.8.3.4.3 Object 0x1802: 3rd TxPDO protocol configuration

Third TxPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1802	1	TPdo3_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0380+127
0x1802	2	TPdo3_TransmissionType	UINT8	rw	Y	UINT8	255
0x1802	3	TPdo3_InhibitTime	UINT16	rw	Y	UINT16	0
0x1802	5	TPdo3_EventTimer	UINT16	rw	Y	UINT16	0

Value description

2.8.3.4.4 Object 0x1803: 4th TxPDO protocol configuration

Fourth TxPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1803	1	TPdo4_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0480+127
0x1803	2	TPdo4_TransmissionType	UINT8	rw	Y	UINT8	255
0x1803	3	TPdo4_InhibitTime	UINT16	rw	Y	UINT16	0
0x1803	5	TPdo4_EventTimer	UINT16	rw	Y	UINT16	0

Value description

2.8.3.5 TxPDO mapping (COB-ID: 0x180, 0x280, 0x380, 0x480)

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 Tx-PDO. An arbitrary combination of different data types is possible, if the sum of the mapped TxPDO data is less or equal 8 bytes.

The default PDO mapping for a hydraulic CANopen device is defined in the device specific profile CiA 408.

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

- Device state machine (DSM) Status Word <StatusWord> (0x6041) (sub-index 0x00)
  ⇨ Chapter "5.2 Device state machine (DSM)", page 46
- Spool position actual value <SplActualValue> (0x6301) (sub-index 0x01)
  ⇨ Chapter "7.3 Spool position controller", page 153
- Pressure actual value <PrsActualValue> (0x6381) (sub-index 0x01)
  ⇨ Chapter "7.5.11 Actual value filter", page 174
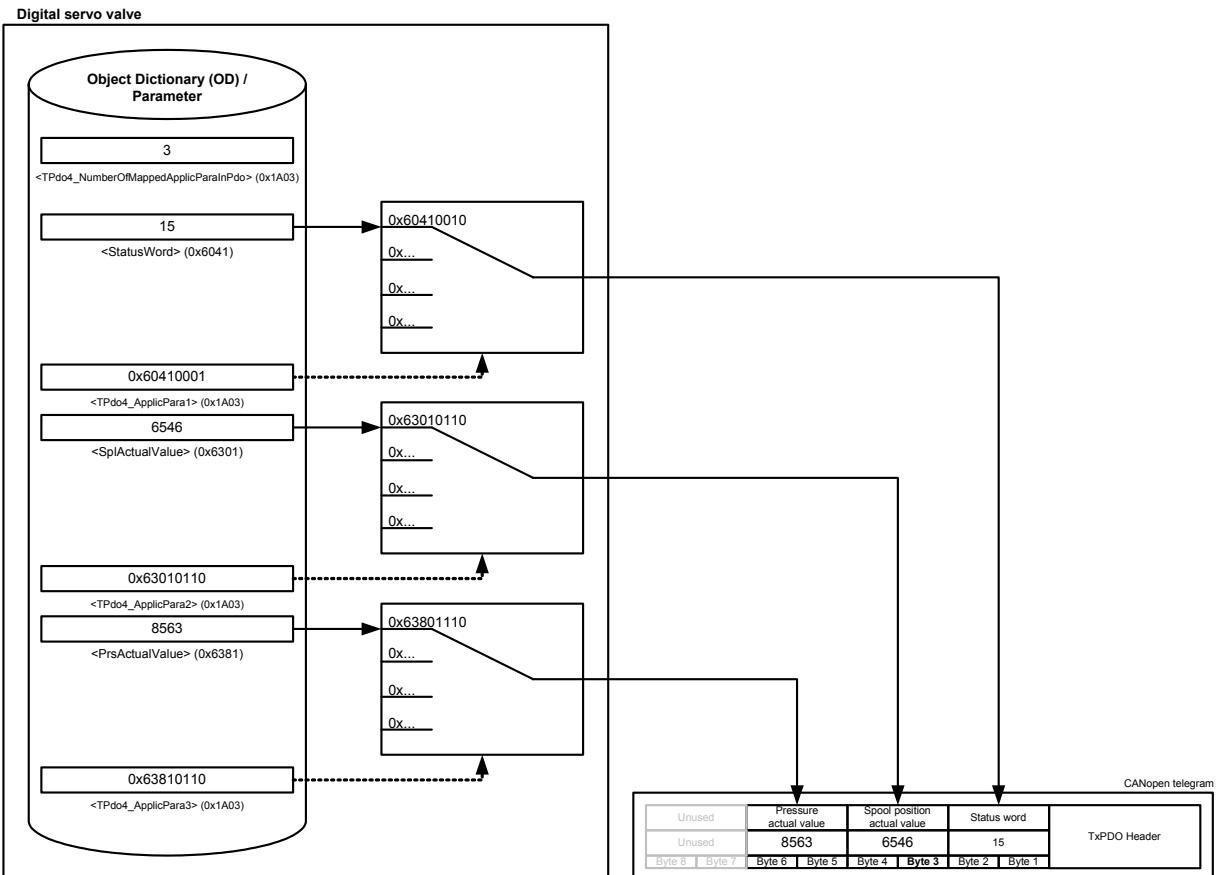
Figure 8: Transmit process data object (TxPDO) mapping

With the parameter <TPdo4_NumberOfMappedApplicParaInPdo> of the mapping object 0x1A00 the number of mapped values is defined. The second sub-index contains the reference to the <StatusWord> (0x6041) (sub-index 0x00) with a combination of index, sub-index and length of the parameter to be used. The references to the values <SplActualValue> (0x6301) (sub-index 0x01) and <PrsActualValue> (0x6381) (sub-index 0x01) are done in the same manner.

2.8.3.5.1 Object 0x1A00: 1st TxPDO mapping

With the parameter <TPdo_NumberOfMappedApplicParaInPdo> (0x1A00) the number of real-time application parameters to be transmitted can be set. To map the application parameter itself, its index, sub-index and length must be combined to a 32 bit value and written to one of the eight possible positions (corresponding to the subindexes 1…8) within the PDO object.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1A00	0	TPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	1
0x1A00	1	TPdo1_ApplicPara1	UINT32	rw	Y	UINT32	0x60410010
0x1A00	2	TPdo1_ApplicPara2	UINT32	rw	Y	UINT32	0
0x1A00	3	TPdo1_ApplicPara3	UINT32	rw	Y	UINT32	0
0x1A00	4	TPdo1_ApplicPara4	UINT32	rw	Y	UINT32	0
0x1A00	5	TPdo1_ApplicPara5	UINT32	rw	Y	UINT32	0
0x1A00	6	TPdo1_ApplicPara6	UINT32	rw	Y	UINT32	0
0x1A00	7	TPdo1_ApplicPara7	UINT32	rw	Y	UINT32	0
0x1A00	8	TPdo1_ApplicPara8	UINT32	rw	Y	UINT32	0

Value description

Parameter	Description
<TPdo1_NumberOfMappedApplicParaInPdo>	Numbers of mapped application parameters
<TPdo1_ApplicPara1>	1st application parameter
<TPdo1_ApplicPara2>	2nd application parameter
<TPdo1_ApplicPara3>	3rd application parameter
<TPdo1_ApplicPara4>	4th application parameter
<TPdo1_ApplicPara5>	5th application parameter
<TPdo1_ApplicPara6>	6th application parameter
<TPdo1_ApplicPara7>	7th application parameter
<TPdo1_ApplicPara8>	8th application parameter

Table 20: Object 0x1600: 1st TxPDO mapping

<TPdo1_ApplicPara1>…<TPdo1_ApplicPara8>			
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length 0x08 or 0x10 or 0x20
Example	0x60	0x41	0x00	0x10

Table 21: Possible values of parameter <TPdo1_ApplicPara1>…<TPdo1_ApplicPara8>

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> (0x6041), sub-index 0x00 with a length of 16 bit (16=0x10).

2.8.3.5.2 Object 0x1A01: 2nd TxPDO mapping

Second TxPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1A01	0	TPdo2_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	2
0x1A01	1	TPdo2_ApplicPara1	UINT32	rw	Y	UINT32	0x60410010
0x1A01	2	TPdo2_ApplicPara2	UINT32	rw	Y	UINT32	0x63010110
0x1A01	3	TPdo2_ApplicPara3	UINT32	rw	Y	UINT32	0
0x1A01	4	TPdo2_ApplicPara4	UINT32	rw	Y	UINT32	0
0x1A01	5	TPdo2_ApplicPara5	UINT32	rw	Y	UINT32	0
0x1A01	6	TPdo2_ApplicPara6	UINT32	rw	Y	UINT32	0
0x1A01	7	TPdo2_ApplicPara7	UINT32	rw	Y	UINT32	0
0x1A01	8	TPdo2_ApplicPara8	UINT32	rw	Y	UINT32	0

The mapping of this PDO is predefined as default mapping for a standard Q-control valve. It maps the <StatusWord> (0x6041) and the <SplActualValue> (0x6301).

Value description
⇨ Table 20, page 28

2.8.3.5.3 Object 0x1A02: 3rd TxPDO mapping

Third TxPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1A02	0	TPdo3_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	2
0x1A02	1	TPdo3_ApplicPara1	UINT32	rw	Y	UINT32	0x60410010
0x1A02	2	TPdo3_ApplicPara2	UINT32	rw	Y	UINT32	0x63810110
0x1A02	3	TPdo3_ApplicPara3	UINT32	rw	Y	UINT32	0
0x1A02	4	TPdo3_ApplicPara4	UINT32	rw	Y	UINT32	0
0x1A02	5	TPdo3_ApplicPara5	UINT32	rw	Y	UINT32	0
0x1A02	6	TPdo3_ApplicPara6	UINT32	rw	Y	UINT32	0
0x1A02	7	TPdo3_ApplicPara7	UINT32	rw	Y	UINT32	0
0x1A02	8	TPdo3_ApplicPara8	UINT32	rw	Y	UINT32	0

The mapping of this PDO is predefined as default mapping for a standard p-control valve. It maps the <StatusWord> (0x6041) and the <PrsActualValue> (0x6381).

Value description
⇨ Table 20, page 28

2.8.3.5.4 Object 0x1A03: 4th TxPDO mapping

Fourth TxPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1A03	0	TPdo4_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	2
0x1A03	1	TPdo4_ApplicPara1	UINT32	rw	Y	UINT32	0x60410010
0x1A03	2	TPdo4_ApplicPara2	UINT32	rw	Y	UINT32	0x63010110
0x1A03	3	TPdo4_ApplicPara3	UINT32	rw	Y	UINT32	0x63810110
0x1A03	4	TPdo4_ApplicPara4	UINT32	rw	Y	UINT32	0
0x1A03	5	TPdo4_ApplicPara5	UINT32	rw	Y	UINT32	0
0x1A03	6	TPdo4_ApplicPara6	UINT32	rw	Y	UINT32	0
0x1A03	7	TPdo4_ApplicPara7	UINT32	rw	Y	UINT32	0
0x1A03	8	TPdo4_ApplicPara8	UINT32	rw	Y	UINT32	0

The mapping of this PDO is predefined as default mapping for a standard p/Q-control valve. It maps the <StatusWord> (0x6041), the <SplActualValue> (0x6301) and the <PrsActualValue> (0x6381).

Value description
⇨ Table 20, page 28

2.8.3.6 Object 0x3011: TxPDO trigger

Writing this parameter triggers a single TxPDO. This can be used to transmit a PDO on request only. To trigger one of the four TxPDOs, the parameter <TPdoTrigger> (0x3011) value must be set to the number of the PDO channel to be sent.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3011	0	TPdoTrigger	UINT8	rw	N	1…4	None

2.8.4 Service data object (SDO) protocol (COB-ID: 0x580, 0x600)

Service data objects (SDOs) enable access to all entries of a CANopen object dictionary. The service data objects are used to configure the settings for the communication and to set or read application parameters. They are transmitted non real-time with low priority.

Example for a typical write service data object request from client (PLC) to server (valve) - Protocol SDO download initiate:
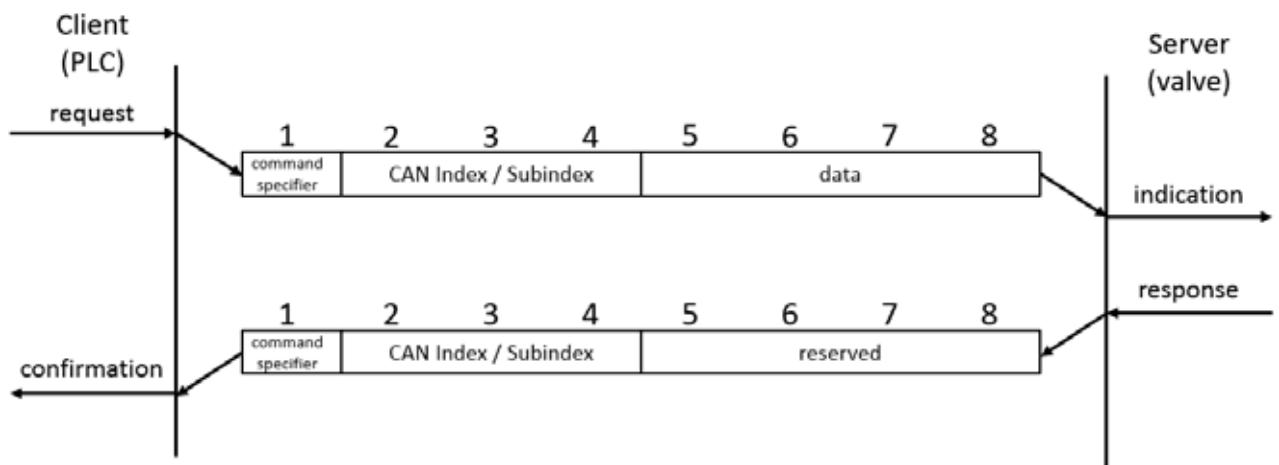
Figure 9: Write service data object request

Example for a typical read service data object request from client (PLC) to server (valve) - Protocol SDO upload initiate:
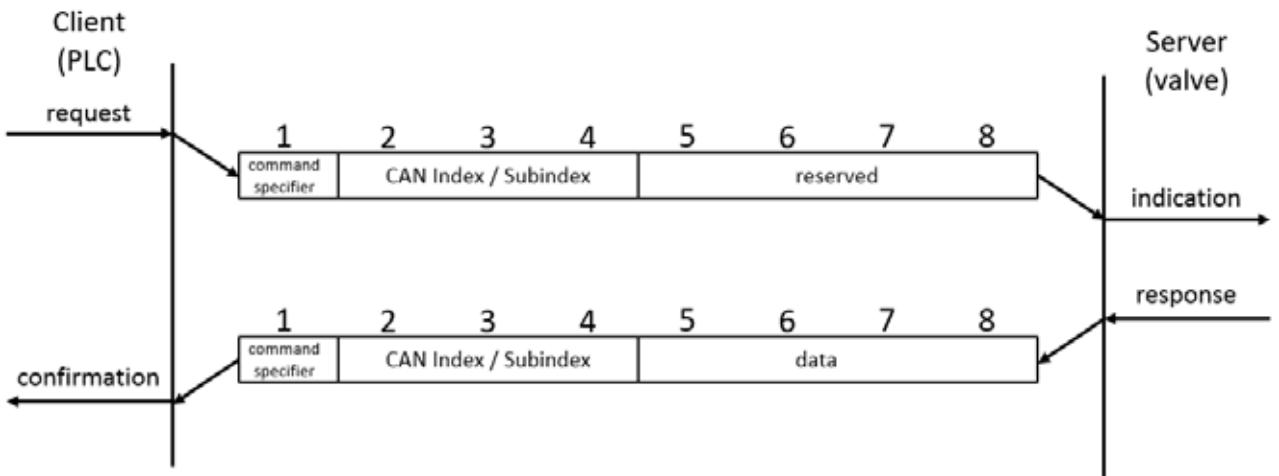
Figure 10: Read service data object request

If an SDO request cannot be processed, e.g., the value is too big or the client has not the access rights to write a parameter, an SDO abort message is sent back from the server to the client.
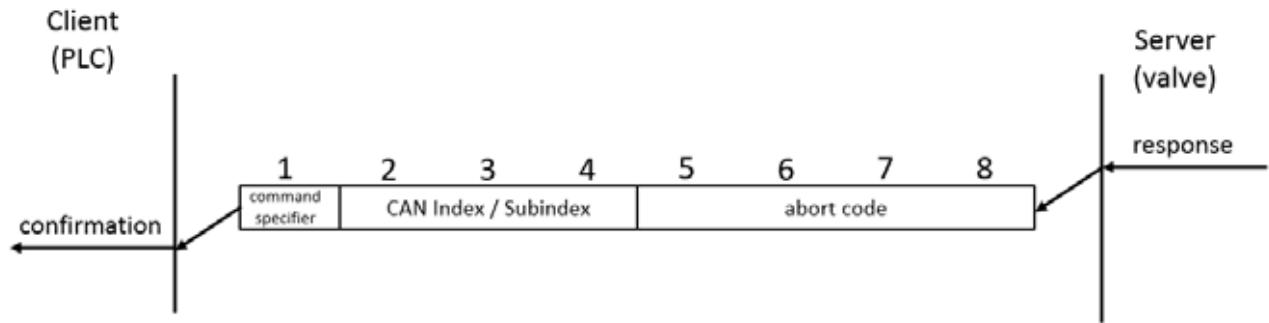
Figure 11: Typical SDO abort message

Description of the different SDO abort codes:
⇨ Chapter "8.3 Abort SDO Transfer Protocol", page 279

For detailed informations how the SDO protocol is working, please take a look into the actual CiA 301 CAN open application layer and communication profile.

2.8.4.1 Object 0x1200: SDO client/server parameter

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1200	1	CobIdClientServer	UINT32	ro	-		0x0600+127
0x1200	2	CobIdServerClient	UINT32	ro	-	UINT8	0x0580+127

Value description

Parameter	Description
<CobIdClientServer>	This parameter contains the COB-ID for the receive service data object.
<CobIdServerClient>	This parameter contains the COB-ID for the transmit service data object.

Table 22: Object 0x1200: SDO client/server parameter

2.9 Network management state machine (NMT state machine)

The CANopen NMT state machine is used to control the network communication. The network management is node-oriented and follows a master/slave structure. It requires one device in the network, which fulfills the function of the NMT master, the other nodes are NMT slaves. The network management provides the following functions:

- Device control services for initialization and start-up of NMT slave nodes that want to take part in the distributed application.
- Node Guarding of NMT slave nodes.

ⓘ The CANopen network management (NMT) state machine must not be mistaken with the device state machine.
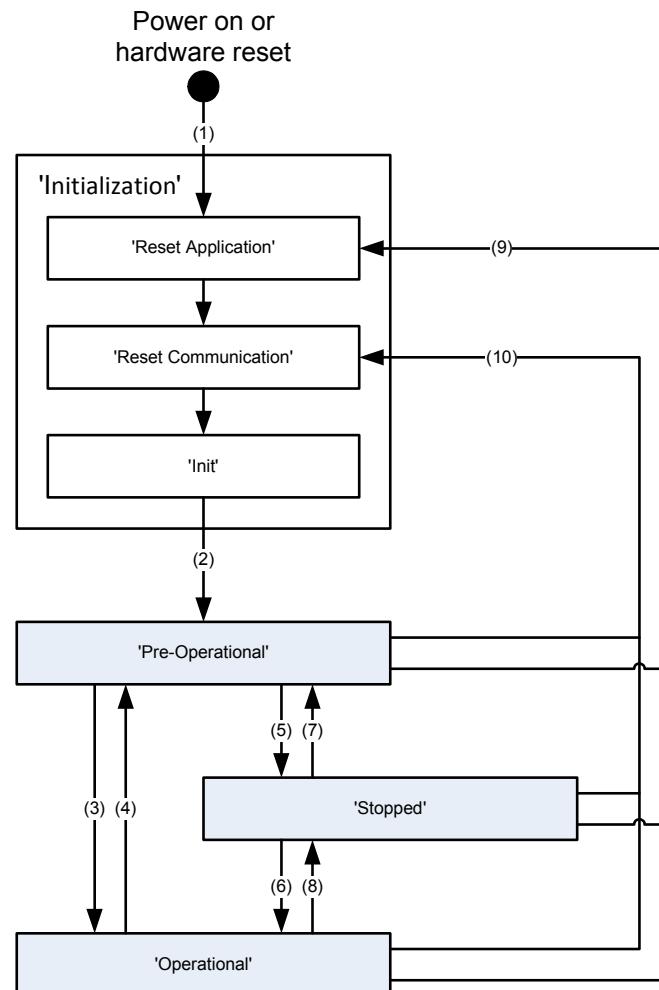
Figure 12: CANopen network state machine (CSM)

The master controls the state transitions of each device in the network. After power-on, the device is initialized and set to the state 'Pre-Operational' automatically. In this state reading and writing to its object dictionary via the service data objects (SDO) is possible. The device can be configured by setting values and by preparing the PDO transmission.

Afterwards the device can be switched into the 'Operational' state by the NMT command "Start Remote Node" in order to start PDO communication. PDO communication can be stopped by the network master by simply switching the remote node back to 'Pre-Operational' by using the command "Enter Pre-Operational".

With the "Stop Remote Node" command the master can force the slave(s) to the state 'Stopped'. In this state no services besides network and error control mechanism are available. The NMT command "Reset Communication" resets the communication of the node. All communication parameters will be set to their defaults. The application will be reset by the NMT command "Reset Node". This command resets all application parameter. All NMT commands use the COB-ID 0. The different NMT commands are distinguished by a command specifier (CS) located in the first data byte of the message.

NMT state transitions are caused by

- Reception of an NMT node control service
- Hardware reset, or
- Node control services locally initiated by application events.

Transition	Description
(1)	At Power on the NMT state initialization is entered autonomously.
(2)	NMT state initialization finished - enter NMT state Pre-operational automatically (Boot-up message will be sent).
(3)	Receive NMT command "start remote node" or by local control.
(4), (7)	Receive NMT command "enter pre-operational".
(5), (8)	Receive NMT command "stop remote node".
(6)	Receive NMT command "start remote node".
(9)	Receive NMT command "reset node".
(10)	Receive NMT command "reset communication".

Table 23: NMT state transitions

Services on the listed communication objects may only be executed if the CANopen device is in the appropriate NMT state.

NMT states	Available services				
	SDO	PDO	EMCY	NMT	NMT Node Guarding
'STOPPED'				x	x
'Pre-Operational'	x		x	x	x
'Operational'	x	x	x	x	x

Table 24: NMT states

2.9.1 Network management (NMT) protocol (COB-ID: 0x000, 0x700)

The NMT master controls the NMT slave nodes using the following commands:
- Device control services for initialization and start-up of NMT slave nodes that want to take part in the distributed application.
- Node Guarding of NMT slave nodes.

The NMT commands are unconfirmed.

2.9.2 Start remote node command (COB-ID:0, CS:1)

This command requests the selected slave node(s) to enter the state 'Operational'.
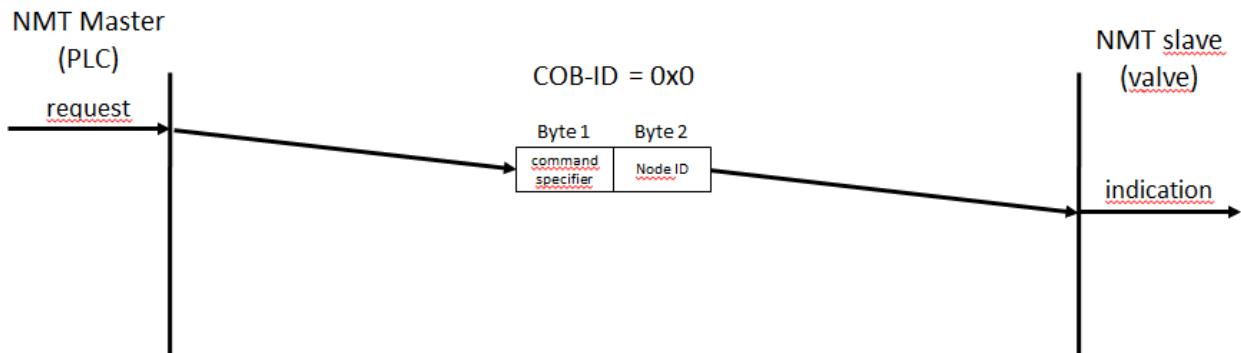
Figure 13: Structure for a start remote node frame
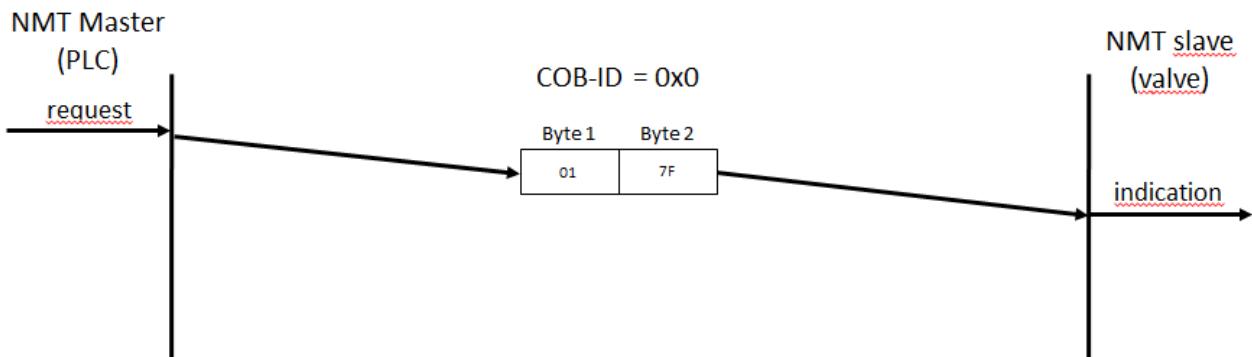
Figure 14: Example for a start remote node frame with Node-ID 127 (0x7F)

2.9.3 Stop remote node command (COB-ID:0, CS:2)

This command requests the selected slave node(s) to enter the state 'Stopped'.

2.9.4 Enter 'Pre-Operational' command (COB-ID:0, CS:128)

This command requests the selected slave node(s) to enter the state 'Pre-Operational'.

2.9.5 Reset node command (COB-ID:0, CS:129)

This command requests the selected slave node(s) to enter the state 'Reset Application'.

All application parameters will be set to their default. The valve application will be re-started. After restarting the application, the reset node calls the reset communication command.

2.9.6 Reset communication command (COB-ID:0, CS:130)

This command requests the selected slave(s) to enter the state 'Reset Communication'.

The PDO communication will be stopped. After re-initialization of the network communication, the NMT state 'Pre-Operational' will be entered.

2.9.7 Bootup message (COB-ID: 0x700)

After a digital servo valve has finished its own initialization and entered the node state 'Pre-Operational' the digital servo valve sends the boot-up message. This message indicates that the slave is ready for operation.

2.9.8 Node guarding (COB-ID: 0x700, RTR:1)

The node guarding is the periodical monitoring of certain network nodes. Each node can be checked by the NMT master with a certain period <GuardTime> (0x100C). A second parameter <LifeTimeFactor> (0x100D) defines a factor after the connection will be detected as lost. The resolution of the guarding time is 1 ms. To enable the node guarding on a slave device, the guard time and life time factor must be set. The guarding is started with the first guarding telegram of the master. During node guarding the master sends a remote trans-mit request (RTR) frame to each guarded slave. The slave answers with its actual NMT state and a toggle bit. This toggle bit alternates in each cycle.

The monitoring time until fault reaction 110 (CAN life guard error or heartbeat error) is triggered is calculated as follows:

Monitoring time = <GuardTime> (0x100C) * <LifeTimeFactor> (0x100D).

2.9.8.1 Object 0x100C: Guard time

This parameter contains the guarding time in milliseconds. The value 0 disables the life guarding.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x100C	0	GuardTime	UINT16	rw	Y	UINT16	0

2.9.8.2 Object 0x100D: Life time factor

This parameter contains the life time factor. The value 0 disables the life guarding.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x100D	0	LifeTimeFactor	UINT8	rw	Y	UINT8	0

2.10 Electronic data sheet (EDS) files

Electronic data sheets (EDS) are files which describe the capabilities and the communication objects of a CANopen device. The EDS is essential to configure CANopen master devices. They are most commonly used when CANopen modules made available to third parties. The digital servo valve EDS file is provided by Moog and can be downloaded from the Moog website http://www.moogsoftwaredownload.com.

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.
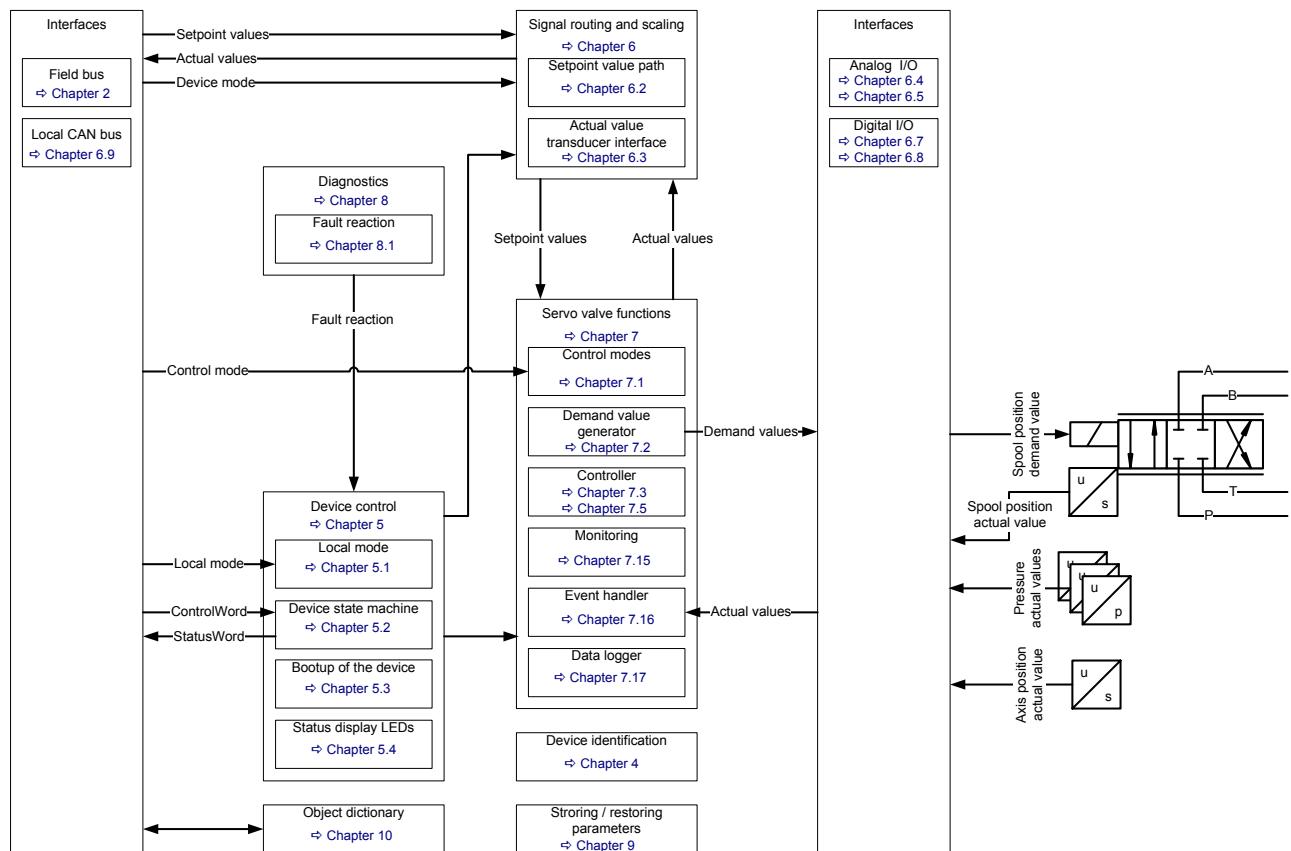
Figure 15: Device structure

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> (0x6042) 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> (0x6043) only a subset of the device structure may be used.
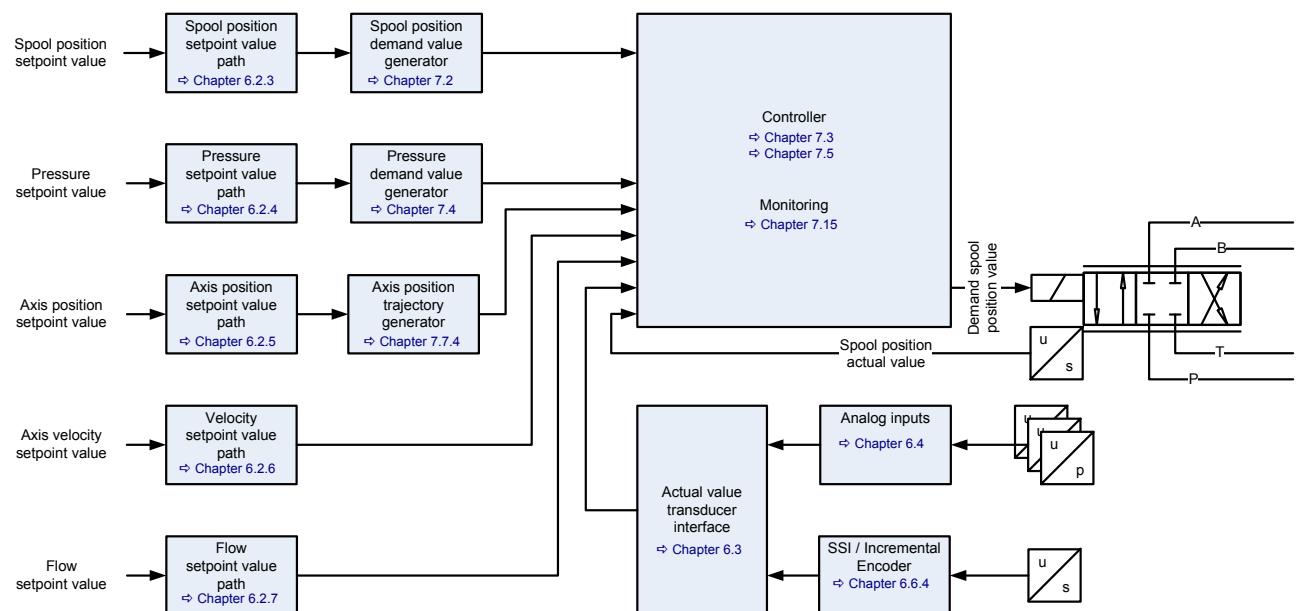
Figure 16: Device controller structure

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 CANopen communication profile defined by CiA 301

4.1.1 Object 0x1000: Device Type

This parameter indicates the code of the underlying device profile. The default value 408 specifies the device profile CiA 408.

Device							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1000	0	DeviceType	UINT32	ro	-	UINT32	408

Value description

<DeviceType>	Description
0	Manufacturer-specific device profile.
408	This device is a servo valve or a pump. Device profile according to CiA 408 "Device profile for fluid power technology proportional valves and hydrostatic transmissions".

Table 25: Possible values of parameter <DeviceType> (0x1000)

4.1.2 Object 0x1008: Manufacturer device name

This parameter indicates the name of the servo valve.

Device							
Index	Sub-index	Parameter name	Data type	Access	Per- sistence	Value range	Default
0x1008	0	ManufacturerDeviceName	STRING	ro	-	None	""

4.1.3 Object 0x1009: Manufacturer hardware version

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per- sistence	Value range	Default
0x1009	0	ManufacturerHardwareVersion	STRING	ro	-	None	""

4.1.4 Object 0x100A: Manufacturer software version

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per- sistence	Value range	Default
0x100A	0	ManufacturerSoftwareVersion	STRING	ro	-	None	Device-specific value

4.1.5 Object 0x1018: Identity object

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1018	1	VendorId	UINT32	ro	-	UINT32	40
0x1018	2	ProductCode	UINT32	ro	-	UINT32	0
0x1018	3	RevisionNumber	UINT32	ro	-	UINT32	0
0x1018	4	SerialNumber	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 17: 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 CANopen 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 "4.2.3 Object 0x6052: Serial number", page 41

Table 26: Possible values of parameter Identity object (0x1018)

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

4.2 Objects defined by Device Profile Fluid Power

4.2.1 Object 0x6050: Version

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6050	0	DeviceVersion	STRING	ro	-	None	""

4.2.2 Object 0x6051: Code number

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6051	0	CodeNumber	UINT16	rw	-	UINT16	0

4.2.3 Object 0x6052: Serial number

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6052	0	SerialNumber	STRING	ro	-	None	""

4.2.4 Object 0x6053: Description

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6053	0	Description	STRING	rw	Y	None	""

4.2.5 Object 0x6054: Model description

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6054	0	ModelDescription	STRING	ro	-	None	""

4.2.6 Object 0x6055: Model URL

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6055	0	ModelURL	STRING	ro	-	None	"www.moog.com"

4.2.7 Object 0x6056: Parameter set code

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6056	0	ParameterSetCode	UINT8	rw	Y	0…254	0

4.2.8 Object 0x6057: Vendor name

This parameter indicates the name of the device vendor.

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6057	0	VendorName	STRING	ro	-	None	"MOOG, Hanns-Klemm-Strasse 28, D-71034 Boeblingen, Germany"

4.2.9 Object 0x605F: Capability

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x605F	0	Capability	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 27: Possible values of parameter <Capability> (0x605F)

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 28: Bit values of parameter <Capability> (0x605F)

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> (0x6040) which can be set via bus using a PDO or locally by the parameter value <LocalControlWord> (0x4040). 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> (0x604F) as shown in the following figure.
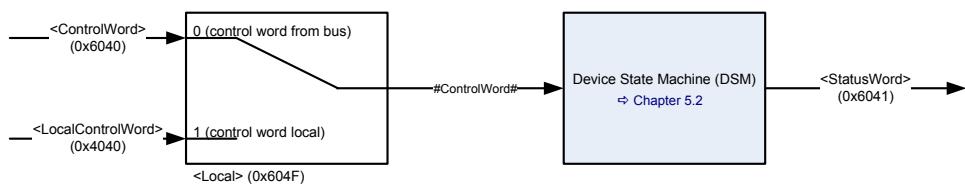
Figure 18: Local mode

5.1.1 Object 0x604F: Local

By writing the value 1 to this parameter, the <LocalControlWord> (0x4040) is used as input signal #ControlWord# for the device state machine (DSM). Setting the <Local> (0x604F) parameter to 0, the <ControlWord> (0x6040) is used as input signal #ControlWord#.

Device							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x604F	0	Local	INT8	rw	Y	0…1	0

Value description

<Local>	Description
0	DSM controlled via bus by the <ControlWord> (0x6040). The <ControlWord> (0x6040) is used as input signal #ControlWord# for the DSM.
1	DSM controlled by the <LocalControlWord> (0x4040). The <LocalControlWord> (0x4040) is used as input signal #ControlWord# for the DSM.

Table 29: Possible values of parameter <Local> (0x604F)

The actual setting of the <Local> (0x604F) parameter is indicated in bit 4 of the <StatusWord> (0x6041).

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

5.1.2 Object 0x6040: Control word

The functionality of the <LocalControlWord> (0x4040) and the <ControlWord> needs to be distinguished. If the parameter <Local> (0x604F) is set, the DSM input #ControlWord# comes from the local parameter <Lo-calControlWord> (0x4040) otherwise it comes from <ControlWord> (0x6040). This bit-coded parameter controls the DSM states.

Device							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6040	0	ControlWord	UINT16	rw	N	UINT16	None

Value description

<ControlWord>				
Bit	Description	<Control Mode> (0x6043) is set to 1…4, 7…14	<Control Mode> (0x6043) 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> (0x6043) 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 30: Possible values of parameter <ControlWord> (0x6040)

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.

Bit 8: Enable pressure controller
This bit activates the pressure controller in the p/Q control mode (<ControlMode> (0x6043) is set to 5).

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

Bit 9: Axis install mode positive direction
If the servo valve is in <DeviceMode> (0x6042) = 3 (Install mode) or <DeviceMode> (0x6042) = 4 (reference mode), then the hydraulic axis can be moved with bit 9 of the <ControlWord> (0x6040) in positive direction. The velocity of the movement is defined in the parameter <ManualMaxSpeed> (0x5614). If an incremental encoder is used to measure the axis position, then <DeviceMode> (0x6042) = 4 (reference mode) is used to run the homing position to calibrate the incremental encoder.

Bit 10: Axis install mode negative direction

If the servo valve is in <DeviceMode> (0x6042) = 3 (Install mode) or <DeviceMode> (0x6042) = 4 (reference mode), then the hydraulic axis can be moved with bit 10 of the <ControlWord> (0x6040) in negative direction. The velocity of the movement is defined in the parameter <ManualMaxSpeed> (0x5614). If an incremental encoder is used to measure the axis position then <DeviceMode> (0x6042) = 4 (reference mode) is used to run the homing position to calibrate the incremental encoder.

⇨ Chapter "6.6.4 Incremental Encoder", page 96
⇨ Chapter "7.7 Axis position setpoint conditioning / demand value generator", page 191

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 140
Pressure demand value generator: ⇨ Chapter "7.4.5 Ramp", page 162

5.1.3 Object 0x4040: Local control word

Parameter description: ⇨ Chapter "5.1.2 Object 0x6040: Control word", page 44

Device							
Index	Sub-index	Parameter name	Data type	Access	Persis- tence	Value range	Default
0x4040	0	LocalControlWord	UINT16	rw	N	UINT16	<LocalControl- WordDefault> (0x403F)

Value description

Same values as <ControlWord> (0x6040)
⇨ Table 30, page 44

5.1.4 Object 0x403F: Local control word default

The <LocalControlWordDefault> (0x403F) defines the control word after power up of the digital servo valve. During startup of the servo valve, the parameter <LocalControlWordDefault> (0x403F) is copied to the parameter <LocalControlWord> (0x4040).

Device							
Index	Sub-index	Parameter name	Data type	Access	Persis- tence	Value range	Default
0x403F	0	LocalControlWordDefault	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> (0x6041) (bits 0…3 of the status word indicate the device condition).
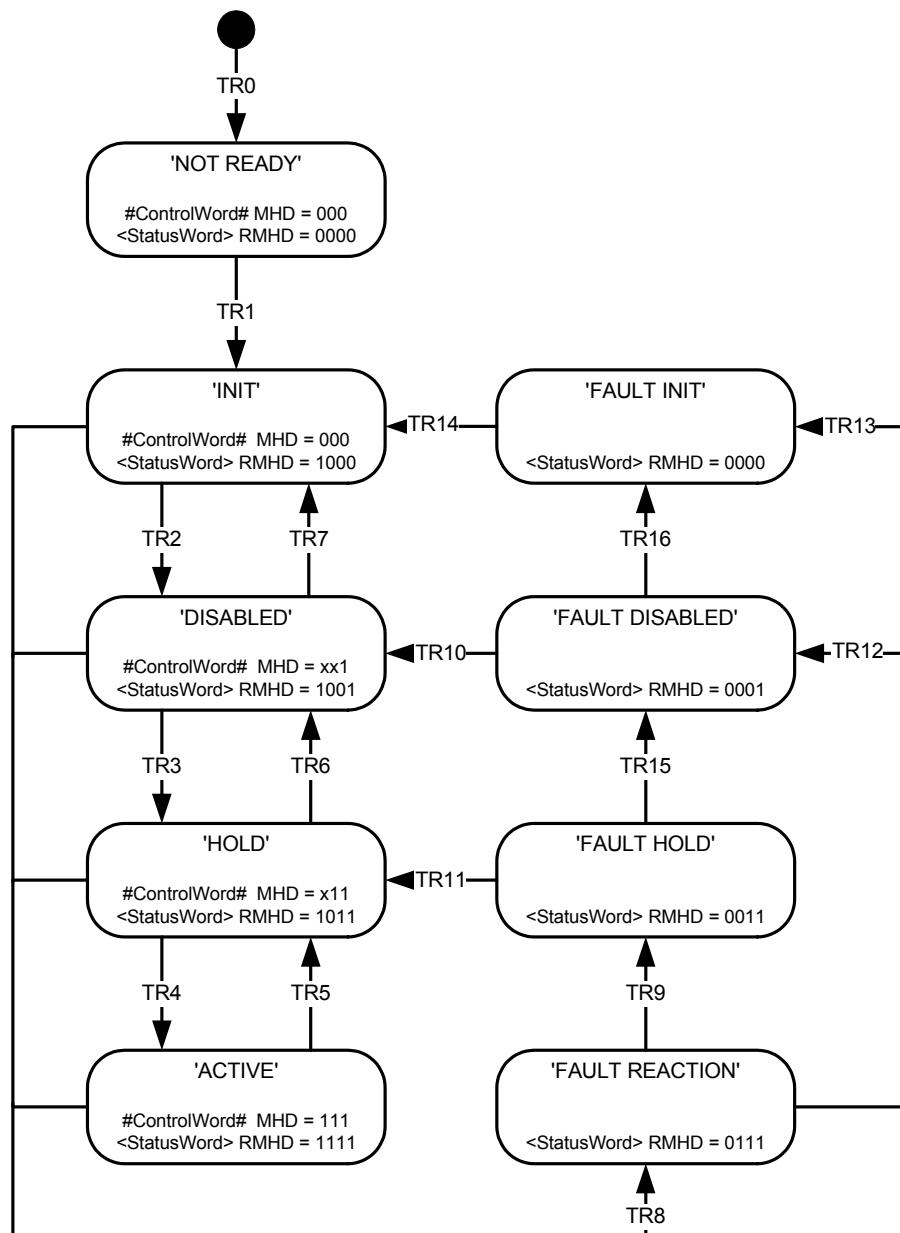
Figure 19: Device state machine

<StatusWord> (0x6041)	#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> (0x6041) 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> (0x6043) is active.
  ⇨ Chapter "7.1 Control modes", page 127
- The setpoint values from the bus or from the analog input according to the chosen <DeviceMode> (0x6042) are not effective.
  ⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 58
- The control loop depending on the selected <ControlMode> (0x6043) is active and the corresponding pre-defined hold setpoint is used, for example, the <SplHoldSetPoint> (0x6314) for the spool control or the <PrsHoldSetPoint> (0x6394) for the pressure control.
  ⇨ Chapter "6.2.3.3 Object 0x6314: Spl hold setpoint", page 60
  ⇨ Chapter "6.2.4.3 Object 0x6394: Prs hold setpoint", page 62
  ⇨ Chapter "6.2.5.3 Object 0x6614: Hold setpoint", page 64
  ⇨ Chapter "6.2.6.3 Object 0x5527: Hold setpoint", page 68
  ⇨ Chapter "6.2.7.3 Object 0x5314: Hold setpoint", page 71

'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> (0x6042) 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> (0x6043) is active.
  ⇨ Chapter "7.1 Control modes", page 127
- The setpoint values from the bus or from the analog input according to the chosen <DeviceMode> (0x6042) are not effective.
  ⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 58
- The control loop depending on the selected <ControlMode> (0x6043) is active and the corresponding pre-defined hold setpoint is used, for example, the <SplHoldSetPoint> (0x6314) for the spool control or the <PrsHoldSetPoint> (0x6394) for the pressure control.
  ⇨ Chapter "6.2.3.3 Object 0x6314: Spl hold setpoint", page 60
  ⇨ Chapter "6.2.4.3 Object 0x6394: Prs hold setpoint", page 62
  ⇨ Chapter "6.2.5.3 Object 0x6614: Hold setpoint", page 64
  ⇨ Chapter "6.2.6.3 Object 0x5527: Hold setpoint", page 68
  ⇨ Chapter "6.2.7.3 Object 0x5314: Hold setpoint", page 71

'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 271
- The resulting fault state depends on the vendor specific <FaultReactionType> (0x2830).
- 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> (0x604F) is set to 1 (Control Word Local) and enable signal on is defined by the parameter <LocalControlModeDefault> (0x403F).
⇨ Chapter "5.1.4 Object 0x403F: Local control word default", page 45

⚠ **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 46

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 43

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 50
TR4	Activate 'ACTIVE'	x	x	x	x	x	1	1	1	Depending on enable signal ⇨ Chapter "5.2.2.4 Enable behavior", page 50
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 50 Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 52
					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 50 Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 52
					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 50 Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 52
					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 52
TR11	Reset 'FAULT_HOLD'	Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 52
TR14	Reset 'FAULT_INIT'	Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 52
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 52
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 52
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 50
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 50
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 50

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 100

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.

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> (0x2420) must be set to 2.

5.2.3 Object 0x6041: Status word

The bit-coded <StatusWord> (0x6041) indicates the current device status.

Device							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6041	0	StatusWord	UINT16	ro	-	UINT16	None

Value description

<StatusWord>				
Bit	<StatusWord> bit	<Control Mode> (0x6043) is set to 1…4, 7…14	<Control Mode> (0x6043) 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> (0x604F) is set			CiA 408
5…7	Reserved			
8	<ControlMode> (0x6043) 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 31: Possible values of parameter <StatusWord> (0x6041)

Bits 0, 1, 2, 3: 'DISABLED', 'HOLD', 'ACTIVE', 'READY'
These bits indicate the state of the device state machine (DSM).

Bit 4: Indicates that bit <Local> (0x604F) is set
The <LocalControlWord> (0x4040) is the active control word.

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 127

ⓘ This bit is only active if the <ControlMode> (0x6043) 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 140
Pressure demand value generator: ⇨ Chapter "7.4.5 Ramp", page 162

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 137
Pressure demand value generator: ⇨ Chapter "7.4.3 Limit function", page 160

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 232

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 140
Pressure demand value generator: ⇨ Chapter "7.4.5 Ramp", page 162

5.2.4 Object 0x1002: Manufacturer Status Register

The <ManufacturerStatusRegister> (0x1002) indicates the current status of the digital input 0 (enable signal).

⇨ Chapter "6.7.1 Digital input 0 (enable signal)", page 100

Device							
Index	Subindex	Name	Data type	Access	Persis-tence	Value range	Default
0x1002	0	ManufacturerStatusRegister	UINT32	ro	-	UINT32	None

5.3 Bootup of the device

The bootup sequence needs about 500 ms. It can be delayed with the <PowerOnDelay> (0x200F) parameter.

5.3.1 Object 0x200F: Power On Delay

The parameter <PowerOnDelay> (0x200F) allows delaying the bootup procedure before establishing the communication and servo valve functions. The <PowerOnDelay> (0x200F) time is provided in seconds.

Device							
Index	Subindex	Name	Data type	Access	Persis-tence	Value range	Default
0x200F	0	PowerOnDelay	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.
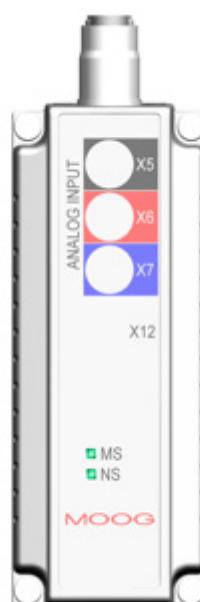
Figure 20: Status display LEDs

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 46

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 271
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 271

5.4.2 Network status LED «NS»

The network status LED displays the state of the network management (NMT) state machine.

⇨ Chapter "2.9 Network management state machine (NMT state machine)", page 32

Network status LED «NS»	NMT state machine (ESM)	Description
Off	Stopped	No power supply or not connected.
Green blinking	'Init' or 'Pre-Operational'	Connected. SDO communication is possible.
Green	'Operational'	Connected. SDO and PDO communication are possible.
Red		A network major error has occured.

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> (0x6043) used. The blocks with gray backgrounds are described in detail in this chapter.

⇨ Chapter "7.1 Control modes", page 127
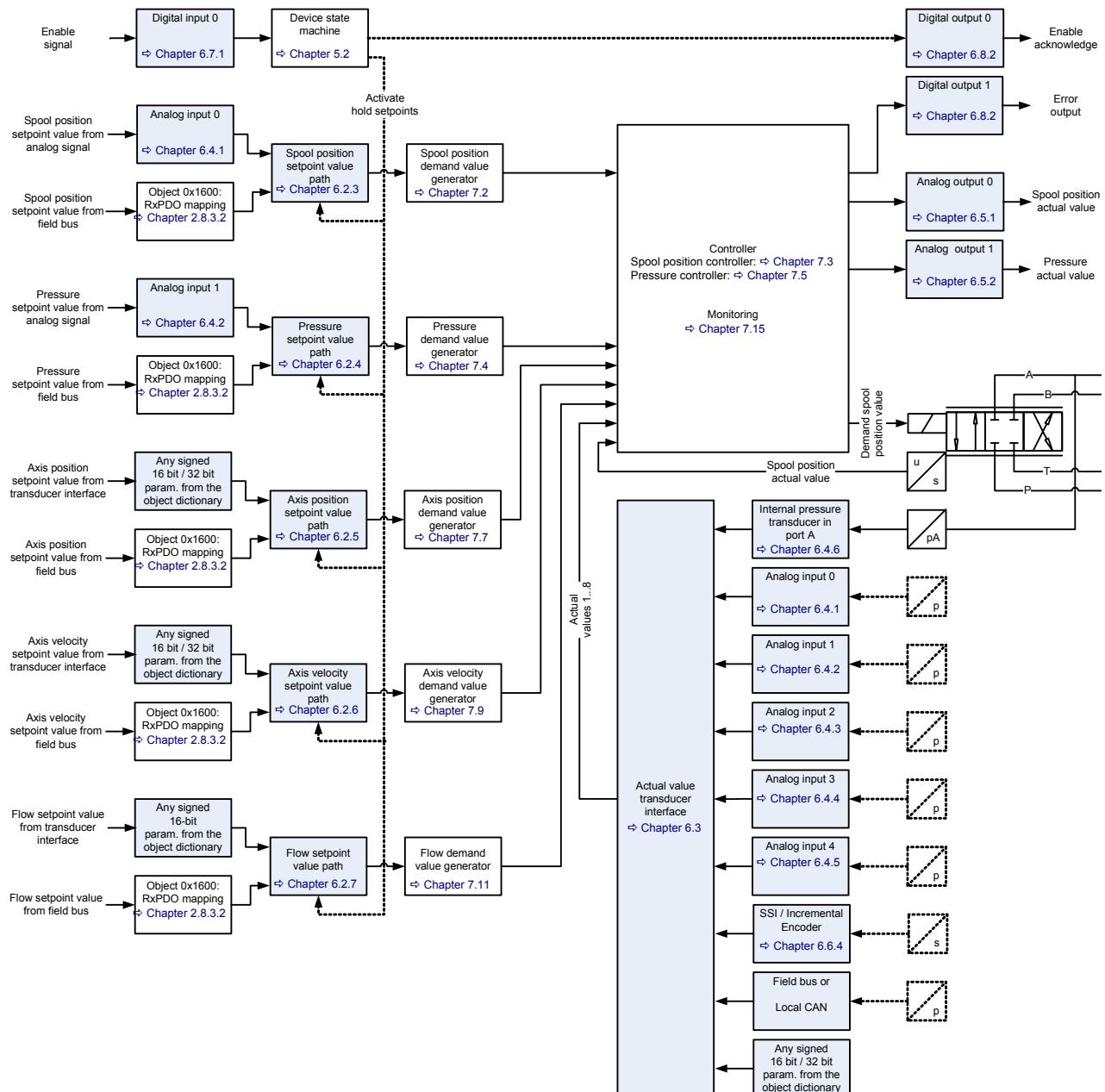
Figure 21: Signal routing

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> (0x6300).
- The pressure setpoint value received via bus is provided by the parameter <PrsSetpoint> (0x6380).
- The axis position setpoint value received via bus is provided by the parameter <PosSetpoint> (0x6600).
- The axis velocity setpoint value received via bus is provided by the parameter <VelSetpoint> (0x6500).
- The flow setpoint value received via bus is provided by the parameter <FlwSetpoint> (0x5300).
- The spool position setpoint value from local source is coming from the analog input 0 <AnaInpActualValue0> (0x3204).
- The pressure setpoint value from local source is coming from the analog input 1 <AnaInpActualValue1> (0x320C).
- 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> (0x6042) and the <StatusWord> (0x6041):

- <DeviceMode> (0x6042) is set to 1 (setpoint input via bus), <StatusWord> (0x6041) is 0111_b ('ACTIVE'): The setpoint value received via field bus is forwarded to the demand value generator.
- <DeviceMode> (0x6042) is set to 2 (setpoint input locally), <StatusWord> (0x6041) is 0111_b ('ACTIVE'): The setpoint value coming from the analog input is forwarded to the demand value generator.
- <StatusWord> (0x6041) 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> (0x6042) is set to 1 (setpoint input via bus) or <DeviceMode> (0x6042) is set to 2 (setpoint input locally).
  The spool position hold setpoint value is stored in the parameter <SplHoldSetpoint> (0x6314). The pressure hold setpoint value is stored in the parameter <PrsHoldSetpoint> (0x6394). In the same manner the setpoints for the axis position, velocity and flow are stored in the parameter <PosHoldSetpoint> (0x6614), <VelHoldSetpoint> (0x6394) and <FlwHoldSetpoint> (0x5313).

6.2.1 Object 0x6042: 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> (0x6042) = 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> (0x6040). The <DeviceMode> (0x6042) = 4 (reference mode) is used to enable the referencing run for an incremental encoder.

⇨ Chapter "5.1.2 Object 0x6040: Control word", page 44
⇨ Chapter "6.6.4 Incremental Encoder", page 96
⇨ Chapter "7.7 Axis position setpoint conditioning / demand value generator", page 191

Device							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6042	0	DeviceMode	INT8	rw	N	1…4	<DeviceMode-Default> (0x4042)

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 32: Possible values of parameter <DeviceMode> (0x6042)

ⓘ The effective <DeviceMode> (0x6042) after power up is defined by the parameter <DeviceModeDefault> (0x4042).
⇨ Chapter "6.2.2 Object 0x4042: Device mode default", page 58

6.2.2 Object 0x4042: Device mode default

The <DeviceModeDefault> (0x4042) defines the active device mode after power up of the servo valve. This is achieved by automatically copying the parameter <DeviceModeDefault> (0x4042) to the parameter <DeviceMode> (0x6042) during the startup procedure of the servo valve

Device							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x4042	0x00	DeviceModeDefault	INT8	rw	Y	1…2	1

6.2.3 Spool position setpoint value path
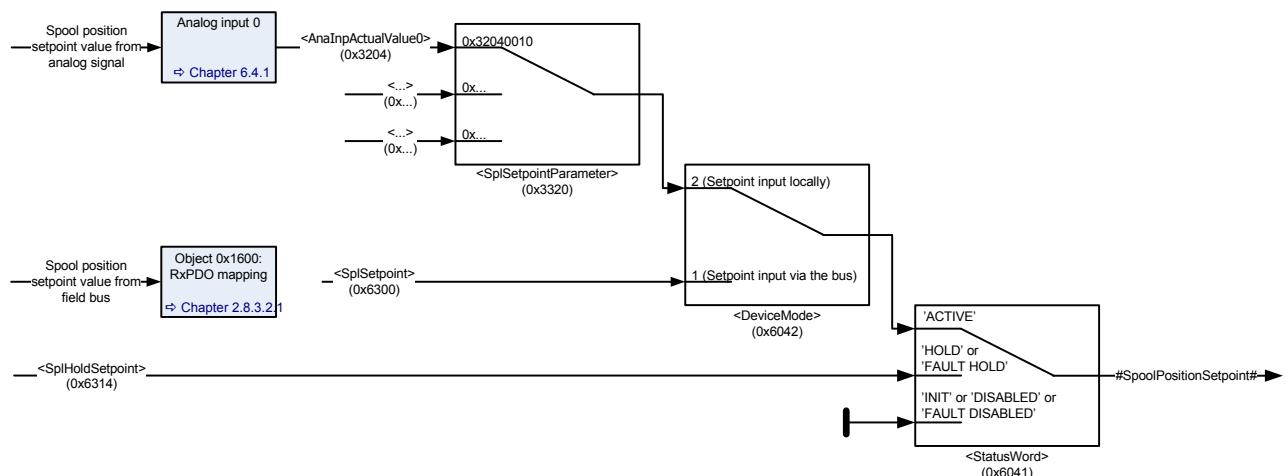
Figure 22: Spool position setpoint value path

6.2.3.1 Object 0x6300: Setpoint

This parameter contains the spool position setpoint value which is received from the field bus. Depending on the <DeviceMode> (0x6042), this parameter is in effect for the following three control modes stored in the parameter <ControlMode> (0x6043):

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> (0x6300) takes only effect if the <StatusWord> (0x6041) is 0111_b ('AC-TIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 52
⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 58

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6300	1	SplSetpoint	INT16	rw	N	INT16	None
0x6300	2	Unit	UINT8	ro	-	UINT8	0
0x6300	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

6.2.3.2 Object 0x3320: Setpoint parameter

The spool position setpoint value parameter <SplSetpointParameter> (0x3320) points to the input where the spool position setpoint value <SplSetpoint> (0x6300) comes from.

The spool position setpoint value <SplSetpoint> (0x6300) is only effective in case the <StatusWord> (0x6041) is 0111_b ('ACTIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 52
⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 58

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3320	0	SplSetpointParameter	INT32	rw	-	INT32	0x63000110

Value description

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

Table 33: Possible values of parameter <SplSetpointParameter> (0x3320)

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> (0x6300), sub-index 0x01 with a length of 16 bit (16=0x10).

6.2.3.3 Object 0x6314: Spl hold setpoint

This parameter defines the spool position hold setpoint value for the <ControlMode> (0x6043):

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 0x6043: Control mode", page 128

The <SplHoldSetpoint> (0x6314) acts as setpoint value in case of <StatusWord> (0x6041) is 1011_b ('HOLD') or 0011_b ('FAULT_HOLD').

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 52

ValvePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6314	1	SplHoldSetpoint	INT16	rw	Y	INT16	0
0x6314	2	Unit	UINT8	ro	-	UINT8	0
0x6314	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

6.2.4 Pressure setpoint value path
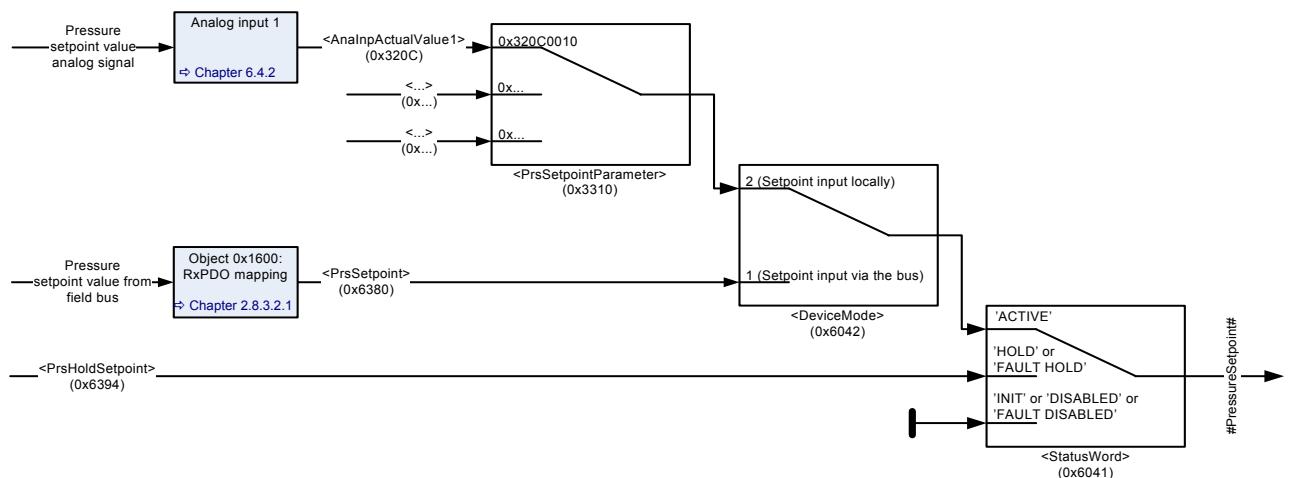
Figure 23: Pressure setpoint value path

6.2.4.1 Object 0x6380: Setpoint

This parameter contains the pressure setpoint value which is received from the field bus. Depending on the <DeviceMode> (0x6042), this parameter is in effect for the following three control modes stored in the parameter <ControlMode> (0x6043):

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

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 128

The setpoint value takes only effect in case the <StatusWord> (0x6041) is 1111_b ('ACTIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 52
⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 58

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6380	1	PrsSetpoint	INT16	rw	N	INT16	None
0x6380	2	Unit	UINT8	ro	-	UINT8	0
0x6380	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

6.2.4.2 Object 0x3310: Setpoint parameter

The pressure setpoint value parameter <PrsSetpointParameter> (0x3310) points to the input where the pressure setpoint value <PrsSetpoint> (0x6380) comes from.

The pressure setpoint value <PrsSetpoint> (0x6380) is only effective in case the <StatusWord> (0x6041) is 1111_b ('ACTIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3310	0x00	PrsSetpointParameter	INT32	rw	Y	INT32	0x63800110

Value description

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

Table 34: Possible values of parameter <PrsSetpointParameter> (0x3310)

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> (0x6380), sub-index 0x01 with the length of 16 bit (16 = 0x10).

6.2.4.3 Object 0x6394: Prs hold setpoint

This parameter defines the pressure hold setpoint value. It is effective for the following <ControlMode> (0x6043):

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

The <PrsHoldSetpoint> (0x6394) acts as setpoint value in case of <StatusWord> (0x6041) equals 'HOLD' or 'FAULT_HOLD'.

ValvePressureControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6394	1	PrsHoldSetPoint	INT16	rw	Y	INT16	0
0x6394	2	Unit	UINT8	ro	-	UINT8	0
0x6394	3	Prefix	INT8	ro	-	INT8	0

6.2.5 Axis position setpoint value path
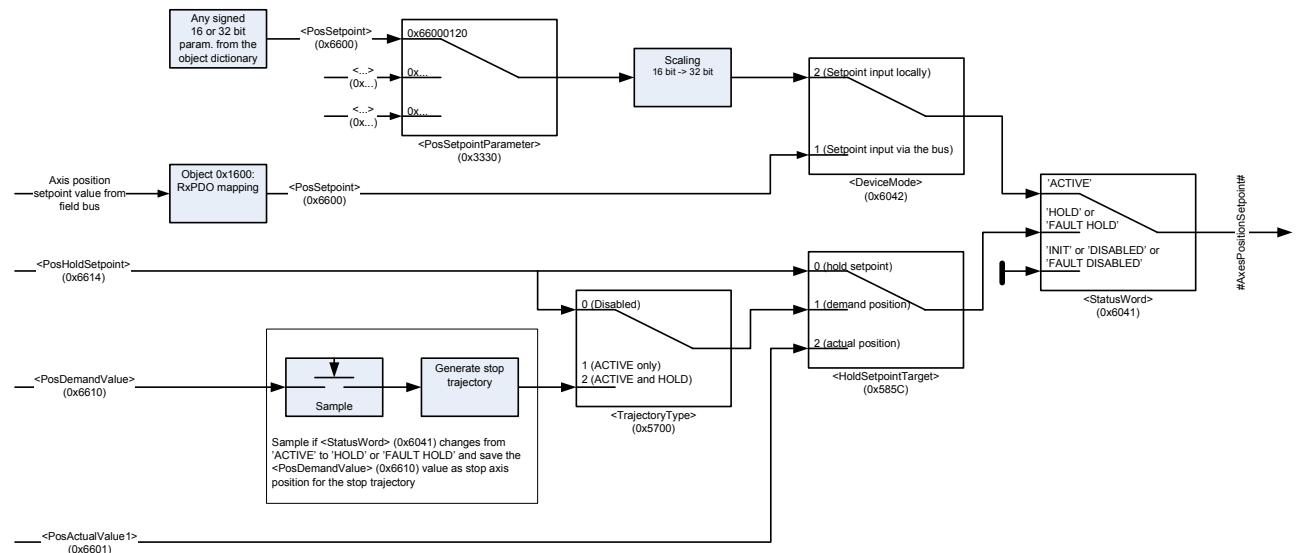
Figure 24: Axis position setpoint value path

⇨ Chapter "7.7.1 Object 0x6610: Position demand value", page 191

6.2.5.1 Object 0x6600: Setpoint

This parameter contains the axis position setpoint value which is received from the field bus. Depending on the <DeviceMode> (0x6042), this parameter is in effect only for the <ControlMode> (0x6043) = 8 (axis position control).

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 128

The setpoint value takes only effect in case the <StatusWord> (0x6041) is 1111_b ('ACTIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 52
⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 58

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6600	1	PosSetpoint	INT32	rw	N	INT32	None
0x6600	2	Unit	UINT8	ro	-	UINT8	0
0x6600	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

6.2.5.2 Object 0x3330: Setpoint parameter

The axis position setpoint value parameter <PosSetpointParameter> (0x3330) points to the input where the axis position setpoint value <PosSetpoint> (0x6600) comes from.

The axis position setpoint value <PosSetpoint> (0x6600) is only effective in case the <StatusWord> (0x6041) is 1111_b ('ACTIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).

DrivePositionControl								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3300	0x00	PosSetpointParameter	INT32	rw	Y	INT32	0x66000120

Value description

<PosSetpointParameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-Index	Parameter bit length
Default	0x63	0x80	0x01	0x20

Table 35: Possible values of parameter <PosSetpointParameter> (0x3300)

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> (0x6600), sub-index 0x01 with the length of 32 bit (32 = 0x20).

6.2.5.3 Object 0x6614: Hold setpoint

This parameter defines the axis position hold setpoint value. This parameter is in effect only for the <ControlMode> (0x6043) = 8 (axis position control).
⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 128

The <PosHoldSetpoint> (0x6614) acts as setpoint value in case of <StatusWord> (0x6041) equals 'HOLD' or 'FAULT_HOLD'.
⇨ Chapter "5.2.3 Object 0x6041: Status word", page 52

DrivePositionControl								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6614	1	PosHoldSetpoint	INT32	rw	Y	INT32	0
0x6614	2	Unit	UINT8	ro	-	UINT8	0
0x6614	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

6.2.5.4 16 to 32 bit scaling

Only if a 16 bit parameter with the <PosSetpointParameter> (0x3330) is mapped, a scaling of 16 bit input value is available.
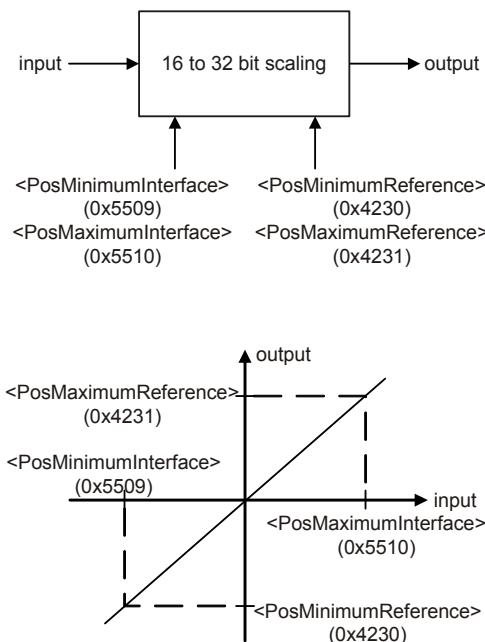
Figure 25: Axis position setpoint value path - 16 to 32 bit scaling

6.2.5.4.1 Object 0x4230: Minimum reference

This parameter together with the parameter <PosMinimumInterface> (0x5509) results in the first point of the linear scaling function.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x4230	0x00	PosMinimumReference	INT32	rw	Y	INT32	−16384

6.2.5.4.2 Object 0x4231: Maximum reference

This parameter together with the parameter <PosMaximumInterface> (0x5510) results in the second point of the linear scaling function.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x4231	0x00	PosMaximumReference	INT32	rw	Y	INT32	16384

6.2.5.4.3 Object 0x5509: Minimum interface

This parameter together with the parameter <PosMinimumReference> (0x4230) results in the first point of the linear scaling function.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5509	0x00	PosMinimumInterface	INT32	rw	Y	INT32	−16384

6.2.5.4.4 Object 0x5510: Maximum interface

This parameter together with the parameter <PosMaximumReference> (0x4231) results in the second point of the linear scaling function.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5510	0x00	PosMaximumInterface	INT32	rw	Y	INT32	16384

6.2.5.5 Object 0x5700: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5700	0	TrajectoryType	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 193
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> (0x6041) is in the state 'ACTIVE'. ⇨ Chapter "7.7.4 Axis position trajectory generator", page 193
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> (0x6041) is in the state 'ACTIVE' or 'HOLD'. ⇨ Chapter "7.7.4 Axis position trajectory generator", page 193

Table 36: Possible values of parameter <TrajectoryType> (0x5700

6.2.5.6 Object 0x585C: Hold setpoint target

DrivePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x585C	0	HoldSetpointTarget	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> (0x6614) while the <StatusWord> (0x6041) is changing from 'ACTIVE' to 'HOLD'.
1 (demand position)	The target position of the trajectory is jumping to the <DemandValue> (0x6610) while the <StatusWord> (0x6041) 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> (0x6601) while the <StatusWord> (0x6041) is changing from 'ACTIVE' to 'HOLD'.

Table 37: Possible values of parameter <HoldSetpointTarget> (0x585C)

6.2.6 Axis velocity setpoint value path
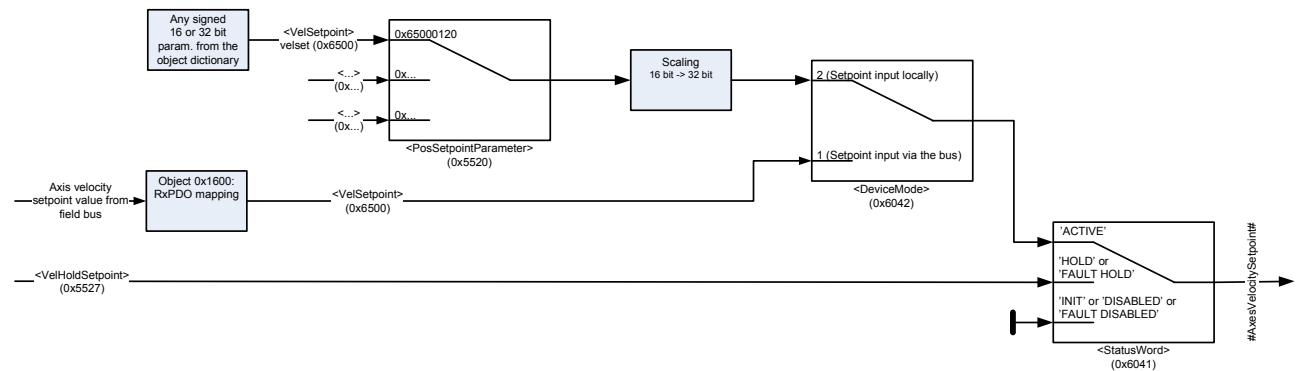
Figure 26: Axis velocity setpoint value path

6.2.6.1 Object 0x6500: Setpoint

This parameter contains the axis velocity setpoint value which is received from the field bus. Depending on the <DeviceMode> (0x6042), this parameter is in effect only for the <ControlMode> (0x6043) = 6 (axis velocity control).

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 128

The setpoint value takes only effect in case the <StatusWord> (0x6041) is 1111_b ('ACTIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 52
⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 58

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6500	1	VelSetpoint	INT32	rw	N	INT32	None
0x6500	2	Unit	UINT8	ro	-	UINT8	0
0x6500	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

6.2.6.2 Object 0x5520: Setpoint parameter

The axis velocity setpoint value parameter <VelSetpointParameter> (0x5520) points to the input where the axis velocity setpoint value <VelSetpoint> (0x6500) comes from.

The axis velocity setpoint value <VelSetpoint> (0x6500) is only effective in case the <StatusWord> (0x6041) is 1111_b ('ACTIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).

DriveVelocityControl								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5520	0x00	VelSetpointParameter	INT32	rw	Y	INT32	0x65000120

Value description

<VelSetpointParameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-Index	Parameter bit length
Default	0x65	0x00	0x01	0x20

Table 38: Possible values of parameter <VelSetpointParameter> (0x5520)

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> (0x6500), sub-index 0x01 with the length of 32 bit (32 = 0x20).

6.2.6.3 Object 0x5527: Hold setpoint

This parameter defines the axis velocity hold setpoint value. This parameter is in effect only for the <ControlMode> (0x6043) = 6 (axis velocity control).
⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 128

The <VelHoldSetpoint> (0x5527) acts as setpoint value in case of <StatusWord> (0x6041) equals 'HOLD' or 'FAULT_HOLD'.
⇨ Chapter "5.2.3 Object 0x6041: Status word", page 52

DriveVelocityControl_DemandValueGenerator								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5527	1	VelHoldSetpoint	INT32	rw	N	INT32	0
0x5527	2	Unit	UINT8	ro	-	UINT8	0
0x5527	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

6.2.6.4 16 to 32 bit scaling

Only if a 16 bit parameter with the <VelSetpointParameter> (0x5520) is mapped, a scaling of 16 bit input value is available.
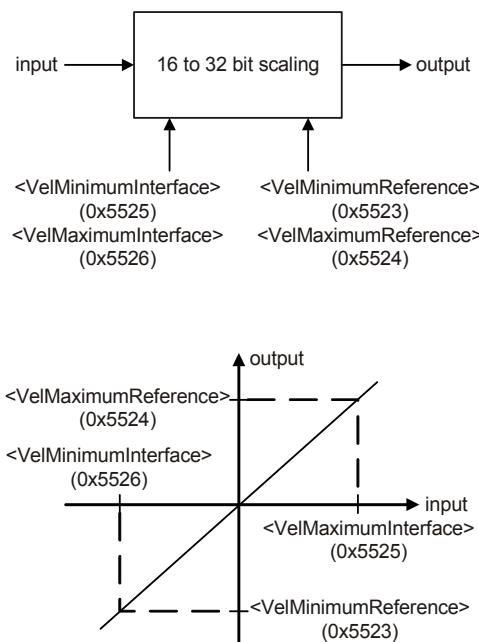
Figure 27: Axis velocity setpoint value path - 16 to 32 bit scaling

6.2.6.4.1 Object 0x5523: Minimum reference

This parameter together with the parameter <VelMinimumInterface> (0x5525) results in the first point of the linear scaling function.

DriveVelocityControl								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5523	0x00	VelMinimumReference	INT32	rw	Y	INT32	−16384

6.2.6.4.2 Object 0x5524: Maximum reference

This parameter together with the parameter <VelMaximumInterface> (0x5526) results in the second point of the linear scaling function.

DriveVelocityControl								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5524	0x00	VelMaximumReference	INT32	rw	Y	INT32	16384

6.2.6.4.3 Object 0x5525: Minimum interface

This parameter together with the parameter <VelMinimumReference> (0x5523) results in the first point of the linear scaling.

DriveVelocityControl								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5525	0x00	VelMinimumInterface	INT32	rw	Y	INT32	−16384

6.2.6.4.4 Object 0x5526: Maximum interface

This parameter together with the parameter <VelMaximumReference> (0x5524) results in the second point of the linear scaling function.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5526	0x00	VelMaximumInterface	INT32	rw	Y	INT32	16384

6.2.7 Flow setpoint value path
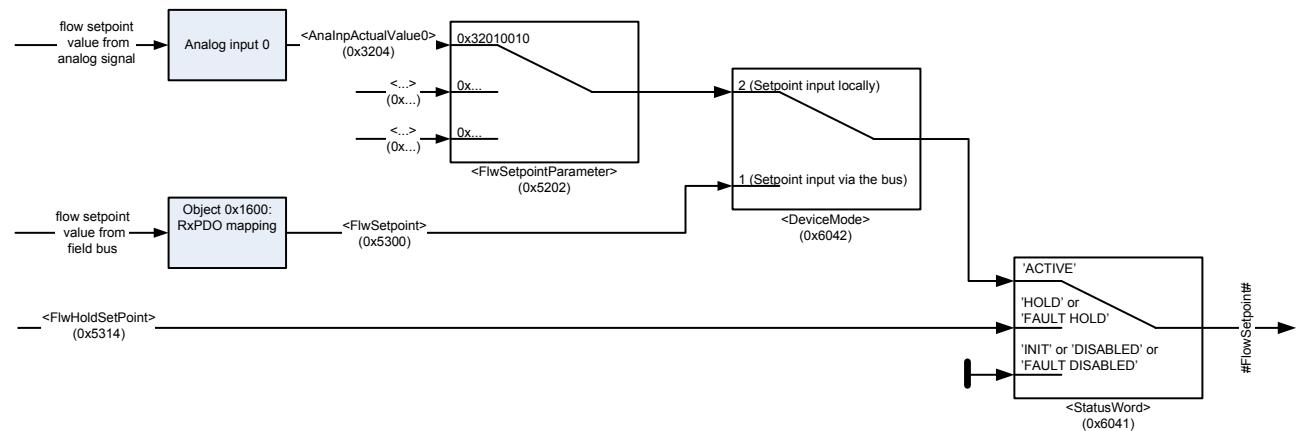
Figure 28: Flow setpoint value path

6.2.7.1 Object 0x5300: Setpoint

This parameter contains the flow setpoint value which is received from the field bus. Depending on the <DeviceMode> (0x6042), this parameter is in effect for the following control modes stored in the parameter <ControlMode> (0x6043):

13   flow control
14   p/flow control

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 128

The setpoint value takes only effect in case the <StatusWord> (0x6041) is 1111_b ('ACTIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).
⇨ Chapter "5.2.3 Object 0x6041: Status word", page 52
⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 58

DriveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5300	1	FlwSetpoint	INT16	rw	N	INT16	None
0x5300	2	Unit	UINT8	ro	-	UINT8	0
0x5300	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

6.2.7.2 Object 0x5202: Setpoint parameter

The flow setpoint value parameter <FlwSetpointParameter> (0x5202) points to the input where the flow set-point value <FlwSetpoint> (0x5300) comes from.

The flow setpoint value <FlwSetpoint> (0x5300) is only effective in case the <StatusWord> (0x6041) is 1111_b ('ACTIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).

DriveFlowControl								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5202	1	FlwSetpointParameter	INT32	rw	N	INT32	0x53000110

Value description

<FlwSetpointParameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-Index	Parameter bit length
Default	0x53	0x00	0x01	0x10

Table 39: Possible values of parameter <FlwSetpointParameter> (0x5202)

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> (0x5300), sub-index 0x01 with the length of 16 bit (16 = 0x10).

6.2.7.3 Object 0x5314: Hold setpoint

This parameter defines the flow hold setpoint value. It is effective for the following <ControlMode> (0x6043):

13 Flow control
14 p/flow control

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 128

The <FlwHoldSetpoint> (0x5314) acts as setpoint value in case of <StatusWord> (0x6041) equals 'HOLD' or 'FAULT_HOLD'.
⇨ Chapter "5.2.3 Object 0x6041: Status word", page 52

DriveFlowControl_DemandValueGenerator								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5314	1	FlwHoldSetpoint	INT32	rw	Y	INT32	0
0x5314	2	Unit	UINT8	ro	-	UINT8	0
0x5314	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

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> (0x6202), the type of value conditioning can be selected depending on the sensor.
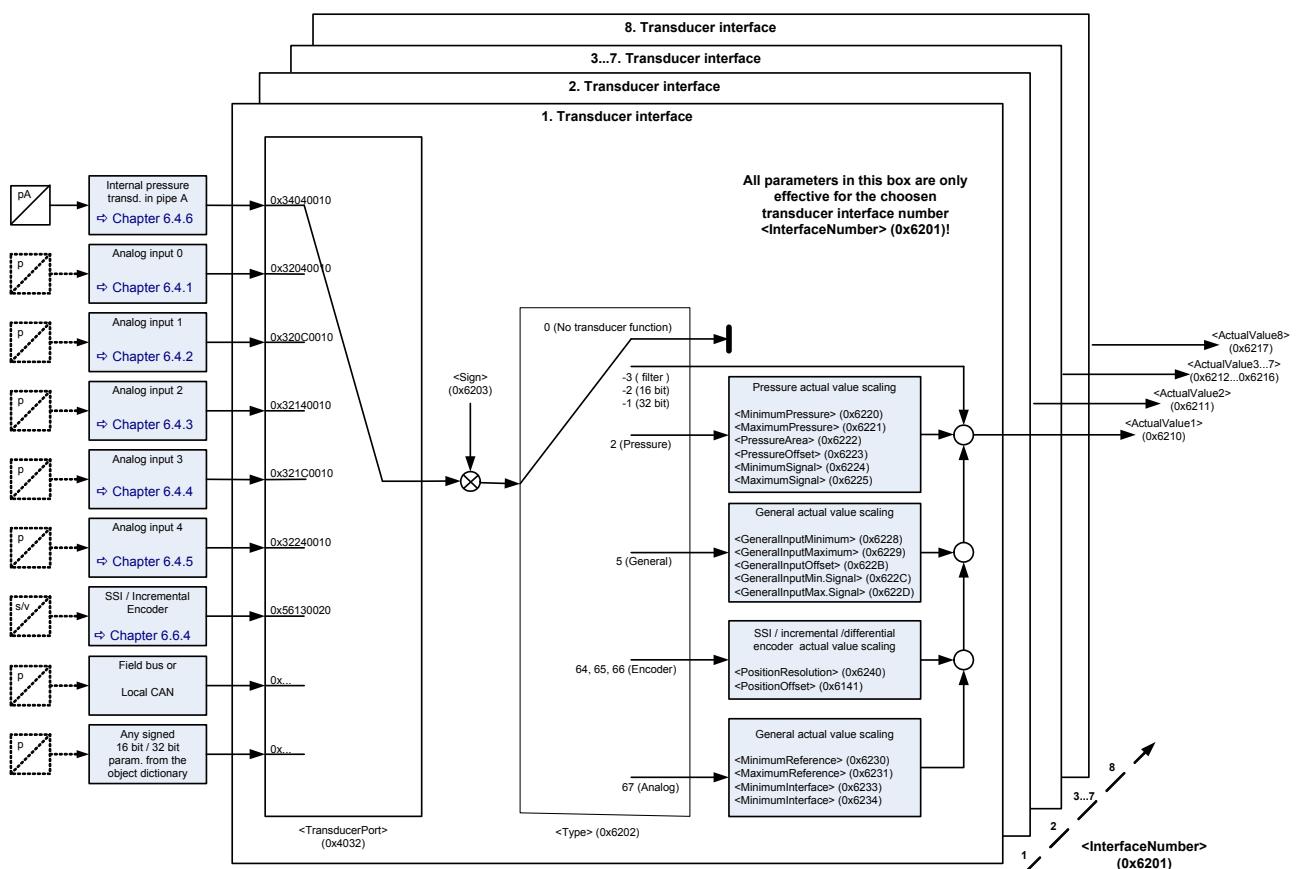
Figure 29: Drive actual value path

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 86

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> (0x6201).

When changing the <InterfaceNumber> (0x6201), the following parameters represent the configuration of the selected interface.

Index	Object Name
0x6200	Max interface number
0x6201	Interface number
0x6202	Type
0x6203	Sign
0x6204	Actual value
0x4032	Transducer port
0x6220	Minimum pressure
0x6221	Maximum pressure
0x6222	Pressure area
0x6223	Pressure offset
0x6224	Minimum transducer signal
0x6225	Maximum transducer signal
0x6240	Position resolution
0x6241	Position offset
0x6230	Minimum reference
0x6231	Maximum reference
0x6233	Minimum interface
0x6234	Maximum interface

6.3.1 Object 0x6210: Actual value 1

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6210	1	ActualValue1	INT32	ro	-	INT32	None

6.3.2 Object 0x6211: Actual value 2

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6211	1	ActualValue2	INT32	ro	-	INT32	None

6.3.3 Object 0x6212: Actual value 3

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6212	1	ActualValue3	INT32	ro	-	INT32	None

6.3.4 Object 0x6213: Actual value 4

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6213	1	ActualValue4	INT32	ro	-	INT32	None

6.3.5 Object 0x6214: Actual value 5

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6214	1	ActualValue5	INT32	ro	-	INT32	None

6.3.6 Object 0x6215: Actual value 6

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6215	1	ActualValue6	INT32	ro	-	INT32	None

6.3.7 Object 0x6216: Actual value 7

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6216	1	ActualValue7	INT32	ro	-	INT32	None

6.3.8 Object 0x6217: Actual value 8

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6217	1	ActualValue8	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> (0x4032). The type of the actual value conditioning is configured by the parameter <Type> (0x6202). All parameters configuring the interface only apply to the interface selected by the parameter <InterfaceNumber> (0x6201).

6.3.9.1 Object 0x6200: Max interface number

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6200	0	MaxInterfaceNumber	UINT8	ro	-	UINT8	8

6.3.9.2 Object 0x6201: Interface number

This parameter defines the actual referenced interface.

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6201	0	InterfaceNumber	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 40: Possible values of parameter <InterfaceNumber> (0x6201)

6.3.9.3 Object 0x6202: Type

This interface type <Type> (0x6202) defines the method of the value conditioning.
This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6201).

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6202	0	Type	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 78
5 (general input)	General 16 or 32 bit input from the object dictionary ⇨ Chapter "6.3.13 General input scaling", page 82
64 (position incremental)	Incremental encoder with A/B channels ⇨ Chapter "6.3.11 Position digital encoder actual value scaling", page 80
65 (position SSI binary)	SSI possition encoder with binary coded signal ⇨ Chapter "6.3.11 Position digital encoder actual value scaling", page 80
66 (position SSI gray)	SSI possition encoder with gray coded signal ⇨ Chapter "6.3.11 Position digital encoder actual value scaling", page 80
67 (position analog)	Use analog input as position signal ⇨ Chapter "6.3.12 Position analog encoder actual value scaling", page 81
−1 (position direct 32 bit)	32 bit signal with no further scaling ⇨ Chapter "6.3.11 Position digital encoder actual value scaling", page 80
−2 (position direct 16 bit)	16 bit signal with no further scaling ⇨ Chapter "6.3.11 Position digital encoder actual value scaling", page 80
All other values	Not used.

Table 41: Possible values of parameter <Type> (0x6202)

6.3.9.4 Object 0x6203: Sign

This parameter defines the sign of the actual value.
This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6201).

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6203	0	Sign	INT8	rw	N	−1 or +1	1

Value description

<Sign>	Description
−1	Negative
1	Positive
All other values	Reserved

Table 42: Possible values of parameter <Sign> (0x6203)

6.3.9.5 Object 0x6204: Actual value

Compared to the objects <Actual value 1…8> (0x6210…0x6217), this parameter contains the output value of the actual selected interface.

Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
Drive_ActualValueConditioning							
0x6204	0	ActualValue	INT32	ro	-	INT32	None

6.3.9.6 Object 0x4032: Transducer port

This parameter defines the transducer port where the actual physical values are coming from. The port is defined by a parameter index, sub-index and length.
This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6201).

Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
Drive_ActualValueConditioning							
0x4032	0	TransducerPort	UINT32	rw	N	UINT32	None

Value description

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

Table 43: Possible values of parameter <TransducerPort> (0x4032)

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> (0x4032) parameter as described in the examples below.

If an incremental encoder or an SSS encoder is selected (<Type> (0x6202) = 64, 65, 66), then the <TransducerPort> (0x4032) is fixed to 0x56130020 which is the index of the encoder input <EncoderValue> (0x5613).

Connector	Index of <ActualValue>	Sub-index of <ActualValue>	Parameter length	Resulting <TransducerPort> value
Analog input 0	0x3204	0x00	0x10	0x32040010
Analog input 1	0x320C	0x00	0x10	0x320C0010
Analog input 2	0x3214	0x00	0x10	0x32140010
Analog input 3	0x321C	0x00	0x10	0x321C0010
Analog input 4	0x3224	0x00	0x10	0x32240010
Internal pressure transducer	0x3404	0x00	0x10	0x34040010
Position encoder	0x5613	0x00	0x20	0x56130020

6.3.9.7 Object 0x3271: Drive transducer structure

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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3271	0	DriveTransducerStructure	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> (0x6202).

This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6201).

⇨ Chapter "6.3.9.3 Object 0x6202: Type", page 76

y(x): <ActualValue> (0x6204)

x_1: <MinimumTransducerSignal> (0x6224)
x_2: <MaximumTransducerSignal> (0x6225)
y_1: <MinimumPressure> (0x6220)
y_2: <MaximumPressure> (0x6221)
y_{off}: <PressureOffset> (0x6223)

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

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$
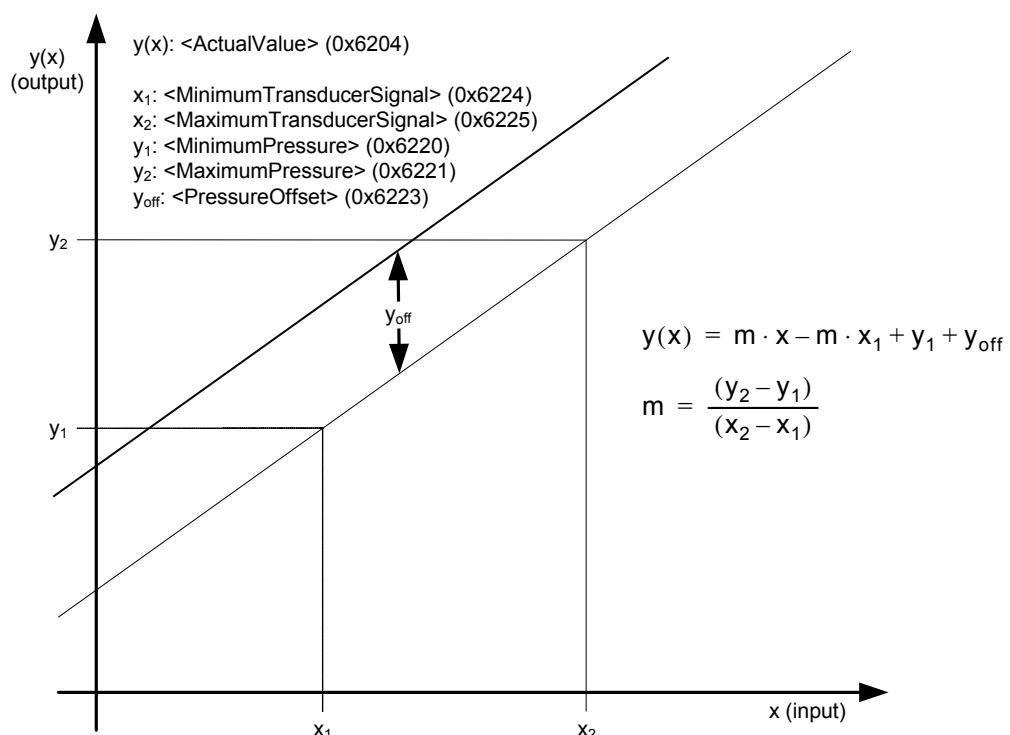
Figure 30: Pressure actual value scaling

6.3.10.1 Object 0x6220: Minimum pressure

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6220	1	MinimumPressure	INT32	rw	N	INT16	16384

6.3.10.2 Object 0x6221: Maximum pressure

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6221	1	MaximumPressure	INT32	rw	N	INT16	16384

6.3.10.3 Object 0x6224: Minimum transducer signal

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6224	1	MinimumTransducerSignal	INT32	rw	N	INT16	16384

6.3.10.4 Object 0x6225: Maximum transducer signal

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6225	1	MaximumTransducerSignal	INT32	rw	N	INT16	16384

6.3.10.5 Object 0x6222: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6222	1	PressureArea	INT32	rw	N	INT16	0

6.3.10.6 Object 0x6223: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6223	1	PressureOffset	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> (0x6202).

This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6201).
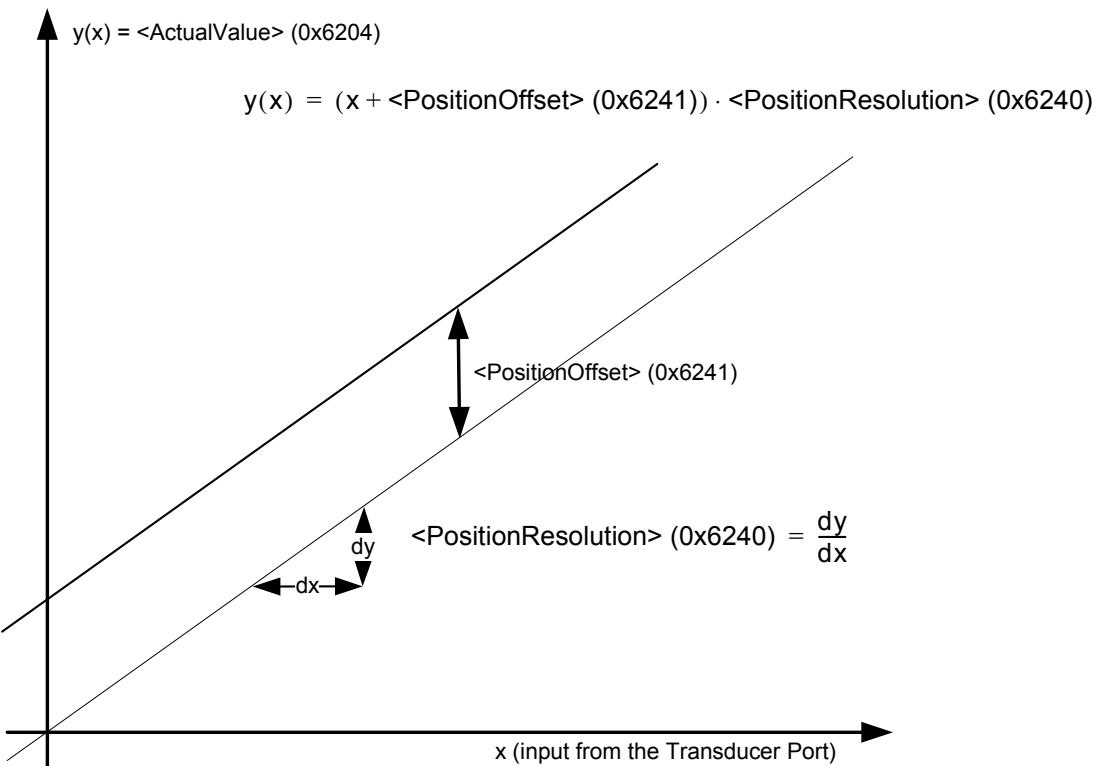
Figure 31: Position digital encoder actual value scaling

6.3.11.1 Object 0x6240: Position resolution

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6240	1	PositionResolution	INT32	rw	N	INT32	1

6.3.11.2 Object 0x6241: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6241	1	PositionOffset	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> (0x6202).

This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6201).

⇨ Chapter "6.3.9.3 Object 0x6202: Type", page 76

y(x): <ActualValue> (0x6204)

x_1: <MinimumInterface> (0x6230)
x_2: <MaximumInterface> (0x6231)
y_1: <MinimumReference> (0x6233)
y_2: <MaximumReference> (0x6234)

$$y(x) = m \cdot x - m \cdot x_1 + y_1$$

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$
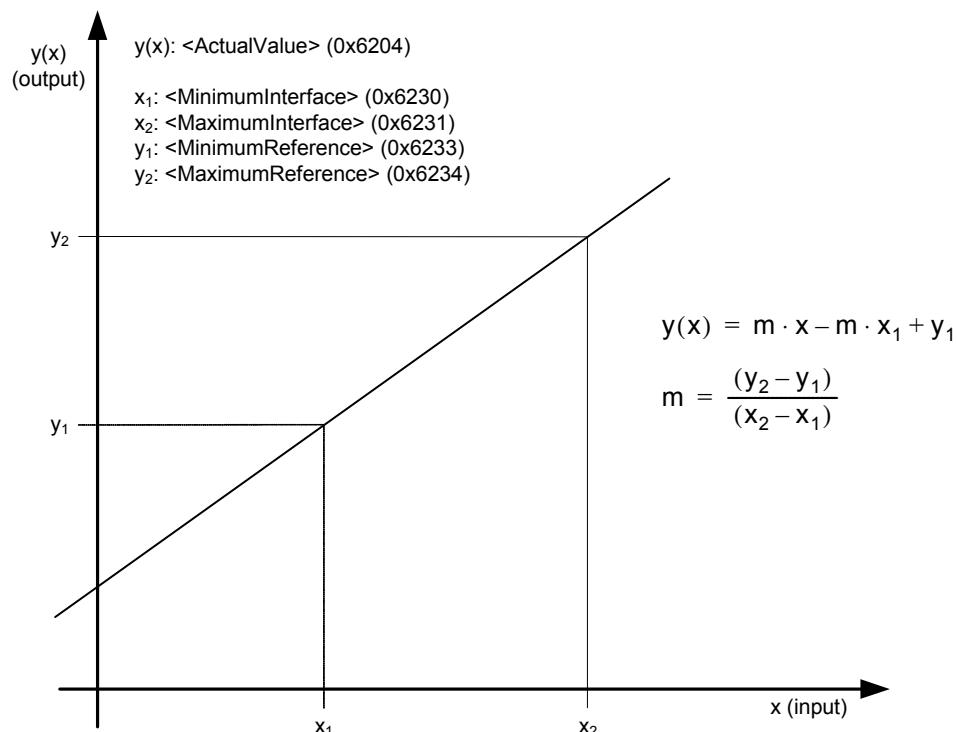
Figure 32: Position analog encoder actual value scaling

6.3.12.1 Object 0x6233: Minimum reference

This parameter defines the transducer signal when position is minimal.

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6233	1	MinimumReference	INT32	rw	N	INT32	−1

6.3.12.2 Object 0x6234: Maximum reference

This parameter defines the transducer signal when position is maximal.

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6234	1	MaximumReference	INT32	rw	N	INT32	1

6.3.12.3 Object 0x6230: Minimum interface

This parameter defines the minimum interface signal.

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6230	1	MinimumInterface	INT32	rw	N	INT32	−1

6.3.12.4 Object 0x6231: Maximum interface

This parameter defines the maximum interface signal.

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6231	1	MinimumInterface	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 trans-ducer with CAN interface is mapped to the receive PDO <Integer32> (0x0004). 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> (0x6202).

This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6201).

y(x): <ActualValue> (0x6204)

x1: <GeneralInputSignalMinimum> (0x622C)
x2: <GeneralInputSignalMaximum> (0x622D)
y1: <GeneralInputMinimum> (0x6228)
y2: <GeneralInputMaximum> (0x6229)
yoff: <GeneralInputOffset> (0x622B)

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

$$m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$
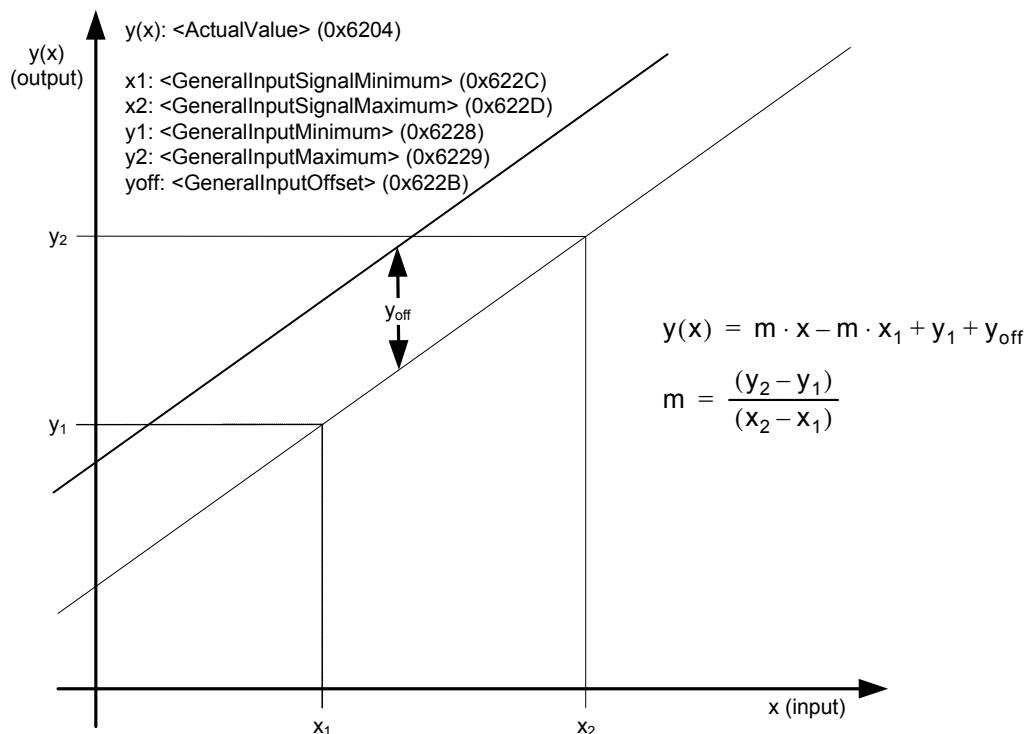
Figure 33: General input scaling

6.3.13.1 Object 0x6228: 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 (0x6228) = 0 % = 0.

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6228	1	GeneralInputMinimum	INT32	rw	N	INT32	0

6.3.13.2 Object 0x6229: 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 (0x6229) = 100 % = 16384.

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6229	1	GeneralInputMaximum	INT32	rw	N	INT32	16384

6.3.13.3 Object 0x622C: 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 (0x622C) = input voltage for 0 %.

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x622C	1	GeneralInputSignalMinimum	INT32	rw	N	INT32	0

6.3.13.4 Object 0x622D: 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 (0x622D) = input voltage for 100 %.

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x622D	1	GeneralInputSignalMaximum	INT32	rw	N	INT32	16384

6.3.13.5 Object 0x622B: 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							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x622B	1	GeneralInputOffset	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> (0x6201) shows the active transducer interface number. The output value of the active transducer interface can be read from the parameter <ActualValue> (0x6204).

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> (0x6201).

2. Enable or disable the selected transducer interface and select the method of conditioning.
   Therefore set the interface type:
   <Type> (0x6202) to 0 (no transducer function) or
   <Type> (0x6202) to 2 (pressure transducer) or
   <Type> (0x6202) to 5 (general input) or
   <Type> (0x6202) to 64 (position incremental) or
   <Type> (0x6202) to 65 (position SSI binary) or
   <Type> (0x6202) to 66 (position SSI gray) or
   <Type> (0x6202) to 67 (positon analog) or
   <Type> (0x6202) to –1 (analog direct with 32 bit) or
   <Type> (0x6202) 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> (0x6201).

2. Change the sign of the transducer interface.
   Therefore set the sign:
   <Sign> (0x6203) to 1 (positive) or
   <Sign> (0x6203) 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> (0x6201).

2. Define the input signal for the transducer interface with the input parameter address.
   Get the index, sub-index and parameter length in bits from the object dictionary or from the parameter description.
   For example, the parameter <dums16> (0x0003) 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	MSB	LSB		Parameter length in bit: 0x10	
Example	0x00	0x03	0x00	0x10	0x00030010

   Write the result 0x00030010 into the parameter <TransducerPort> (0x4032).

3. Check the transducer interfaces <Sign> (0x6203) and change the value (1 or –1) if needed.

4. Set the transducer interface <Type> (0x6202) 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> (0x6201).

2. Define the input signal for the transducer interface with the input parameter address.
   Get the index, sub-index and parameter length in bits from the object dictionary or from the parameter description.
   For example, the analog input 2 <ActualValue2> (0x6211) 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	MSB	LSB		Parameter bit length: 0x10	
Example	0x32	0x14	0x00	0x10	0x32140010

   Write the result 0x00030010 in the parameter <TransducerPort> (0x4032).

3. Check transducer interfaces <Sign> (0x6203) and change value (1 or –1) if needed.

4. Set transducer interface <Type> (0x6202) to 2 (pressure transducer).

5. Set the <PressureOffset> (0x6223) 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> (0x6224)
   x_2: <MaximumTransducerSignal> (0x6225)
   y_1: <MinimumPressure> (0x6220)
   y_2: <MaximumPressure> (0x6221)

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.
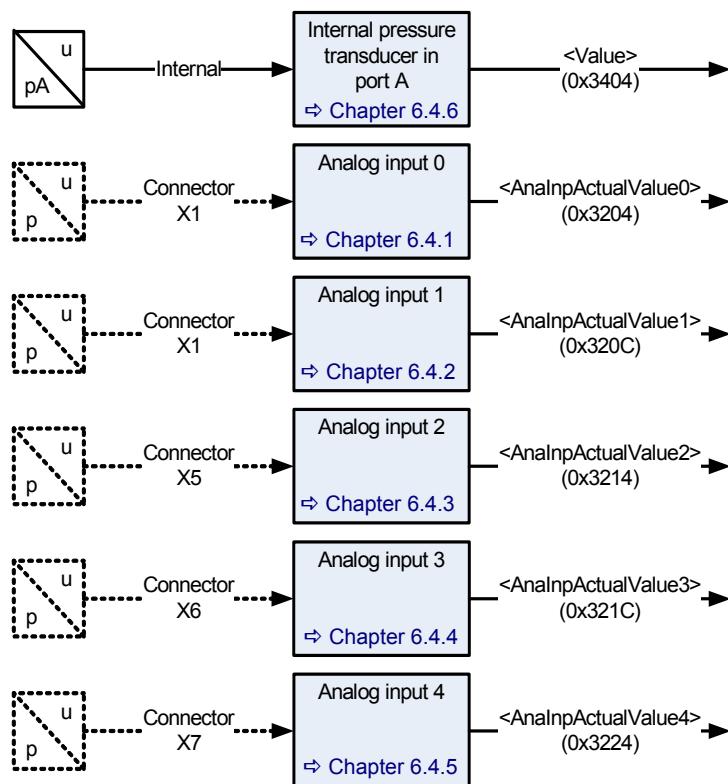
Figure 34: Analog inputs

⇨ 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> (0x6042) is set to 2 (setpoint input locally). The setpoint values are provided by the field bus, if the <DeviceMode> (0x6042) 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 59

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 74

ⓘ The analog inputs 0 and 1 are only effective as setpoint value inputs, if the <DeviceMode> (0x6042) is set to 2 (setpoint input locally).

6.4.1 Analog input 0

6.4.1.1 Object 0x3200: Input type

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

AnalogInput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3200	0	AnaInpType0	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 44: Possible values of parameter <AnaInpType0> (0x3200)

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

6.4.1.2 Object 0x3204: Actual value

Actual value of the analog input 0.

AnalogInput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3204	0	AnaInpActualValue0	INT16	ro	-	INT16	None

6.4.2 Analog input 1

6.4.2.1 Object 0x3208: Input type

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

AnalogInput1							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3208	0	AnaInpType1	INT8	rw	Y	INT8	2

Value description

6.4.2.2 Object 0x320C: Actual value

Actual value of the analog input 1.

AnalogInput1							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x320C	0	AnaInpActualValue1	INT16	ro	-	INT16	None

6.4.3 Analog input 2

6.4.3.1 Object 0x3210: Input type

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

AnalogInput2							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3210	0	AnaInpType2	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 45: Possible values of parameter <AnaInpType2> (0x3210)

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

6.4.3.2 Object 0x3214: Actual value

Actual value of the analog input 2.

AnalogInput2							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3214	0	AnaInpActualValue2	INT16	ro	-	INT16	None

Value description

6.4.4 Analog input 3

6.4.4.1 Object 0x3218: Input type

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

AnalogInput3							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3218	0	AnaInpType3	INT8	rw	Y	INT8	2

Value description

⇨ Table 45, page 88

6.4.4.2 Object 0x321C: Actual value

Actual value of the analog input 3.

AnalogInput3							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x321C	0	AnaInpActualValue3	INT16	ro	-	INT16	None

6.4.5 Analog input 4

6.4.5.1 Object 0x3220: Input type

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

AnalogInput4							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3220	0	AnaInpType4	INT8	rw	Y	INT8	2

Value description

⇨ Table 45, page 88

6.4.5.2 Object 0x3224: Actual value

Actual value of the analog input 4.

AnalogInput4							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3224	0	AnaInpActualValue4	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 74

6.4.6.1 Object 0x3404: Actual value

Actual value of the internal pressure transducer input.

PressureTransducer							
Index	Sub-index	Parameter name	Data type	Access	Persis- tence	Value range	Default
0x3404	0	AnaInpActualValue4	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.
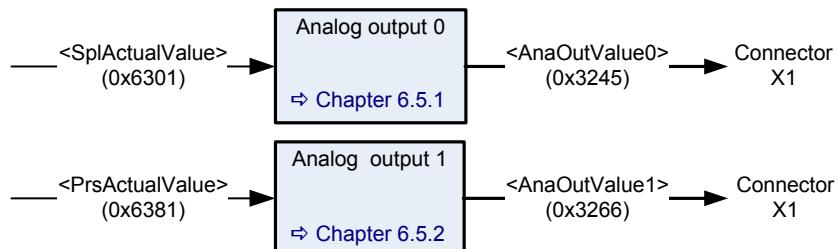
Figure 35: Analog outputs

⇨ Document CA63420-001 "User Manual Electrical Interfaces"

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

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

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

num: <Numerator>
den: <Denominator>
y_{off}: <Offset>
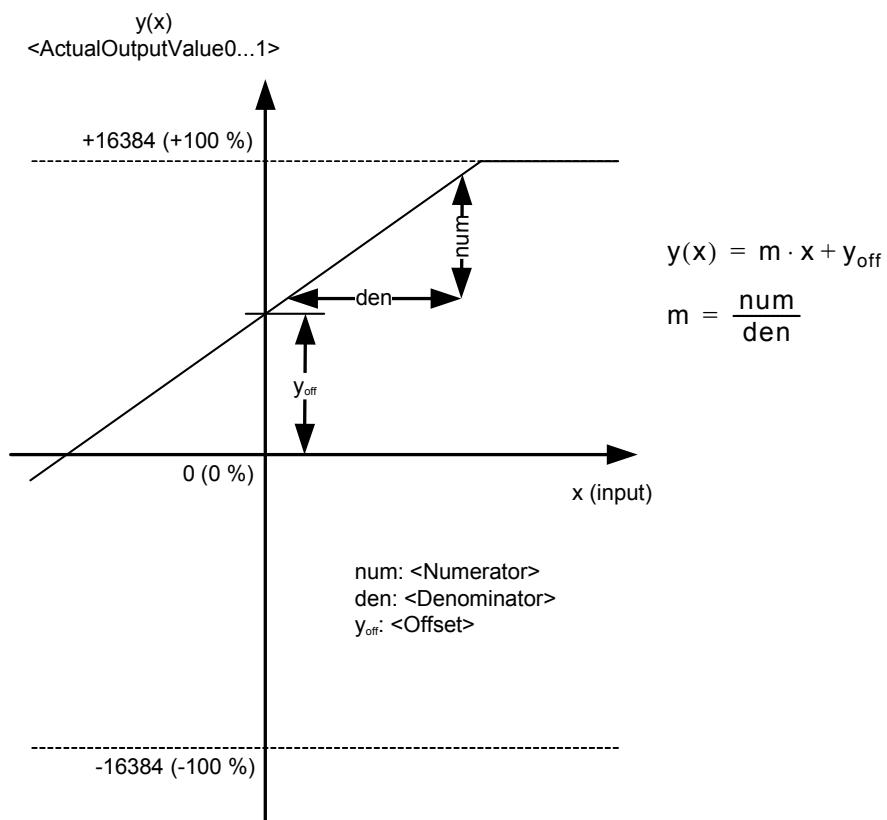
Figure 36: Analog output scaling

6.5.1 Analog output 0

6.5.1.1 Object 0x3244: Scaling

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3244	1	AnaOutScaNumerator0	INT16	rw	Y	INT16	16384
0x3244	2	AnaOutScaDenominator0	INT16	rw	Y	INT16	16384
0x3244	3	AnaOutScaOffset0	INT16	rw	Y	INT16	0

6.5.1.2 Object 0x3245: Actual value

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3245	0	AnaOutValue0	INT16	ro	-	Depending on <AnaOutType0> (0x3243)	None

6.5.1.3 Object 0x3240: Mapping parameter

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

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3240	0	AnaOutMappingParameter0	UINT32	rw	Y	UINT32	0x63010110

Value description

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

Table 46: Possible values of parameter <AnaOutMappingParameter0> (0x3240)

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> (0x6301), sub-index 0x01 with a length of 16 bit (16 = 0x10).

6.5.1.4 Object 0x3243: Type

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3243	0	AnaOutType0	UINT8	rw	-	0…1	0

Value description

<Type>	<AnaOutValue0> (0x3245) 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 47: Possible values of parameter <AnaOutType0> (0x3243)

6.5.2 Analog output 1

6.5.2.1 Object 0x3265: Scaling

AnalogOutput1							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3265	1	AnaOutScaNumerator1	INT16	rw	Y	INT16	16384
0x3265	2	AnaOutScaDenominator1	INT16	rw	Y	INT16	16384
0x3265	3	AnaOutScaOffset1	INT16	rw	Y	INT16	0

6.5.2.2 Object 0x3266: Actual value

AnalogOutput1							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3266	0	AnaOutValue1	INT16	ro	-	Depending on <AnaOutType0> (0x3263)	None

6.5.2.3 Object 0x3260: Mapping parameter

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

AnalogOutput1							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3260	0	AnaOutMappingParameter1	UINT32	rw	Y	UINT32	0x63810110

Value description

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

Table 48: Possible values of parameter <AnaOutMappingParameter1> (0x3260)

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> (0x6381), sub-index 0x01 with a length of 16 bit (16=0x10).

6.5.2.4 Object 0x3263: Type

AnalogOutput1							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3263	0	AnaOutType0	UINT8	rw	-	0…1	0

Value description

<Type>	<AnaOutValue1> (0x3266) 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 49: Possible values of parameter <AnaOutType0> (0x3263)

6.6 Encoder input

The servo valve has one encoder input.
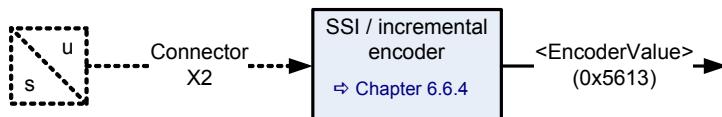
Figure 37: Encoder input

⇨ 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> (0x6202) will select the type of the encoder (SSI binary, SSI gray or incremental)
It is possible to map the output of the encoder <EncoderValue> (0x5613) to more than one <TransducerPort> (0x4032) . In this case the first of this transducer ports is used to initialize the encoder hardware.

6.6.1 Object 0x5613: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5613	0	EncoderValue	UINT32	rw	N	UINT32	None

6.6.2 Object 0x5621: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5621	0	EncoderSetValue	INT32	rw	Y	INT32	0

6.6.3 Object 0x5619: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5619	0	SensorSupplyEnable	UINT8	rw	Y	0, 5, 24	0

Value description

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

Table 50: Possible values of parameter <SensorSupplyEnable> (0x5619)

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> (0x6042) to 4 (reference mode) and set bit 9 (axis install mode positive direction) or bit 10 (axis install mode negative direction) of the <ControlWord> (0x6040).

In the first step of the reference run, axis will move with the <ReferencingVelocity> (0x5614) 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> (0x6381) reaches the <ReferencingForce> (0x561A) 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> (0x5617) = 1, the valve is waiting for the Z pulse. If the Z pulse appears or the axis moved the distance <ReferencingStop> (0x561B), the reference run is finished and the <EncoderValue> (0x5613) is set to 0.

A manual search for the reference position is also possible if the <DeviceMode> (0x6042) is switched to 1 or 2 (setpoint input via bus or locally). If the flag <ZPulseEnable> (0x5610) is set, the valve is waiting for the next Z pulse. When the Z pulse appears, the predefined position <ZPulseSet> (0x5612) is copied to the <EncoderValue> (0x5613) and <ZPulseEnable> (0x5610) is reset to indicate the successful referencing.

6.6.4.1 Object 0x5614: Referencing velocity

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

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5614	0	ReferencingVelocity	INT32	rw	Y	1000…INT32	1000

6.6.4.2 Object 0x561A: 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> (0x6381) reaches the <ReferencingForce> (0x561A) to finish the first step.

⇨ Chapter "7.5.1 Object 0x6381: Actual value", page 168

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x561A	0	ReferencingForce	INT16	rw	Y	INT16	0

6.6.4.3 Object 0x561B: Referencing stop

In the second step of the reference run, the axis is moving slowly back from the "end of stroke position". If <ZPulseTrigger> (0x5617) = 1, the valve is waiting for the Z pulse. If the Z pulse appears or the axis has moved the distance <ReferencingStop> (0x561B), the reference run is finished.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x561B	0	ReferencingStop	FLOAT32	rw	Y	FLOAT32	0.0

6.6.4.4 Object 0x5617: Z pulse trigger

In the second step of the reference run, the parameter <ZPulseTrigger> (0x5617) will enable the Z pulse to trigger the reference position of the encoder. If <ZPulseTrigger> (0x5617) = 0, the Z pulse is ignored and will not be dedected.

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5617	0	ZPulseTrigger	UINT8	rw	N	0…1	0

6.6.4.5 Object 0x5611: 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> (0x6042) is switched to 4 (reference mode).

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5611	0	ZPulseDetected	UINT8	rw	N	0…1	0

6.6.4.6 Object 0x5618: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5618	0	ZPulseClear	UINT8	rw	N	0…1	0

6.6.4.7 Object 0x5610: 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> (0x5612) is copied to the <EncoderValue> (0x5613) and <ZPulseEnable> (0x5610) is reset to indicate the successful referencing. Only if the <DeviceMode> (0x6042) is switched to 1 or 2 (setpoint input via bus or locally).

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5610	0	ZPulseEnable	UINT8	rw	N	0…1	0

6.6.4.8 Object 0x5612: Z pulse set

This is a predefined position. It will be copied to the <EncoderValue> (0x5613), if a Z pulse appears while <ZPulseEnable> (0x5610) is set. Only if the <DeviceMode> (0x6042) is switched to 1 or 2 (setpoint input via bus or locally).

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5612	0	ZPulseSet	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> (0x6243) and <BitRate> (0x561F).

6.6.5.1 Object 0x5620: Master slave

Only the SSI master interface is supported.

Drive_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5620	0	MasterSlave	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 51: Possible values of parameter <MasterSlave> (0x5620)

6.6.5.2 Object 0x6243: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6243	0	BitSize	UINT8	rw	N	0…32	24

6.6.5.3 Object 0x561F: Bit rate

The <BitRate> (0x561F) 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x561F	0	BitRate	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 52: Possible values of parameter <BitRate> (0x561F)

6.6.5.4 Object 0x3252: SSI Error Count

The <SSIErrorCount> (0x3252) 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3252	0	SSIErrorCount	UINT8	rw	Y	0…254	3

6.7 Digital inputs

The servo valve has one digital input.
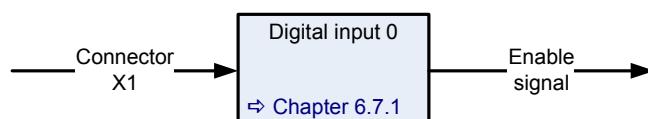
Figure 38: Digital inputs

⇨ 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 51
- Fault confirmation by toggling the digital enable signal.
  ⇨ Chapter "5.2.2.4.2 Fault confirmation with the enable signal", page 51

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.

(i)   Changes are possible. Please refer to the specific pin configuration of your valve.
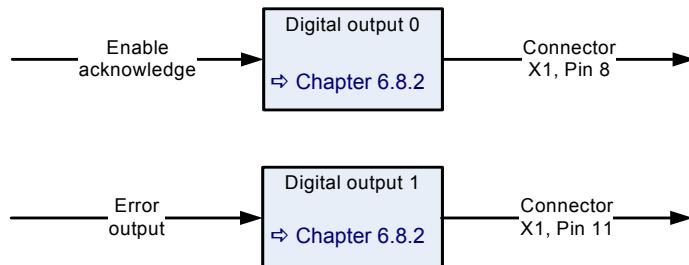
Figure 39: Digital outputs in the default configuration

⇨ Document CA63420-001 "User Manual Electrical Interfaces"

6.8.1 Object 0x5E42: Digital output setpoint

The <DigitalOutputSetpoint> (0x5E42) controls the state of the digital outputs in case the <DigitalOutputConfiguration> (0x5E41) is set to 0.

ValveDigitalOutputValue							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5E42	1	DigitalOutputSetpoint0	INT8	rw	-	0…1	0
0x5E42	2	DigitalOutputSetpoint1	INT8	rw	-	0…1	0

6.8.2 Object 0x5E41: Digital output configuration

The behavior of the digital outputs can be configured by the parameter <DigitalOutputConfiguration> (0x5E41).

ValveDigitalOutputType							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5E41	1	DigitalOutputConfiguration0	INT8	ro	-	0…4	3
0x5E41	2	DigitalOutputConfiguration1	INT8	ro	-	0…4	0

Value description

<DigitalOutputType>	Description
0	The servo valve's digital outputs are controlled by the parameter <DigitalOutputSetpoint> (0x5E42). 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 241
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 52
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> (0x6041) is low. That means there is no control error. ⇨ Chapter "7.15 Monitoring", page 232

Table 53: Possible values of parameter <DigitalOutputConfiguration> (0x5E41)

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 0x2420: Digital output 1 type

The <DigitalOutput1Type> (0x2420) is a copy of the <DigitalOutputConfiguration1> (0x5E41) and is used to be compatible to old firmware versions.

ValveDigitalOutputType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x2420	0	DigitalOutput1Type	INT8	ro	-	0…4	0

6.8.3 Object 0x5E44: Digital output value

The <DigitalOutputValue> (0x5E44) shows the state of the digital outputs.

ValveDigitalOutputMonitor							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5E44	1	DigitalOutputValue_0	UINT8	ro	-	0…1	None
0x5E44	2	DigitalOutputValue_1	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 105
- 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 119
- Network management (NMT) protocol
  The NMT protocol command "start remote node" is supported.
  ⇨ Chapter "6.9.1.3 Object 0x5B02: Start remote node", page 104
- 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 121

ⓘ 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.
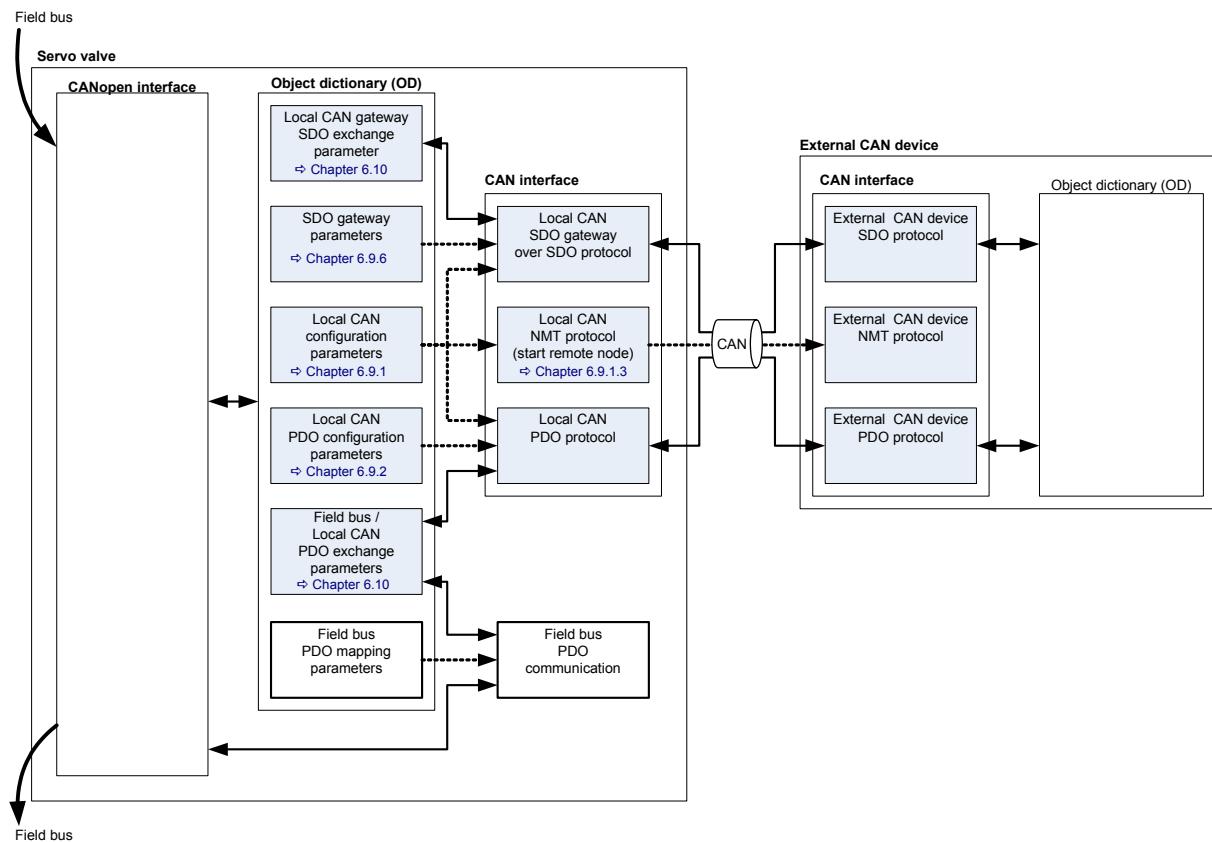
Figure 40: Structure of local CAN bus

6.9.1 Local CAN general configuration

6.9.1.1 Object 0x5B00: 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							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5B00	0	LocalCANModuleIdentifier	UINT8	rw	Y	1…127	127

6.9.1.2 Object 0x5B01: Bit rate

The local CAN bit rate in bits per second.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5B01	0	LocalCANBitrate	UINT32	rw	Y	0…1000000	500000

6.9.1.3 Object 0x5B02: 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> (0x5B02). 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							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5B02	0	LocalCANStartRemoteNode	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 0x5B03: 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> (0x5B03) value must be set to the PDO number which is to be sent.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5B03	0	LocalCANTPdoTrigger	UINT8	rw	N	0…4	None

6.9.1.5 Object 0x5B14: 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> (0x5B14) to 1.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5B14	0	LocalCANTerminationResistor	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 106
- Transmit process data object (TxPDO) protocol
  ⇨ Chapter "6.9.4 Transmit process data object (TxPDO) configuration", page 112
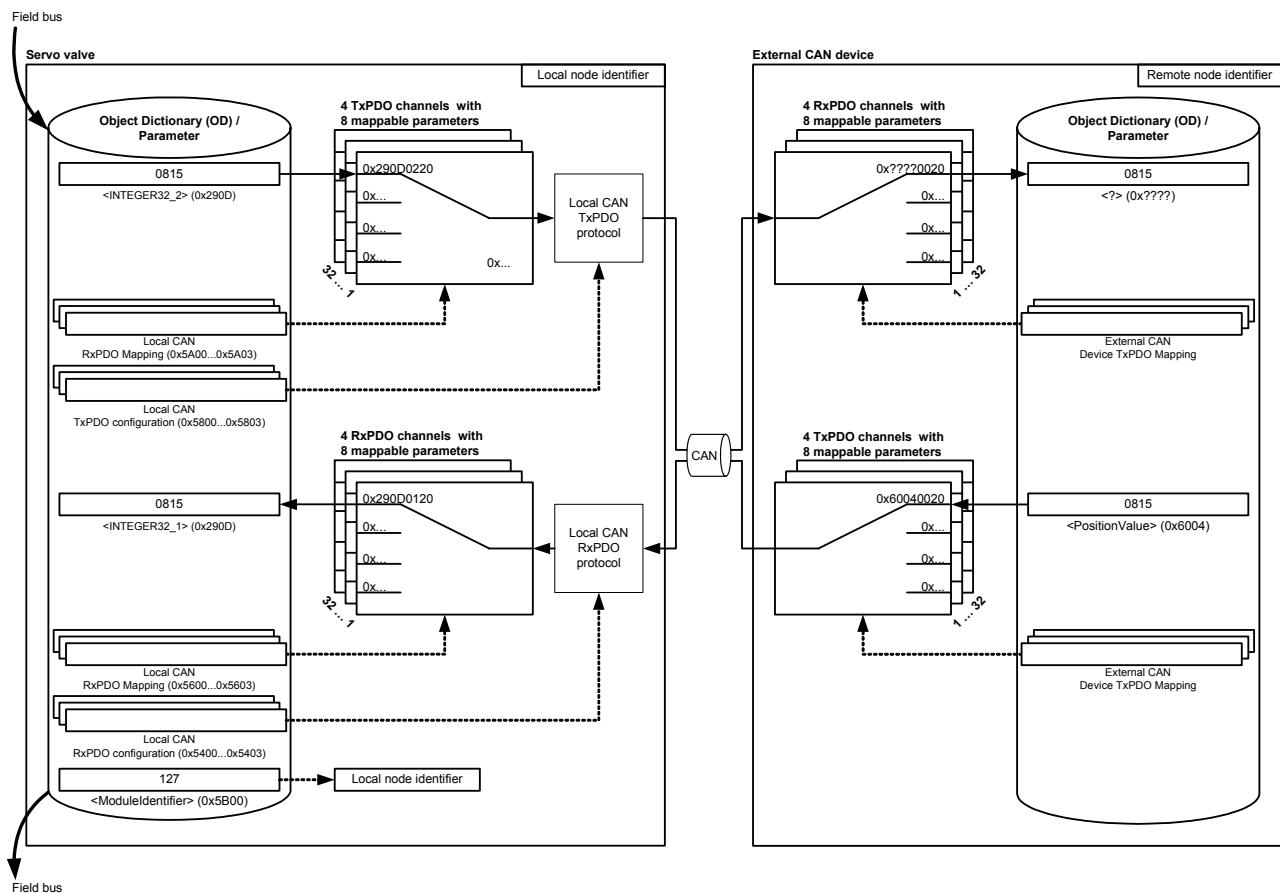
Figure 41: Local CAN process data object (PDO) configuration

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 mappings of the four possible local CAN receive RxPDOs is the same as the mapping of the field bus RxPDO.
⇨ Chapter "2.8.3.2 RxPDO mapping (COB-ID: 0x200, 0x300, 0x400, 0x500)", page 21

The parameters described in the ⇨ Chapter "6.10 Free to use parameters", page 122 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> (0x5400…0x5403) 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 54: 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 0x5B12: Remote node identifier", page 121

6.9.3.1 Object 0x5400: 1st RxPDO configuration

First receive PDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5400	1	LocalRPdo1_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0200+127
0x5400	2	LocalRPdo1_TransmissionType	UINT8	rw	Y	UINT8	255
0x5400	3	LocalRPdo1_EventTimer	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 55: Parameters of 1st RxPDO configuration object (0x5400)

<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 56: Possible values of parameter <LocalRPdo1_CobIdUsedByPdo> (0x5400)

<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 57: Possible values of parameter <LocalRPdo1_TransmissionType> (0x5400)

6.9.3.2 Object 0x5401: 2nd RxPDO configuration

Second receive PDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5401	1	LocalRPdo2_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0300+127
0x5401	2	LocalRPdo2_TransmissionType	UINT8	rw	Y	UINT8	255
0x5401	3	LocalRPdo2_EventTimer	UINT16	rw	Y	UINT16	0

Value description

6.9.3.3 Object 0x5402: 3rd RxPDO configuration

Third receive PDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5402	1	LocalRPdo3_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0400+127
0x5402	2	LocalRPdo3_TransmissionType	UINT8	rw	Y	UINT8	255
0x5402	3	LocalRPdo3_EventTimer	UINT16	rw	Y	UINT16	0

Value description

⇨ Chapter "6.9.3.1 Object 0x5400: 1st RxPDO configuration", page 107

6.9.3.4 Object 0x5403: 4th RxPDO configuration

Fourth receive PDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5403	1	LocalRPdo4_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0500+127
0x5403	2	LocalRPdo4_TransmissionType	UINT8	rw	Y	UINT8	255
0x5403	3	LocalRPdo4_EventTimer	UINT16	rw	Y	UINT16	0

Value description

⇨ Chapter "6.9.3.1 Object 0x5400: 1st RxPDO configuration", page 107

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> (0x6040)
  ⇨ Chapter "5.2 Device state machine (DSM)", page 46

- Spool position setpoint value <SplSetpoint> (0x6300)
  ⇨ Chapter "6.2.3 Spool position setpoint value path", page 59

- Pressure setpoint value <PrsSetpoint> (0x6380)
  ⇨ Chapter "6.2.4 Pressure setpoint value path", page 61
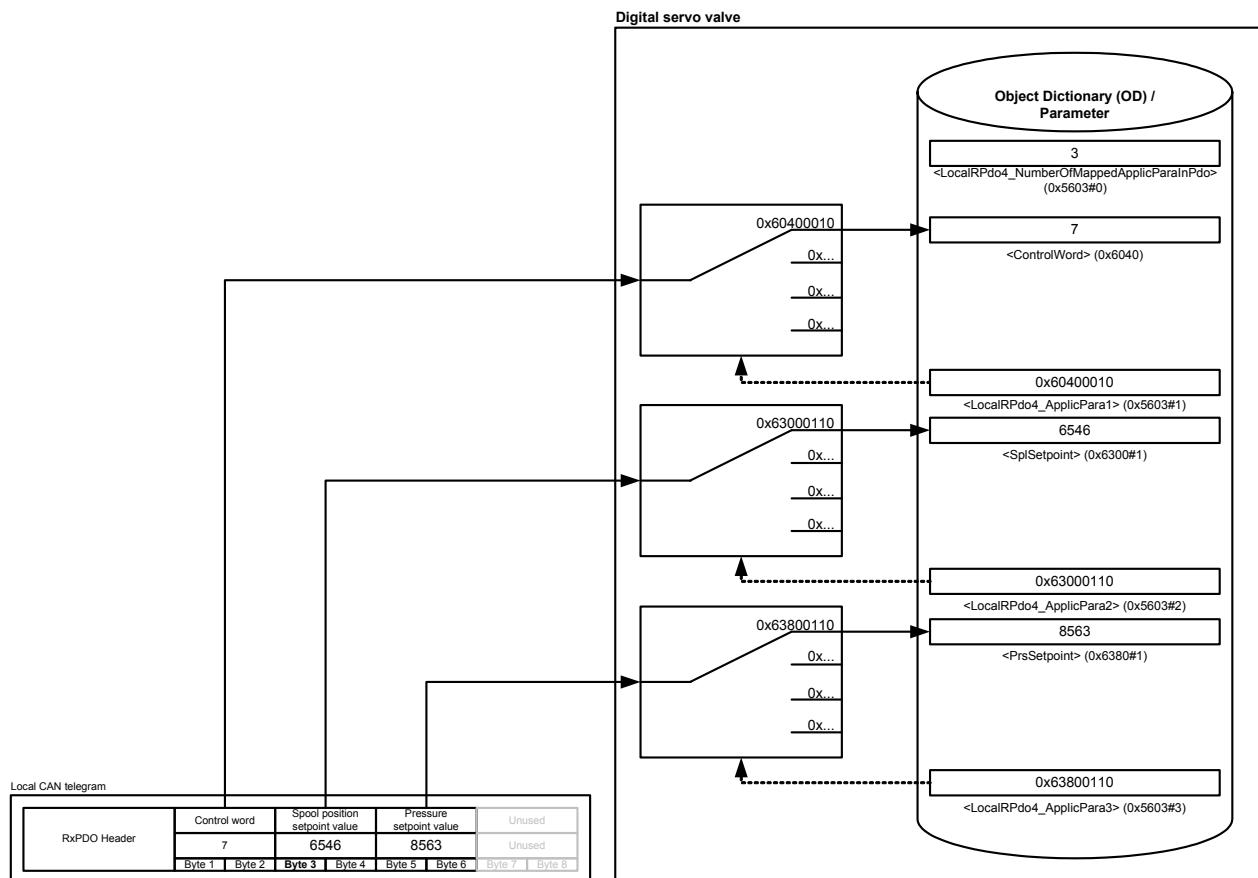
Figure 42: Receive process data object (RxPDO) mapping

The parameter <LocalRPdo4_NumberOfMappedApplicParaInPdo> (0x5603#0) defines the number of mapped values for the fourth RxPDO. The parameter <LocalRPdo1_ApplicPara1> (0x5603#1) references to the <ControlWord> (0x6040#0). The references to the parameters <SplSetpoint> (0x6300#1) and <PrsSetpoint> (0x6380#1) are defined in the same manner.

6.9.3.6 Object 0x5600: 1st RxPDO mapping

With the parameter <LocalRPdo1_NumberOfMappedApplicParaInPdo> (0x5600) 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 positions (corresponding to the sub-indexes 1…8) within the PDO object.

⇨ Chapter "2.8.3.2.1 Object 0x1600: 1st RxPDO mapping", page 22

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5600	0	LocalRPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5600	1	LocalRPdo1_ApplicPara1	UINT32	rw	Y	UINT32	0
0x5600	2	LocalRPdo1_ApplicPara2	UINT32	rw	Y	UINT32	0
0x5600	3	LocalRPdo1_ApplicPara3	UINT32	rw	Y	UINT32	0
0x5600	4	LocalRPdo1_ApplicPara4	UINT32	rw	Y	UINT32	0
0x5600	5	LocalRPdo1_ApplicPara5	UINT32	rw	Y	UINT32	0
0x5600	6	LocalRPdo1_ApplicPara6	UINT32	rw	Y	UINT32	0
0x5600	7	LocalRPdo1_ApplicPara7	UINT32	rw	Y	UINT32	0
0x5600	8	LocalRPdo1_ApplicPara8	UINT32	rw	Y	UINT32	0

Value description

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

Table 58: Parameters of 1st RxPDO mapping object (0x5600)

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

Table 59: Value description of mapping parameter <LocalRPdo1_ApplicPara1…8>

6.9.3.7 Object 0x5601: 2nd RxPDO mapping

Second receive PDO mapping.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5601	0	LocalRPdo2_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5601	1	LocalRPdo2_ApplicPara1	UINT32	rw	Y	UINT32	0
0x5601	2	LocalRPdo2_ApplicPara2	UINT32	rw	Y	UINT32	0
0x5601	3	LocalRPdo2_ApplicPara3	UINT32	rw	Y	UINT32	0
0x5601	4	LocalRPdo2_ApplicPara4	UINT32	rw	Y	UINT32	0
0x5601	5	LocalRPdo2_ApplicPara5	UINT32	rw	Y	UINT32	0
0x5601	6	LocalRPdo2_ApplicPara6	UINT32	rw	Y	UINT32	0
0x5601	7	LocalRPdo2_ApplicPara7	UINT32	rw	Y	UINT32	0
0x5601	8	LocalRPdo2_ApplicPara8	UINT32	rw	Y	UINT32	0

Value description

6.9.3.8 Object 0x5602: 3rd RxPDO mapping

Third receive PDO mapping.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5602	0	LocalRPdo3_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5602	1	LocalRPdo3_ApplicPara1	UINT32	rw	Y	UINT32	0
0x5602	2	LocalRPdo3_ApplicPara2	UINT32	rw	Y	UINT32	0
0x5602	3	LocalRPdo3_ApplicPara3	UINT32	rw	Y	UINT32	0
0x5602	4	LocalRPdo3_ApplicPara4	UINT32	rw	Y	UINT32	0
0x5602	5	LocalRPdo3_ApplicPara5	UINT32	rw	Y	UINT32	0
0x5602	6	LocalRPdo3_ApplicPara6	UINT32	rw	Y	UINT32	0
0x5602	7	LocalRPdo3_ApplicPara7	UINT32	rw	Y	UINT32	0
0x5602	8	LocalRPdo3_ApplicPara8	UINT32	rw	Y	UINT32	0

Value description

6.9.3.9 Object 0x5603: 4th RxPDO mapping

Fourth receive PDO mapping.

Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
LocalCAN							
0x5603	0	LocalRPdo4_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5603	1	LocalRPdo4_ApplicPara1	UINT32	rw	Y	UINT32	0
0x5603	2	LocalRPdo4_ApplicPara2	UINT32	rw	Y	UINT32	0
0x5603	3	LocalRPdo4_ApplicPara3	UINT32	rw	Y	UINT32	0
0x5603	4	LocalRPdo4_ApplicPara4	UINT32	rw	Y	UINT32	0
0x5603	5	LocalRPdo4_ApplicPara5	UINT32	rw	Y	UINT32	0
0x5603	6	LocalRPdo4_ApplicPara6	UINT32	rw	Y	UINT32	0
0x5603	7	LocalRPdo4_ApplicPara7	UINT32	rw	Y	UINT32	0
0x5603	8	LocalRPdo4_ApplicPara8	UINT32	rw	Y	UINT32	0

Value description

⇨ Chapter "6.9.3.6 Object 0x5600: 1st RxPDO mapping", page 110

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 mappings of the four possible local CAN transmit TxPDO's is the same as the mapping of the field bus TxPDO.
⇨ Chapter "2.8.3.5.1 Object 0x1A00: 1st TxPDO mapping", page 28

The parameters described in the ⇨ Chapter "6.10 Free to use parameters", page 122 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> (0x5800…0x5803) 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 60: 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 0x5B12: Remote node identifier", page 121

6.9.4.1 Object 0x5800: 1st TxPDO configuration

First transmit PDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5800	1	LocalTPdo1_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0180+127
0x5800	2	LocalTPdo1_TransmissionType	UINT8	rw	Y	UINT8	255
0x5800	5	LocalTPdo1_EventTimer	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 61: Parameters of 1st TxPDO configuration object (0x5800)

<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 62: Possible values of parameter <LocalTPdo1_CobIdUsedByPdo> (0x5800)

<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 63: Possible values of parameter <LocalTPdo1_TrasnmissionType> (0x5800)

ⓘ To initiate only a single request, the <LocalTPdo1_EventTimer> (0x5800) must be set to 0. The transmission then can be triggered by the parameter <LocalCANTPdoTrigger> (0x5B03).

6.9.4.2 Object 0x5801: 2nd TxPDO configuration

Second transmit PDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5801	1	LocalTPdo2_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0280+127
0x5801	2	LocalTPdo2_TransmissionType	UINT8	rw	Y	UINT8	255
0x5801	3	LocalTPdo2_EventTimer	UINT16	rw	Y	UINT16	0

Value description

⇨ Chapter "6.9.4.1 Object 0x5800: 1st TxPDO configuration", page 113

6.9.4.3 Object 0x5802: 3rd TxPDO configuration

Third transmit PDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5802	1	LocalTPdo3_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0380+127
0x5802	2	LocalTPdo3_TransmissionType	UINT8	rw	Y	UINT8	255
0x5802	3	LocalTPdo3_EventTimer	UINT16	rw	Y	UINT16	0

Value description

⇨ Chapter "6.9.4.1 Object 0x5800: 1st TxPDO configuration", page 113

6.9.4.4 Object 0x5803: 4th TxPDO configuration

Fourth transmit PDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5803	1	LocalTPdo4_CobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF | 0x80000000	0x0480+127
0x5803	2	LocalTPdo4_TransmissionType	UINT8	rw	Y	UINT8	255
0x5803	3	LocalTPdo4_EventTimer	UINT16	rw	Y	UINT16	0

Value description

⇨ Chapter "6.9.4.1 Object 0x5800: 1st TxPDO configuration", page 113

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> (0x6041)
  ⇨ Chapter "5.2 Device state machine (DSM)", page 46

- Spool position actual value <SplActualValue> (0x6301)
  ⇨ Chapter "7.3 Spool position controller", page 153

- Pressure actual value <PrsActualValue> (0x6381)
  ⇨ Chapter "7.5.11 Actual value filter", page 174
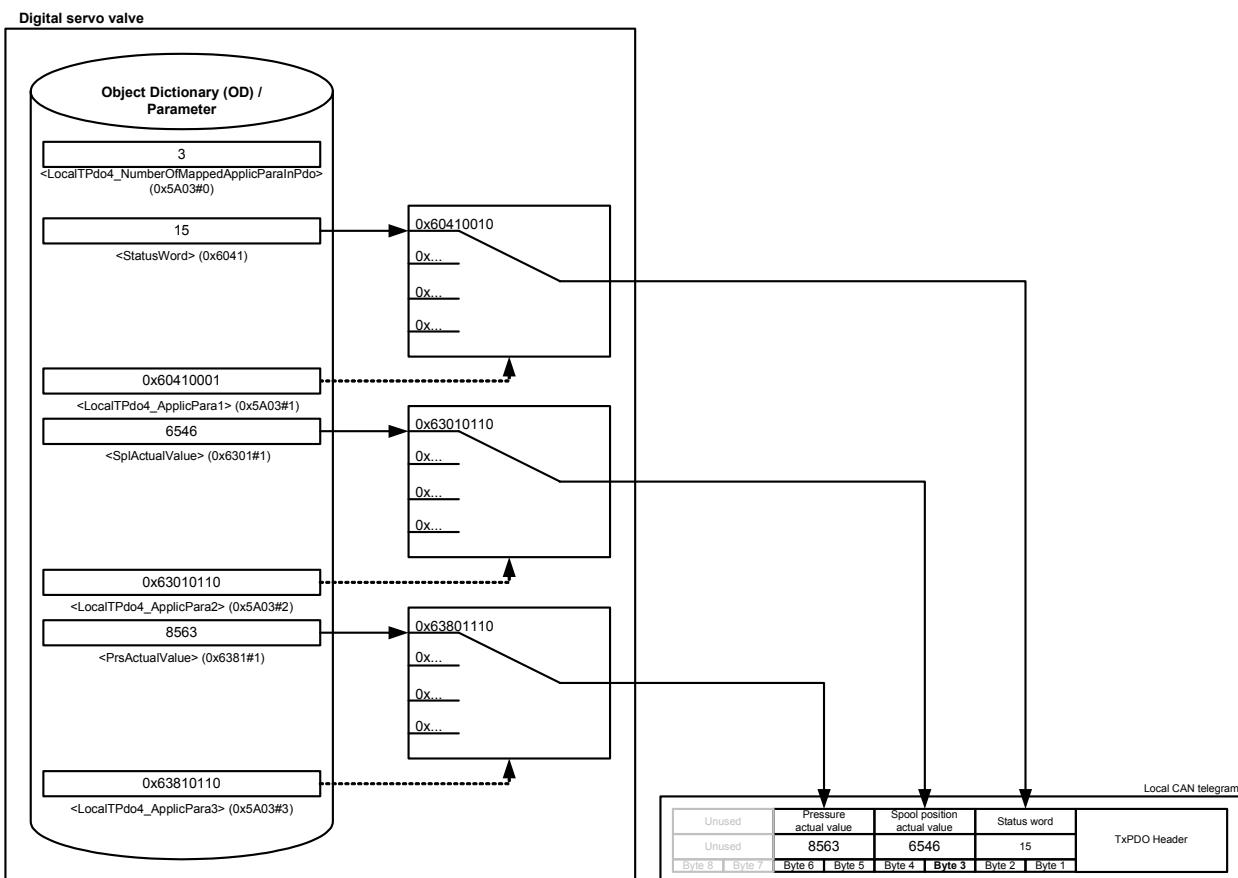
Figure 43: Transmit process data object (TxPDO) mapping

The parameter <LocalTPdo4_NumberOfMappedApplicParaInPdo> (0x5A03#0) defines the number of mapped values for the fourth TxPDO. The parameter <LocalTPdo4_ApplicPara1> (0x5A03#1) references to the <StatusWord> (0x6041#0). The references to the parameters <SplActualValue> (0x6301#1) and <PrsActualValue> (0x6381#1) are defined in the same manner.

6.9.4.6 Object 0x5A00: 1st TxPDO mapping

First transmit PDO mapping.

With the parameter <LocalTPdo1_NumberOfMappedApplicParaInPdo> (0x5600) 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 positions (corresponding to the sub-indexes 1…8) within the PDO object.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5A00	0	LocalTPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5A00	1	LocalTPdo1_ApplicPara1	UINT32	rw	Y	UINT32	0
0x5A00	2	LocalTPdo1_ApplicPara2	UINT32	rw	Y	UINT32	0
0x5A00	3	LocalTPdo1_ApplicPara3	UINT32	rw	Y	UINT32	0
0x5A00	4	LocalTPdo1_ApplicPara4	UINT32	rw	Y	UINT32	0
0x5A00	5	LocalTPdo1_ApplicPara5	UINT32	rw	Y	UINT32	0
0x5A00	6	LocalTPdo1_ApplicPara6	UINT32	rw	Y	UINT32	0
0x5A00	7	LocalTPdo1_ApplicPara7	UINT32	rw	Y	UINT32	0
0x5A00	8	LocalTPdo1_ApplicPara8	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 64: Parameters of 1st TxPDO mapping object (0x5A00)

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

Table 65: Possible values of parameter <LocalTPdo1_ApplicPara1…8>

6.9.4.7 Object 0x5A01: 2nd TxPDO mapping

Second transmit PDO mapping.

Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5A01	0	LocalTPdo2_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5A01	1	LocalTPdo2_ApplicPara1	UINT32	rw	Y	UINT32	0
0x5A01	2	LocalTPdo2_ApplicPara2	UINT32	rw	Y	UINT32	0
0x5A01	3	LocalTPdo2_ApplicPara3	UINT32	rw	Y	UINT32	0
0x5A01	4	LocalTPdo2_ApplicPara4	UINT32	rw	Y	UINT32	0
0x5A01	5	LocalTPdo2_ApplicPara5	UINT32	rw	Y	UINT32	0
0x5A01	6	LocalTPdo2_ApplicPara6	UINT32	rw	Y	UINT32	0
0x5A01	7	LocalTPdo2_ApplicPara7	UINT32	rw	Y	UINT32	0
0x5A01	8	LocalTPdo2_ApplicPara8	UINT32	rw	Y	UINT32	0

Value description

⇨ Chapter "6.9.4.6 Object 0x5A00: 1st TxPDO mapping", page 116

6.9.4.8 Object 0x5A02: 3rd TxPDO mapping

Third transmit PDO mapping.

Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5A02	0	LocalTPdo3_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5A02	1	LocalTPdo3_ApplicPara1	UINT32	rw	Y	UINT32	0
0x5A02	2	LocalTPdo3_ApplicPara2	UINT32	rw	Y	UINT32	0
0x5A02	3	LocalTPdo3_ApplicPara3	UINT32	rw	Y	UINT32	0
0x5A02	4	LocalTPdo3_ApplicPara4	UINT32	rw	Y	UINT32	0
0x5A02	5	LocalTPdo3_ApplicPara5	UINT32	rw	Y	UINT32	0
0x5A02	6	LocalTPdo3_ApplicPara6	UINT32	rw	Y	UINT32	0
0x5A02	7	LocalTPdo3_ApplicPara7	UINT32	rw	Y	UINT32	0
0x5A02	8	LocalTPdo3_ApplicPara8	UINT32	rw	Y	UINT32	0

Value description

⇨ Chapter "6.9.4.6 Object 0x5A00: 1st TxPDO mapping", page 116

6.9.4.9 Object 0x5A03: 4th TxPDO mapping

Fourth transmit PDO mapping.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5A03	0	LocalTPdo4_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5A03	1	LocalTPdo4_ApplicPara1	UINT32	rw	Y	UINT32	0
0x5A03	2	LocalTPdo4_ApplicPara2	UINT32	rw	Y	UINT32	0
0x5A03	3	LocalTPdo4_ApplicPara3	UINT32	rw	Y	UINT32	0
0x5A03	4	LocalTPdo4_ApplicPara4	UINT32	rw	Y	UINT32	0
0x5A03	5	LocalTPdo4_ApplicPara5	UINT32	rw	Y	UINT32	0
0x5A03	6	LocalTPdo4_ApplicPara6	UINT32	rw	Y	UINT32	0
0x5A03	7	LocalTPdo4_ApplicPara7	UINT32	rw	Y	UINT32	0
0x5A03	8	LocalTPdo4_ApplicPara8	UINT32	rw	Y	UINT32	0

Value description

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:

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> (0x5B12)	Transmit service data object (TxSDO)	CiA 301
0x600	0x600 + <LocalCANRemoteNodeId> (0x5B12)	Receive service data object (RxSDO)	CiA 301

For the SDO transmission a timeout of 500 ms is set.
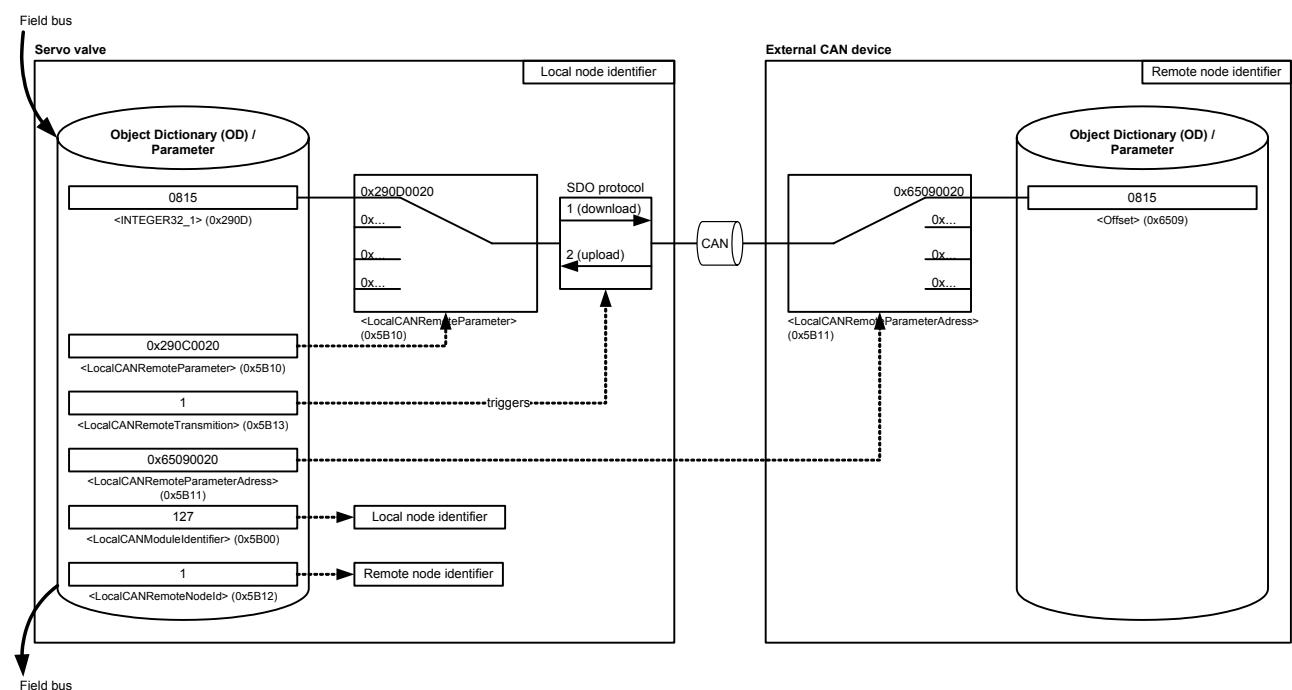
Figure 44: Local CAN service data object (SDO) gateway

6.9.6.1 Object 0x5B10: Remote parameter

This parameter defines the 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 122

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5B10	0	LocalCANRemoteParameter	UINT32	rw	N	UINT32	None

Value description

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

Table 66: Possible values of parameter <LocalCANRemoteParameter> (0x5B10)

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 0x5B11: Remote parameter address

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

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

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5B11	0	LocalCANRemoteParameterAdress	UINT32	rw	N	UINT32	None

Value description

<LocalCANRemoteParameterAdress>			
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length: 0x10
Default	0x00	0x00	0x00	0x00

Table 67: Possible values of parameter <LocalCANRemoteParameterAdress> (0x5B11)

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 0x5B12: Remote node identifier

This parameter represents the node identifier of the external device.

This parameter must be different from the node identifier <LocalCANModuleIdentifier> (0x5B00) of the local CAN bus of the servo valve.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5B12	0	LocalCANRemoteNodeId	UINT8	rw	N	0…127	None

6.9.6.4 Object 0x5B13: Remote transmission

Writing to the <LocalCANRemoteTransmission> (0x5B13) parameter initiates a read or write SDO transmission. The address of the considered remote parameter will be defined with the parameter <LocalCANRemoteParameterAdress> (0x5B11). If the <LocalCANRemoteTransmission> is set to 1 (download) the value of the address <LocalCANRemoteParameterAdress> (0x5B11) will be read and saved in the local parameter defined by the <LocalCANRemoteParameter> (0x5B107). If <LocalCANRemoteTransmission> (0x5B13) is set to 2 (upload), the local value at the address <LocalCANRemoteParameter> (0x5B10) will be written to the remote parameter <LocalCANRemoteParameterAdress> (0x5B11).

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5B13	0	LocalCANRemoteTransmission	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> (0x5B13) changes to 0. If the download operation is not successful the <LocalCANRemoteTransmission> (0x5B13) 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> (0x5B13) changes to 0. If the upload operation is not successful the <LocalCANRemoteTransmission> (0x5B13) changes to −1.

Table 68: Possible values of parameter <LocalCANRemoteTransmission> (0x5B13)

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> (0x5803#1):   0x80 (SYNC Protocol COB-ID)
<LocalCANEventTimer> (0x5803#3):   Sync cycle time in milliseconds
<LocalCANNumberOfMappedApplicationParametersInPdo> (0x5A03):   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 0x0002: Signed one byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x0002	0	DummyDataS08	INT8	rw	N	INT8	None

6.10.2 Object 0x0003: Signed two byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x0003	0	DummyDataS16	INT16	rw	N	INT16	None

6.10.3 Object 0x0004: Signed four byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x0004	0	DummyDataS32	INT32	rw	N	INT32	None

6.10.4 Object 0x0005: Unsigned one byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x0005	0	DummyDataU08	UINT8	rw	N	UINT8	None

6.10.5 Object 0x0006: Unsigned two byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x0006	0	DummyDataU16	UINT16	rw	N	UINT16	None

6.10.6 Object 0x0007: Unsigned four byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x0007	0	DummyDataU32	UINT32	rw	N	UINT32	None

6.10.7 Object 0x0008: Float32

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x0008	0	DummyDataF32	FLOAT32	rw	N	FLOAT32	None

6.10.8 Object 0x0009: Visible string

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x0009	0	DummyDataVisibleString	String	rw	N	64 byte	None

6.10.9 Object 0x290B: Signed one byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x290B	1	Integer08_1	INT8	rw	N	INT8	None
0x290B	2	Integer08_2	INT8	rw	N	INT8	None
0x290B	3	Integer08_3	INT8	rw	N	INT8	None
0x290B	4	Integer08_4	INT8	rw	N	INT8	None
0x290B	5	Integer08_5	INT8	rw	N	INT8	None
0x290B	6	Integer08_6	INT8	rw	N	INT8	None
0x290B	7	Integer08_7	INT8	rw	N	INT8	None
0x290B	8	Integer08_8	INT8	rw	N	INT8	None

6.10.10 Object 0x290C: Signed two byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x290C	1	Integer16_1	INT16	rw	N	INT16	None
0x290C	2	Integer16_2	INT16	rw	N	INT16	None
0x290C	3	Integer16_3	INT16	rw	N	INT16	None
0x290C	4	Integer16_4	INT16	rw	N	INT16	None
0x290C	5	Integer16_5	INT16	rw	N	INT16	None
0x290C	6	Integer16_6	INT16	rw	N	INT16	None
0x290C	7	Integer16_7	INT16	rw	N	INT16	None
0x290C	8	Integer16_8	INT16	rw	N	INT16	None

Moog ACV with CANopen bus interface
6 Signal routing and scaling
Free to use parameters

6.10.11 Object 0x290D: Signed four byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x290D	1	Integer32_1	INT32	rw	N	INT32	None
0x290D	2	Integer32_2	INT32	rw	N	INT32	None
0x290D	3	Integer32_3	INT32	rw	N	INT32	None
0x290D	4	Integer32_4	INT32	rw	N	INT32	None
0x290D	5	Integer32_5	INT32	rw	N	INT32	None
0x290D	6	Integer32_6	INT32	rw	N	INT32	None
0x290D	7	Integer32_7	INT32	rw	N	INT32	None
0x290D	8	Integer32_8	INT32	rw	N	INT32	None

6.10.12 Object 0x290E: Unsigned one byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x290E	1	Unsigned08_1	UINT8	rw	N	UINT8	None
0x290E	2	Unsigned08_2	UINT8	rw	N	UINT8	None
0x290E	3	Unsigned08_3	UINT8	rw	N	UINT8	None
0x290E	4	Unsigned08_4	UINT8	rw	N	UINT8	None
0x290E	5	Unsigned08_5	UINT8	rw	N	UINT8	None
0x290E	6	Unsigned08_6	UINT8	rw	N	UINT8	None
0x290E	7	Unsigned08_7	UINT8	rw	N	UINT8	None
0x290E	8	Unsigned08_8	UINT8	rw	N	UINT8	None

6.10.13 Object 0x290F: Unsigned two byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x290F	1	Unsigned16_1	UINT16	rw	N	UINT16	None
0x290F	2	Unsigned16_2	UINT16	rw	N	UINT16	None
0x290F	3	Unsigned16_3	UINT16	rw	N	UINT16	None
0x290F	4	Unsigned16_4	UINT16	rw	N	UINT16	None
0x290F	5	Unsigned16_5	UINT16	rw	N	UINT16	None
0x290F	6	Unsigned16_6	UINT16	rw	N	UINT16	None
0x290F	7	Unsigned16_7	UINT16	rw	N	UINT16	None
0x290F	8	Unsigned16_8	UINT16	rw	N	UINT16	None

6.10.14 Object 0x2910: Unsigned four byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x2910	1	Unsigned32_1	UINT32	rw	N	UINT32	None
0x2910	2	Unsigned32_2	UINT32	rw	N	UINT32	None
0x2910	3	Unsigned32_3	UINT32	rw	N	UINT32	None
0x2910	4	Unsigned32_4	UINT32	rw	N	UINT32	None
0x2910	5	Unsigned32_5	UINT32	rw	N	UINT32	None
0x2910	6	Unsigned32_6	UINT32	rw	N	UINT32	None
0x2910	7	Unsigned32_7	UINT32	rw	N	UINT32	None
0x2910	8	Unsigned32_8	UINT32	rw	N	UINT32	None

7 Servo valve functions

This chapter describes how the servo valve operates depending on the <ControlMode> (0x6043):

- 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 127
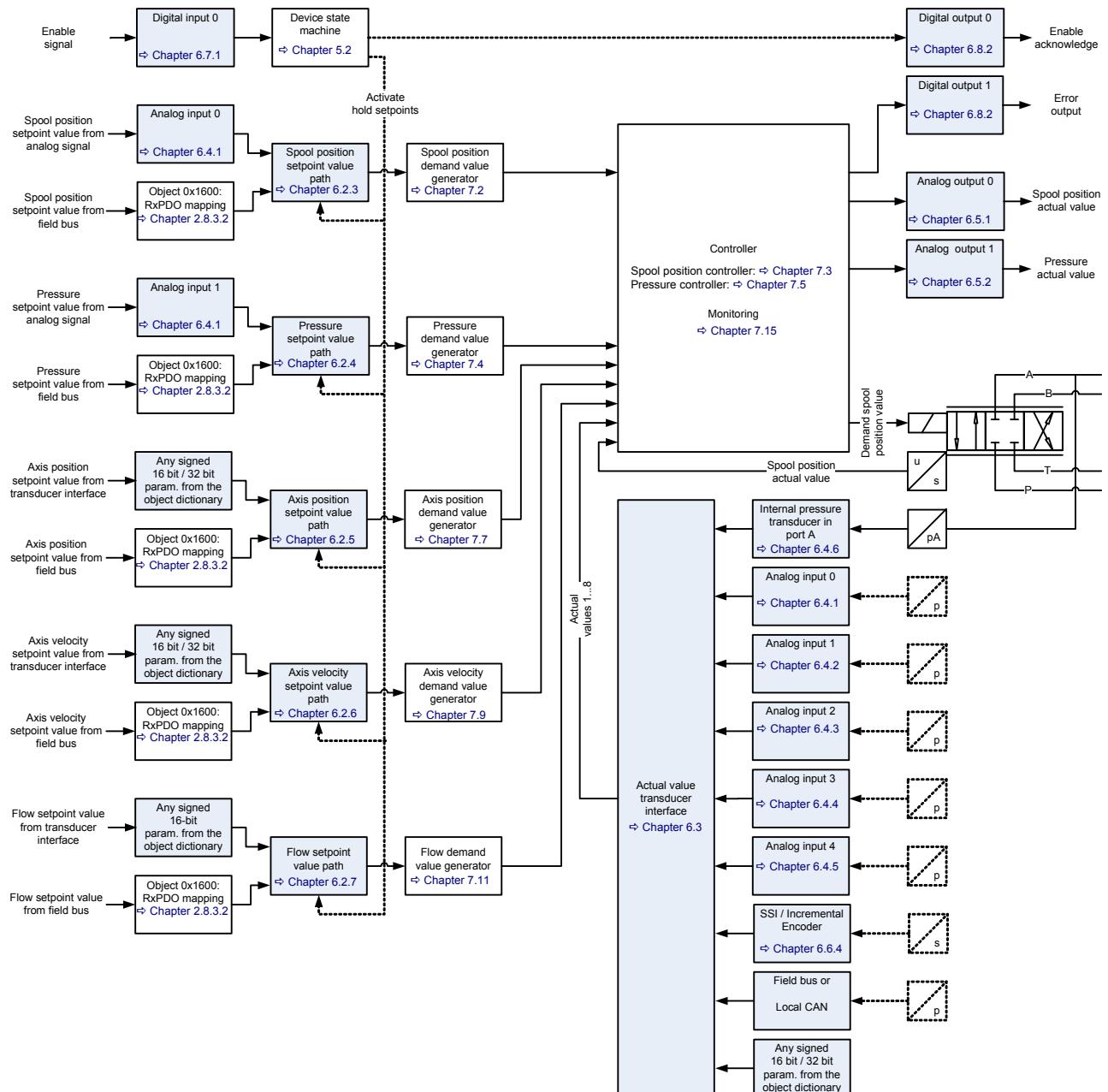
Figure 45: Servo valve controller and command signal conditioning

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 153
- Pressure controller (configured by user)
  ⇨ Chapter "7.5 Pressure controller", page 168
- Pressure demand signal polarity
  ⇨ Chapter "7.4.6 Pressure demand signal sign", page 167
- Spool position (Q) / pressure (P) switchover
  ⇨ Chapter "7.6 Spool position (Q) / pressure (P) switchover", page 187
- Axis position controller
  ⇨ Chapter "7.8 Axis position controller", page 195
- Axis velocity controller
  ⇨ Chapter "7.10 Axis velocity controller", page 209
- Flow controller
  ⇨ Chapter "7.12 Flow control", page 218
- Flow (Q) / pressure (P) switchover
  ⇨ Chapter "7.13 Flow / pressure (P) switchover", page 228

With these control structures the user can build several controllers depending on the <ControlMode> (0x6043).

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> (0x6043).

<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 129
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 130
3	Pressure control open loop Used for tests. Behaves like a closed loop Q control. ⇨ Chapter "7.1.6 Pressure control open loop", page 130
4	Pressure control closed loop Pressure / force control ⇨ Chapter "7.1.7 Pressure control closed loop", page 130
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 131
7	Axis velocity control ⇨ Chapter "7.10 Axis velocity controller", page 209
8	Axis force control (equal to the pressure closed loop control) ⇨ Chapter "7.1.7 Pressure control closed loop", page 130
9	Axis position control ⇨ Chapter "7.8 Axis position controller", page 195
13	Flow control ⇨ Chapter "7.12 Flow control", page 218
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 228

Table 69: Control mode values

7.1.1 Object 0x6043: Control mode

This parameter selects the servo valve control mode.

ⓘ   The available control modes are defined by the <Capability> (0x605F) parameter.

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6043	0	ControlMode	INT8	rw	N	−1…14	<ControlModeDefault> (0x4043)

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 70: Possible values of parameter <ControlMode> (0x6043)

7.1.2 Object 0x4043: Control mode default

The <ControlModeDefault> (0x4043) defines the control mode after power-up of the servo valve. During start-up of the servo valve, the parameter <ControlModeDefault> (0x4043) is copied to the parameter <ControlMode> (0x6043).

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x4043	0	ControlModeDefault	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> (0x6043) to 1 (spool position control open loop).
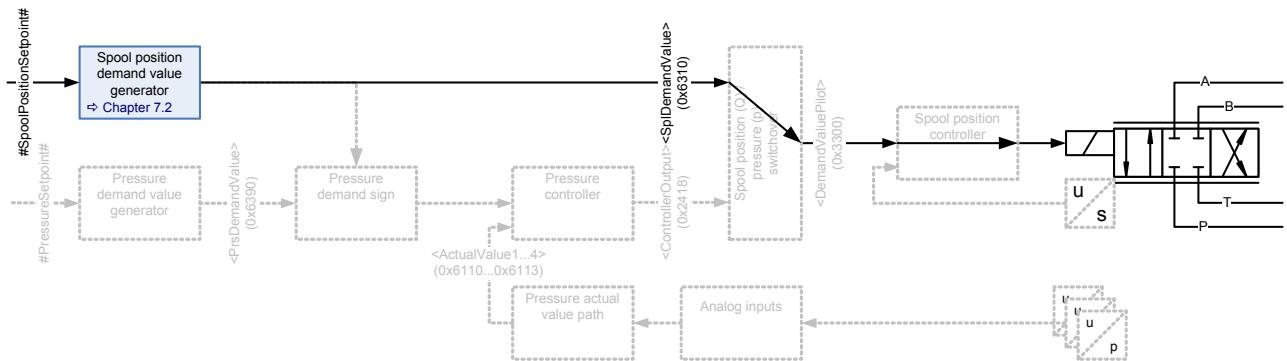
Figure 46: Spool position control open loop for single stage valves

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 128
⇨ Chapter "7.2.1 Object 0x6310: Demand value", page 136

ⓘ 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 59

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> (0x6043) 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.
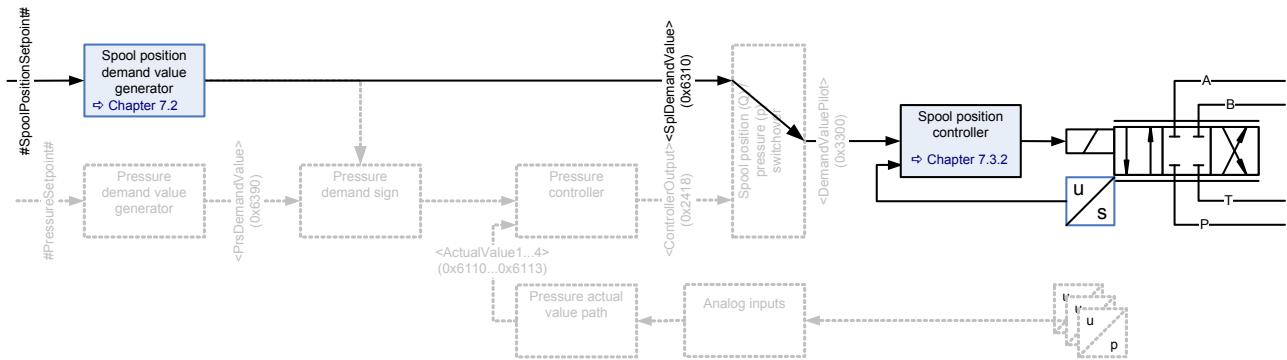
Figure 47: Spool position control open loop for dual stage valves

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 128
⇨ Chapter "7.2.1 Object 0x6310: Demand value", page 136

ⓘ 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 59

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> (0x6043) to 2 (spool position control closed loop).
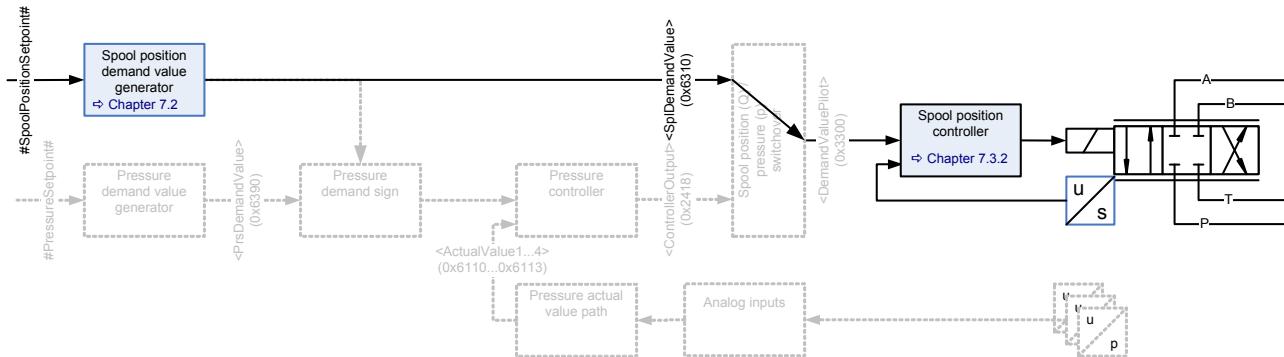
Figure 48: Spool position control closed loop for single stage and dual stage valves

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 128
⇨ Chapter "7.2.1 Object 0x6310: Demand value", page 136

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 59

7.1.6 Pressure control open loop

The pressure control open loop mode is selected by setting the parameter <ControlMode> (0x6043) 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 130

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> (0x6043) to 4 (pressure control closed loop).
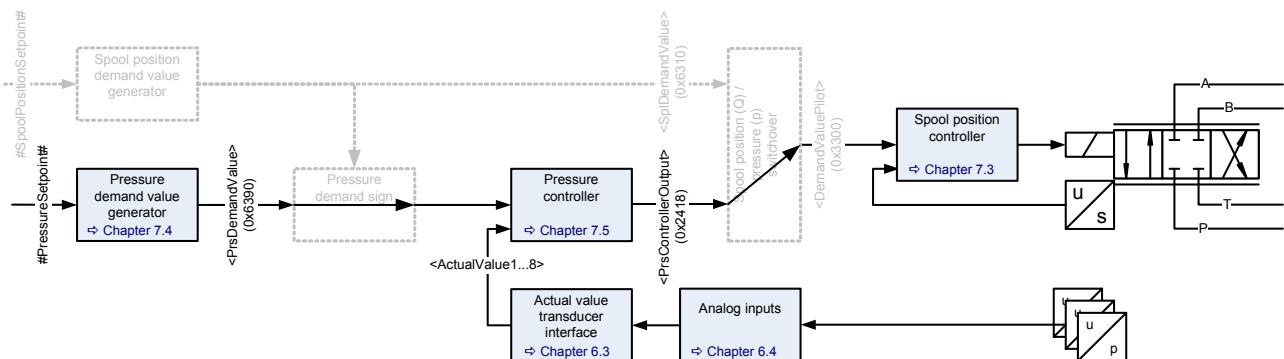
Figure 49: Pressure control closed loop

Description of the feedback signal <ActualValue1…8> (0x6210…0x6217):
⇨ Chapter "6.3 Drive transducer interface", page 72

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 128
⇨ Chapter "7.4.1 Object 0x6390: Demand value", page 159
⇨ Chapter "7.5.7 Object 0x2418: Controller output", page 170

ⓘ The signal #PressureSetpoint# is an internal signal only. It links the signal from the pressure set-point value path to the pressure demand value generator.
⇨ Chapter "6.2.4 Pressure setpoint value path", page 61

7.1.8 p/Q control closed loop

The p/Q control mode is selected by setting the parameter <ControlMode> (0x6043) to 5 (p/Q control).
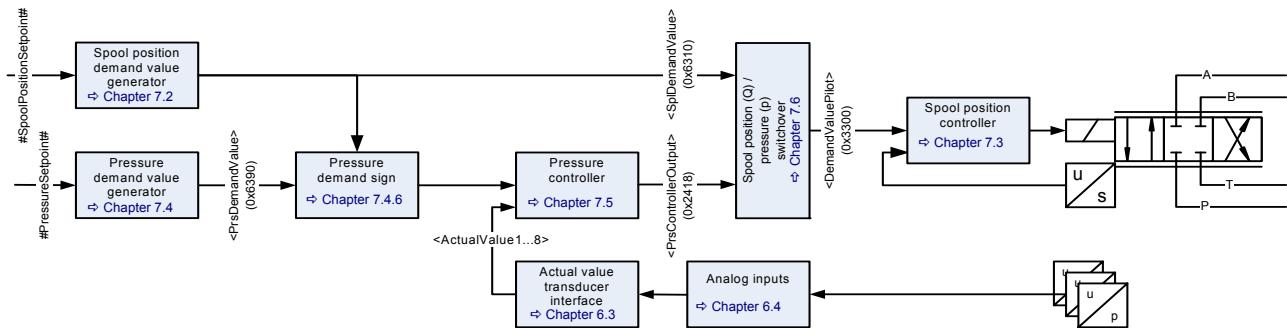
Figure 50: p/Q control closed loop

Description of the feedback signal <ActualValue1…8> (0x6210…0x6217):
⇨ Chapter "6.3 Drive transducer interface", page 72

⇨ Chapter "7.2.1 Object 0x6310: Demand value", page 136
⇨ Chapter "7.4.1 Object 0x6390: Demand value", page 159
⇨ Chapter "7.5.7 Object 0x2418: Controller output", page 170
⇨ Chapter "7.6.1.1 Object 0x3300: Demand value pilot", page 188

ⓘ The signal #PressureSetpoint# is an internal signal only. It links the signal from the pressure set-point value path to the pressure demand value generator.
⇨ Chapter "6.2.4 Pressure setpoint value path", page 61

ⓘ The signal #SpoolPositionSetpoint# is an internal signal only. It links the signal from the spool po-sition setpoint value path to the spool position demand value generator.
⇨ Chapter "6.2.3 Spool position setpoint value path", page 59

7.1.9 Axis position control

The axis position control mode is selected by setting the parameter <ControlMode> (0x6043) to 9 (axis position control closed loop).
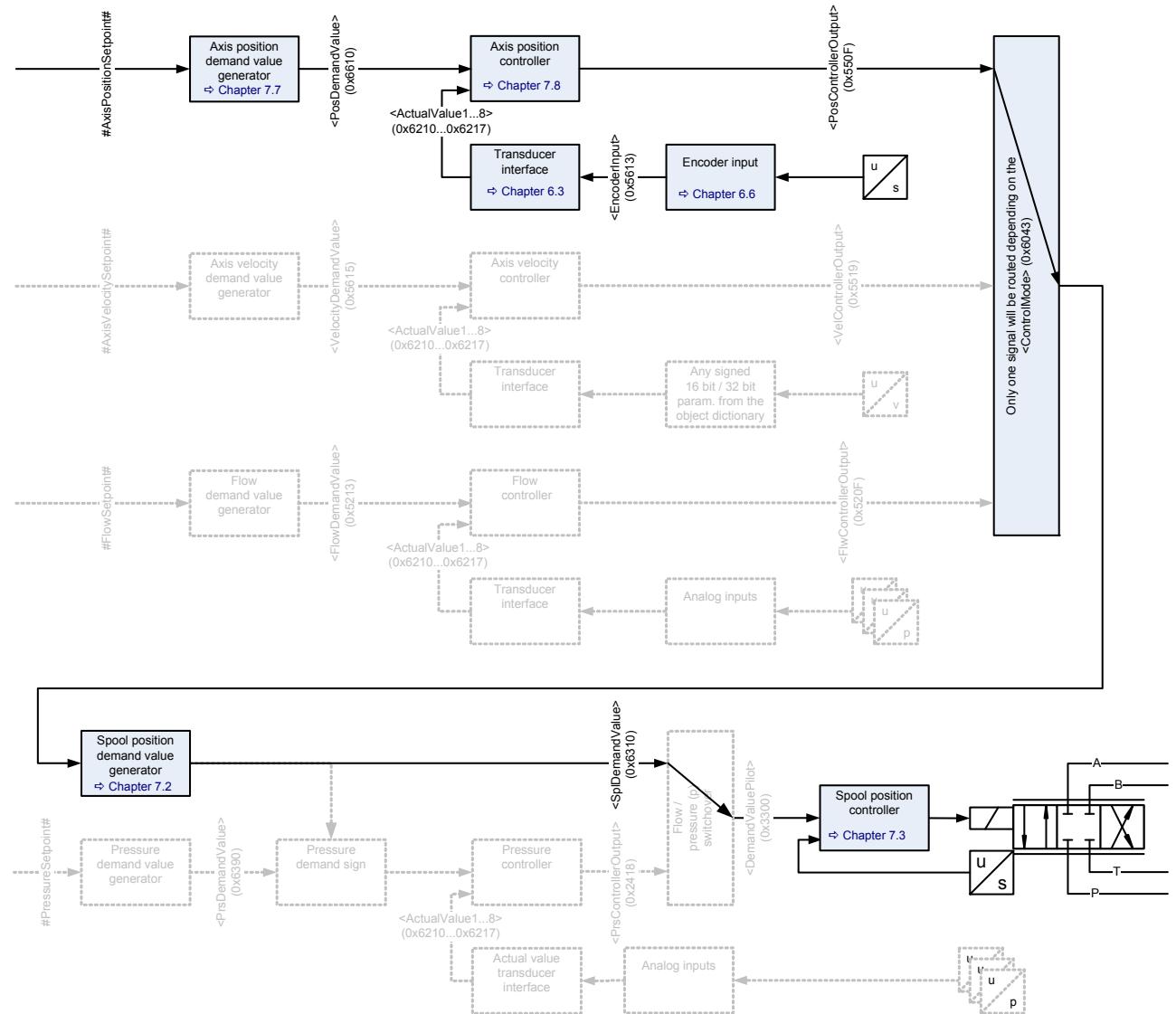
Figure 51: Axis position control

ⓘ 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 63

7.1.10 Axis velocity control

The axis velocity control mode is selected by setting the parameter <ControlMode> (0x6043) to 7 (axis velocity control closed loop).
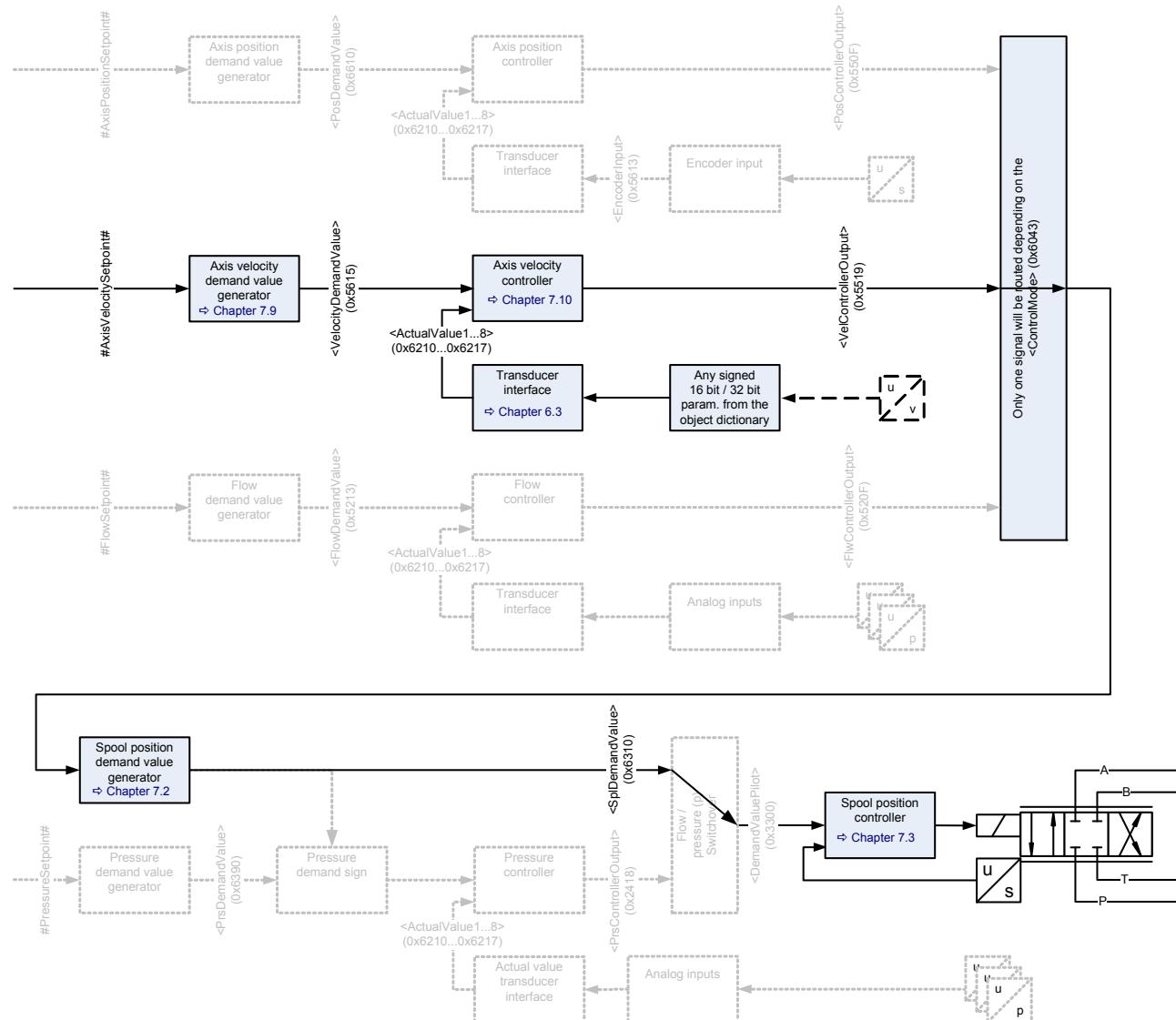
Figure 52: Axis velocity control

ⓘ 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 67

7.1.11 Flow control

The flow control mode is selected by setting the parameter <ControlMode> (0x6043) to 13 (flow control closed loop).
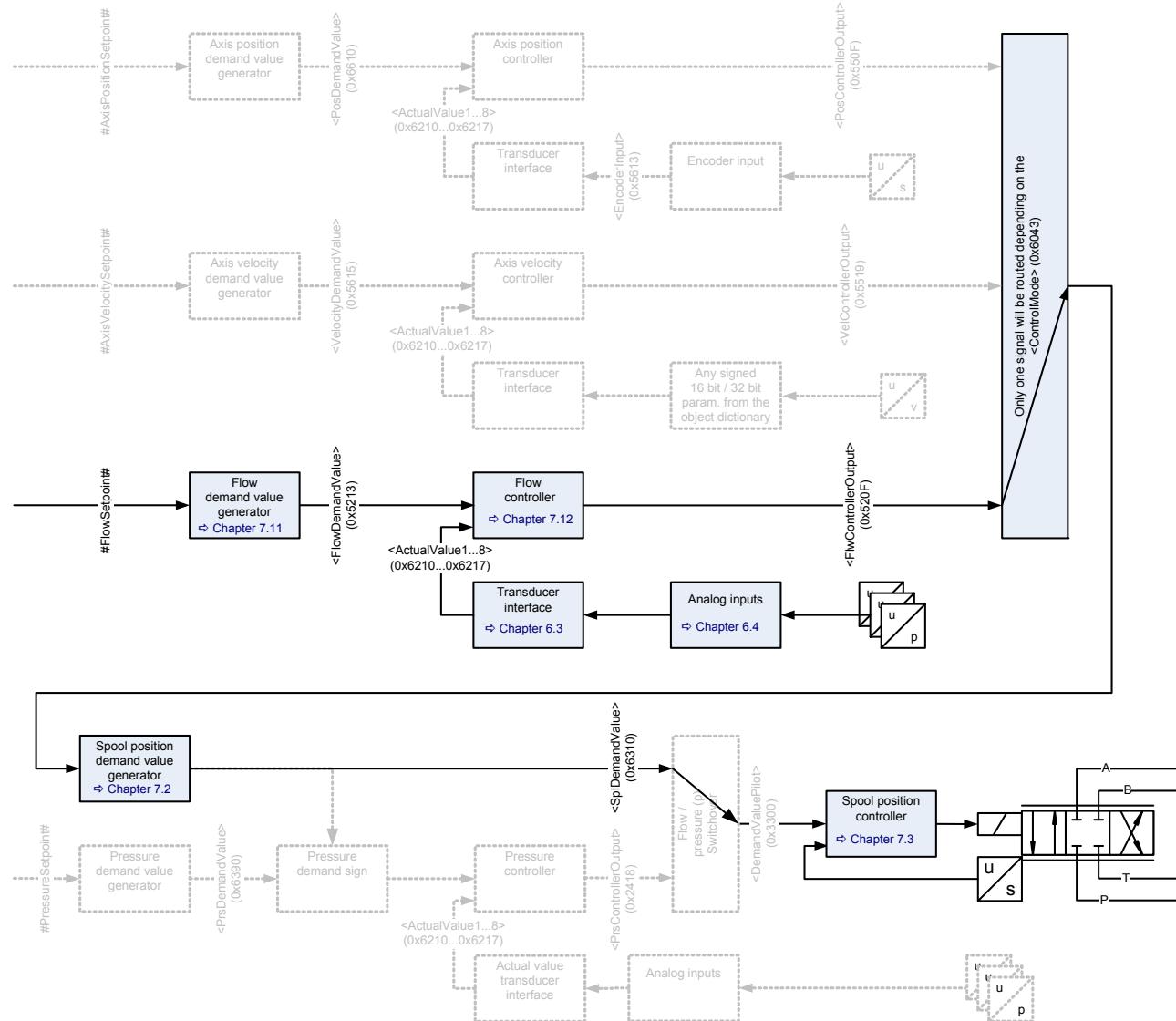
Figure 53: Flow control

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 70

7.1.12 p/flow control

The p/flow control mode is selected by setting the parameter <ControlMode> (0x6043) to 14 (p/flow control).
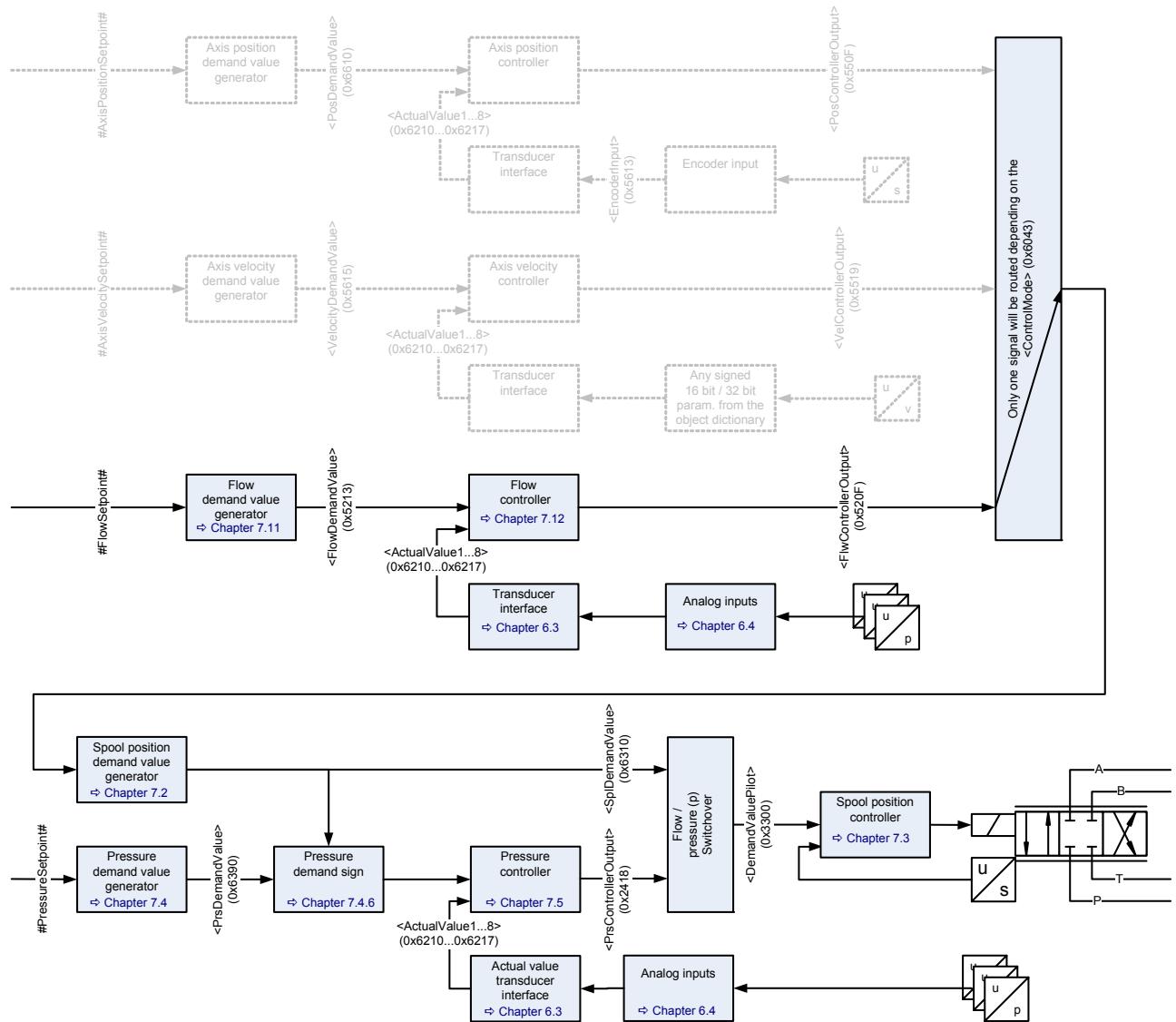
Figure 54: p/flow control

> ℹ️ The signal #PressureSetpoint# is an internal signal only. It links the signal from the pressure set-point value path to the pressure demand value generator.
> ⇨ Chapter "6.2.4 Pressure setpoint value path", page 61

> ℹ️ 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 70

7.1.13 Sample frequency

The <BasicSampleFrequency> (0x3030) 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
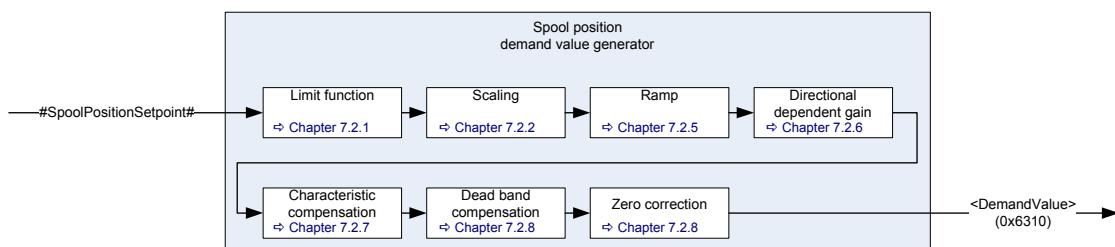
Figure 55: Spool position demand value generator

> (i) 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 59

7.2.1 Object 0x6310: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6310	1	SplDemandValue	INT16	ro	-	INT16	None
0x6310	2	Unit	UINT8	ro	-	UINT8	0
0x6310	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.2.2 Object 0x6311: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6311	1	SplReferenceValue	INT16	ro	-	INT16	16384
0x6311	2	Unit	UINT8	ro	-	UINT8	0
0x6311	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

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 0x6041: Status word", page 52
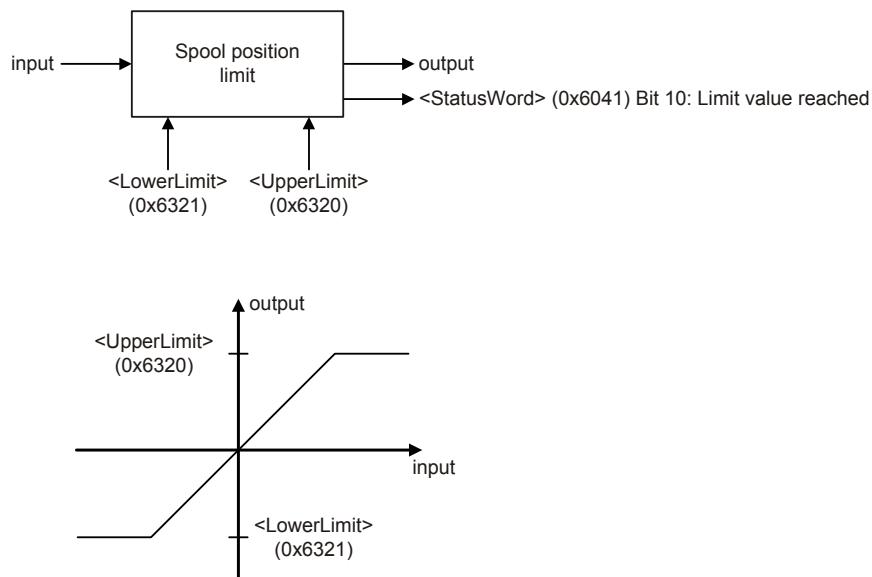
Figure 56: Limit function

ℹ The <UpperLimit> (0x6320) must be greater than the <LowerLimit> (0x6321).
If <LowerLimit> (0x6321) will be set greater than the <UpperLimit> (0x6320), the <UpperLimit> (0x6320) will be set to the value of the <LowerLimit> (0x6321).

7.2.3.1 Object 0x6320: Upper Limit

ValvePositionControl_DemandValueGenerator_Limit							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6320	1	UpperLimit	INT16	rw	Y	<LowerLimit> (0x6321)…32767	16384
0x6320	2	Unit	UINT8	ro	-	UINT8	0
0x6320	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.2.3.2 Object 0x6321: Lower Limit

ValvePositionControl_DemandValueGenerator_Limit							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6321	1	LowerLimit	INT16	rw	Y	–32768…<UpperLimit> (0x6320)	–16384
0x6321	2	Unit	UINT8	ro	-	UINT8	0
0x6321	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

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.

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

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

num: #Numerator#
den: #Denominator#
y_{off}: <Offset> (0x6323)
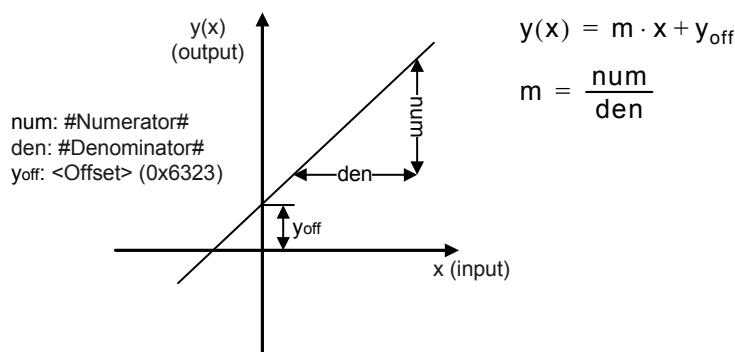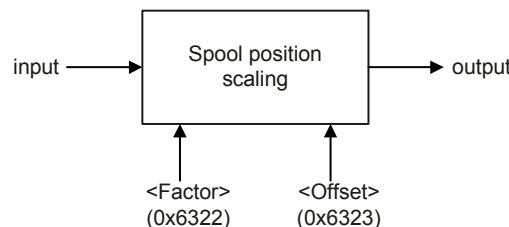
Figure 57: Scaling function

7.2.4.1 Object 0x6322: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6322	0	Factor	UINT32	rw	Y	UINT32	0x00010001

Value description

<Factor>			
Bit	31 16	15	0
Description	#Numerator#	#Denominator#	

Table 71: Data structure of the slope factor

7.2.4.2 Object 0x6323: Offset

This parameter is the offset of the linear output function.

ValvePressureControl_DemandValueGenerator_Scaling							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6323	1	Offset	INT16	rw	Y	INT16	0
0x6323	2	Unit	UINT8	ro	-	UINT8	0
0x6323	3	Prefix	INT8	ro	-	INT8	0

7.2.5 Ramp

The ramp function limits the slew rate of the input signal. The <Type> (0x6330) 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> (0x6041) bits are set:

<StatusWord> (0x6041) 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.

Figure 58: Ramp function

7.2.5.1 Object 0x6330: Type

This parameter defines the slope rate of the ramp dependent on the direction of movement.

ValvePositionControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6330	0	Type	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 72: Possible values of parameter <Type> (0x6330)

7.2.5.2 One-quadrant ramp (ramp type 1)

This function limits the input signal's rate of change to the defined <AccelerationTime> (0x6331).

This ramp type is active, if the parameter <Type> (0x6330) is set to 1.

Figure 59: Ramp type 1

7.2.5.2.1 Object 0x6331: 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 59, page 141. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6331	1	AccelerationTime	UINT16	rw	Y	UINT16	0
0x6331	2	Unit	UINT8	ro	-	UINT8	3
0x6331	3	AccelerationTime_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.2.5.3 Two-quadrant ramp (ramp type 2)

This function limits the input signal's rate of change to the defined <AccelerationTime> (0x6331) and <DeccelerationTime> (0x6332).

This ramp type is active, if the parameter <Type> (0x6330) is set to 2.

Figure 60: Ramp type 2

7.2.5.3.1 Object 0x6331: Acceleration time

This parameter is the same as the acceleration time for ramp type 1.

7.2.5.3.2 Object 0x6334: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6334	1	DecelerationTime	UINT16	rw	Y	UINT16	0
0x6334	2	Unit	UINT8	ro	-	UINT8	3
0x6334	3	DecelerationTime_Prefix	INT8	rw	Y	–4…0	–3

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> (0x6330) is set to 3.

Figure 61: Ramp type 3

7.2.5.4.1 Object 0x6332: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6332	1	AccelerationTimePositive	UINT16	rw	Y	UINT16	0
0x6332	2	Unit	UINT8	ro	-	UINT8	3
0x6332	3	AccelerationTimePositive_Prefix	INT8	rw	Y	−4…0	−3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.2.5.4.2 Object 0x6333: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6333	1	AccelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x6333	2	Unit	UINT8	ro	-	UINT8	3
0x6333	3	AccelerationTimeNegative_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.2.5.4.3 Object 0x6335: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6335	1	DecelerationTimePositive	UINT16	rw	Y	UINT16	0
0x6335	2	Unit	UINT8	ro	-	UINT8	3
0x6335	3	DecelerationTimePositive_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.2.5.4.4 Object 0x6336: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6336	1	DecelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x6336	2	Unit	UINT8	ro	-	UINT8	3
0x6336	3	DecelerationTimeNegative_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.2.6 Directional dependent gain

This function reduces the input signal's gain by a <DirectionalDependentGain_Factor> (0x6341) 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.

ℹ️ The actual flow depends not only on servo valve opening but on pressure loss in the servo valve and the system load.
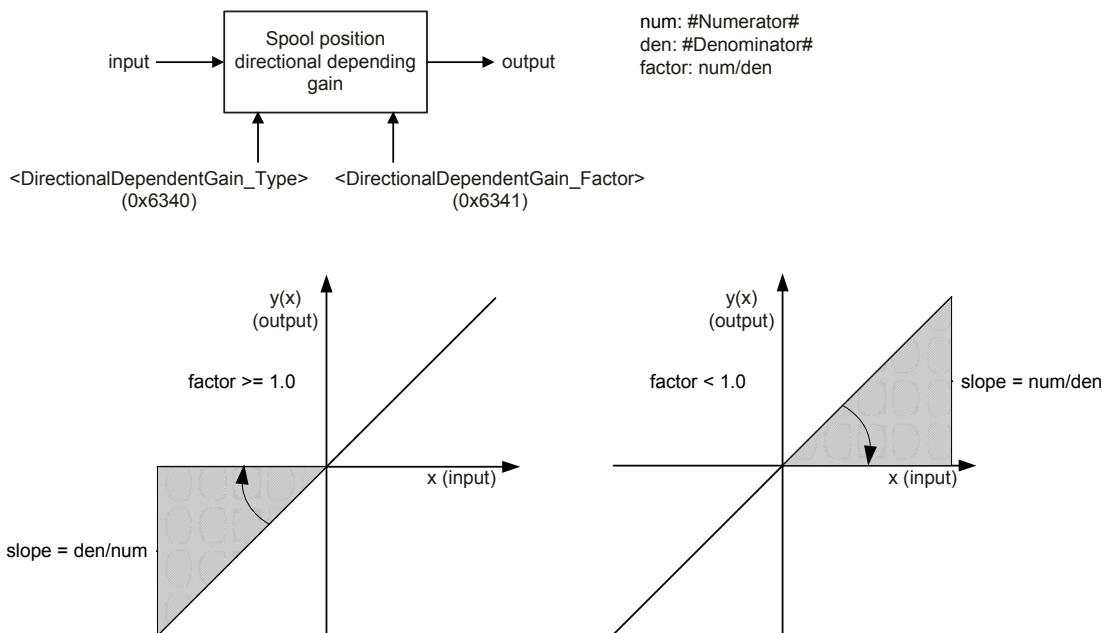
Figure 62: Directional depending gain

<Factor>	Input	Resulting output
|Factor| < 1	Input positive	$\text{Output} = \text{Input} \times \|\text{Factor}\|$
	Input negative	$\text{Output} = \text{Input}$
|Factor| ≥ 1	Input positive	$\text{Output} = \text{Input}$
	Input negative	$\text{Output} = \dfrac{\text{Input}}{\|\text{Factor}\|}$

Table 73: Definition of the directional dependent gain factor values

7.2.6.1 Object 0x6340: Type

This parameter switches the directional dependent gain function on or off.

ValvePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6340	0	DirectionalDependentGain_Type	INT8	rw	Y	0…1	0

Value description

<DirectionalDependentGain_Type>	Description
0	Directional dependent gain function switched off.
1	Directional dependent gain activated.

Table 74: Possible values of parameter <DirectionalDependentGain_Type> (0x6340)

7.2.6.2 Object 0x6341: 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> (0x6340) is set to 1 (directional dependent gain function activated). The default value 0x00010001 corresponds to the factor 1.0.

ValvePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6341	0	DirectionalDependentGain_Factor	UINT32	rw	Y	UINT32	0x00010001

Value description

<DirectionalDependentGain_Factor>			
Bit	31 16	15	0
Description	#Numerator#	#Denominator#	

Table 75: 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> (0x6346), 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.
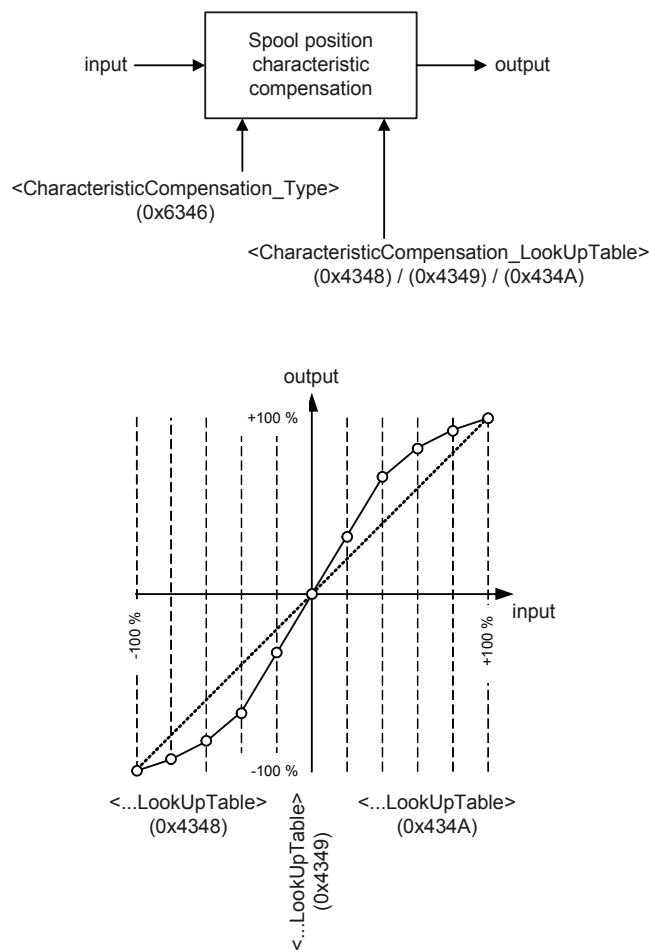
Figure 63: Characteristic compensation

7.2.7.1 Object 0x6346: Type

This parameter switches the characteristic compensation function on or off.

ValvePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6346	0	CharacteristicCompensation_Type	INT8	rw	Y	−1…0	0

Value description

<CharacteristicCompensation_Type>	Description
0	Characteristic compensation switched off.
−1	Characteristic compensation activated.

Table 76: Possible values of parameter <CharacteristicCompensation_Type> (0x6346)

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> (0x6346) set to –1.

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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x4347	0	CharacteristicCompensation_LookUpTable	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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x4348	1…128	CharacteristicCompensation_LookUpTable	INT16	rw	Y	INT16	0

7.2.7.2.3 Object 0x4349: Look-up table

This parameter contains the output values corresponding to the input value 0.

ValvePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x4349	0	CharacteristicCompensation_LookUpTable	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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x434A	1…128	CharacteristicCompensation_LookUpTable	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> (0x6342), 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> (0x6343) and <DeadbandCompensation_BSide> (0x6344).

The parameter <DeadbandCompensation_ASide> (0x6343) specifies the step height on the positive side and the parameter <DeadbandCompensation_BSide> (0x6344) the step height on the negative side. The border where the dead band compensation is effective is set by the parameter <DeadbandCompensation_Thresh-old> (0x6345).
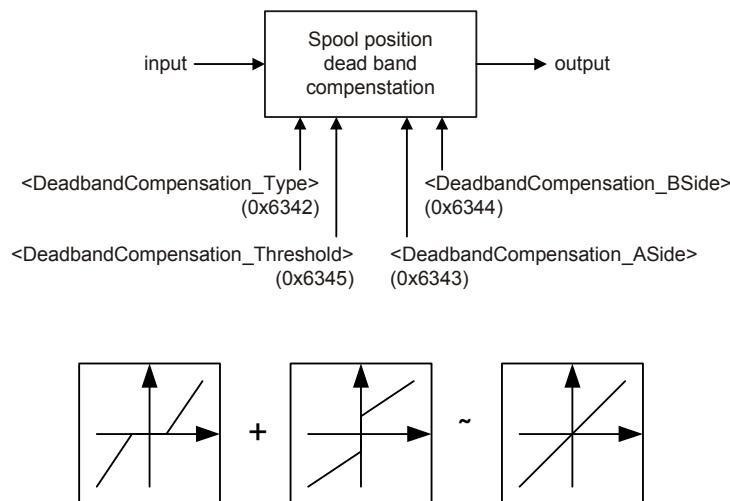
Figure 64: Dead band compensation

7.2.8.1 Object 0x6342: Type

This parameter is used to select the compensation type or to switch off the dead band compensation function.

ValvePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6342	0	DeadbandCompensation_Type	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 77: Possible values of parameter <DeadbandCompensation_Type> (0x6342)

7.2.8.2 Object 0x6343: A side

This parameter defines the step height of the dead band on the positive side.

ValvePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6343	1	DeadbandCompensation_ASide	INT16	rw	Y	0…16384	0
0x6343	2	Unit	UINT8	ro	-	UINT8	0
0x6343	3	Prefix	INT8	ro	-	INT8	0

7.2.8.3 Object 0x6344: B side

This parameter defines the step height of the dead band on the negative side.

ValvePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6344	1	DeadbandCompensation_BSide	INT16	rw	Y	0…16384	0
0x6344	2	Unit	UINT8	ro	-	UINT8	0
0x6344	3	Prefix	INT8	ro	-	INT8	0

7.2.8.4 Object 0x6345: Threshold

This parameter defines the positions of the steps (<DeadbandCompensation_ASide> (0x6343) and <DeadbandCompensation_BSide> (0x6344)). This determines the starting point of the dead band compensation step.

ValvePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6345	1	DeadbandCompensation_Threshold	INT16	rw	Y	0…16384	0
0x6345	2	Unit	UINT8	ro	-	UINT8	0
0x6345	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

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>, 0x6345), 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> (0x6343) or <DeadbandCompensation_BSide> (0x6344). 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> (0x6342) set to 1 (dead band jump function).
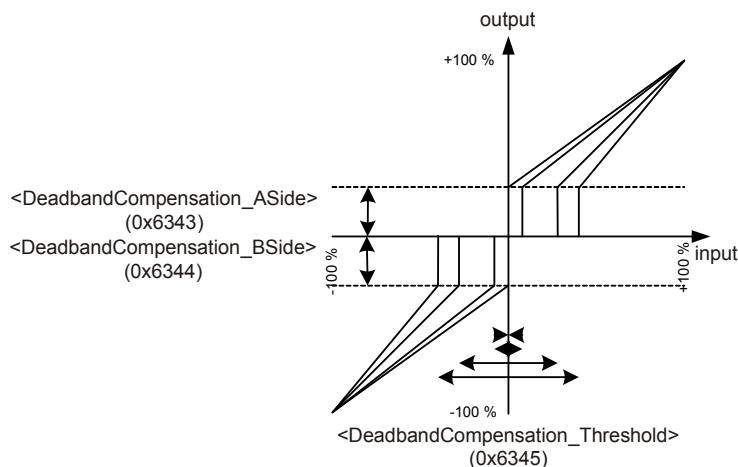
Figure 65: Dead band compensation type 1

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>, 0x6345), the output values will be interpolated between zero and the values defined by the parameters <DeadbandCompensation_ASide> (0x6343) or <DeadbandCompensation_BSide> (0x6344), respectively. At the threshold position the output signal is increased or decreased by the values set for the <DeadbandCompensation_ASide> (0x6343) or <DeadbandCompensation_BSide> (0x6344). 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> (0x6342) set to 2 (dead band continuous function).
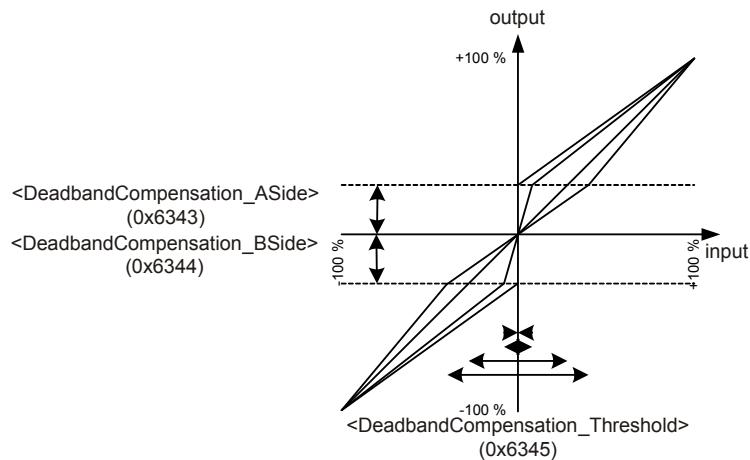
Figure 66: Dead band compensation type 2

7.2.9 Zero correction

The zero correction enables shifting of the input signal up and down by any desired offset. The <Offset> (0x6324) is added to the input signal according to the following figure.
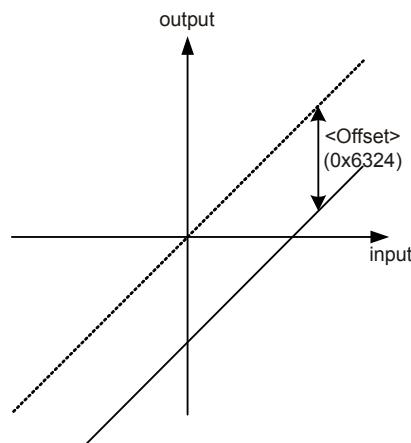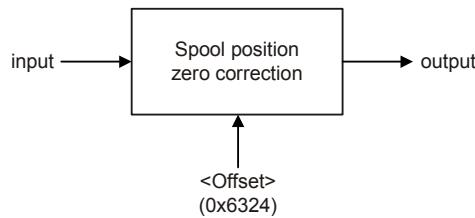
Figure 67: Zero correction

7.2.9.1 Object 0x6324: Offset

ValvePositionControl_DemandValueGenerator_ZeroCorrection							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6324	1	Offset	INT16	rw	Y	INT16	0
0x6324	2	Unit	UINT8	ro	-	UINT8	0
0x6324	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

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.
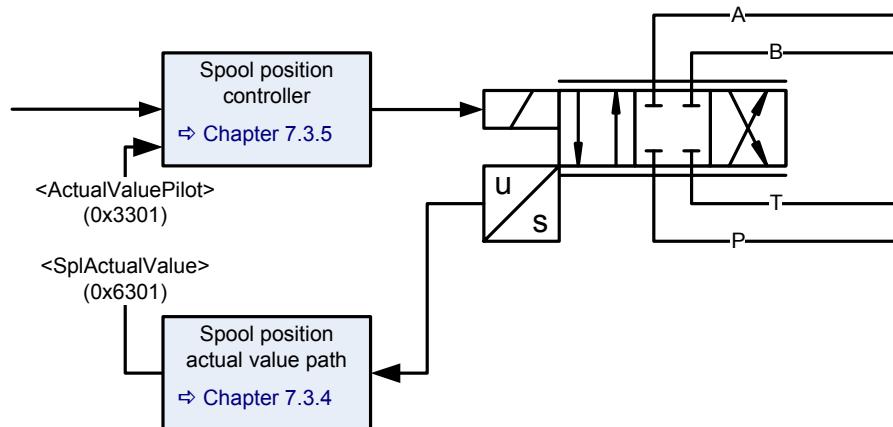
Figure 68: Single stage servo valve

7.3.2 Dual stage servo valve for open loop control

Control structure of a dual stage servo valve open loop control.
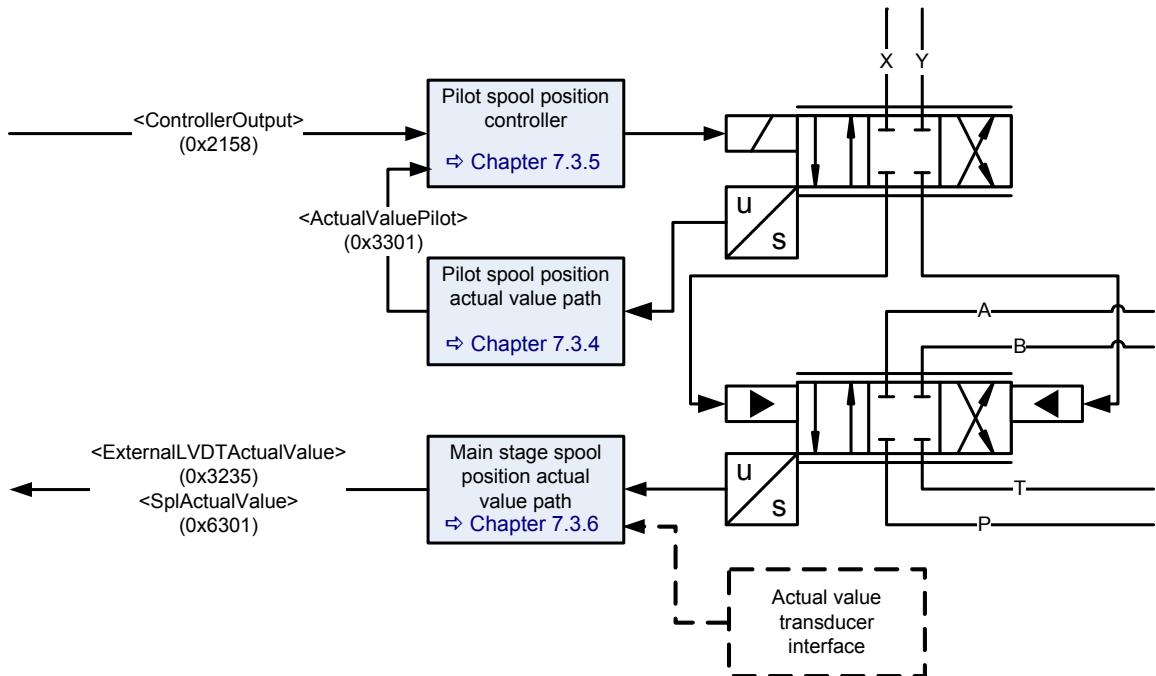
Figure 69: Dual stage servo valve for open loop control

7.3.3 Dual stage servo valve for closed loop control

Control structure of a dual stage servo valve closed loop control.
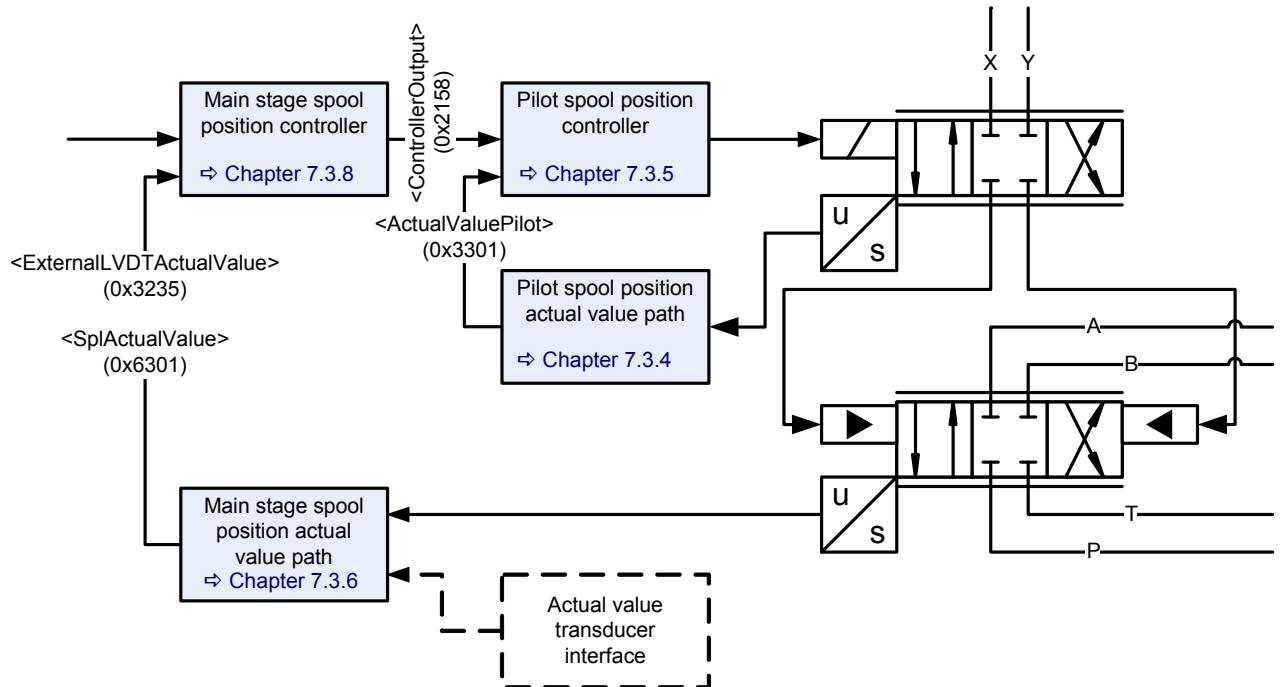
Figure 70: Dual stage servo valve for closed loop control

Moog ACV with CANopen bus interface
7 Servo valve functions
Spool position controller

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 0x6301: 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> (0x6301).

For a dual stage servo valve, the spool position value of the main stage is scaled and mapped to the same signal <SplActualValue> (0x6301).

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6301	1	SplActualValue	INT16	ro	-	INT16	None
0x6301	2	Unit	UINT8	ro	-	UINT8	0
0x6301	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.3.4.2 Object 0x3301: Actual value pilot

For a single stage servo valve, the spool position value is scaled and mapped to the signal <ActualValuePilot> (0x3301).

For a dual stage servo valve, the spool position value of the pilot stage is scaled and mapped to the same signal <ActualValuePilot> (0x3301).

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3301	0	ActualValuePilot	INT16	ro	-	INT16	None

7.3.4.3 Object 0x3506: Customer Scaling Offset

This parameter allows an offset of ± 5 % to the pilot valve spool position <ActualValuePilot> (0x3301). This offset may help for example in case of a temperature drift or another drift.

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3506	0	CustomerScalingOffset	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> (0x6350) holds the spool position control deviation. Also parameter <PilotControlDeviation> (0x3302) holds the spool 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> (0x6350) holds the main stage spool position control deviation. The parameter <PilotControlDeviation> (0x3302) holds the pilot stage spool control deviation.

7.3.5.1 Object 0x6350: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6350	1	SplControlDeviation	INT16	ro	-	INT16	None
0x6350	2	Unit	UINT8	ro	-	UINT8	0
0x6350	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.3.5.2 Object 0x3302: Pilot Control deviation

The pilot 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 pilot spool position. In case of a single stage valve, this parameter has the same value as <SplControlDeviation> (0x6350).

IntegratorTestValue							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3302	0	PilotControlDeviation	INT16	r	Y	INT16	-

7.3.5.3 Object 0x2416: Integrator Test Value

This parameter holds the controller output of the I-part of the pilot spool position controller.

IntegratorTestValue							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2416	0	IntegratorTestValue	INT32	r	Y	INT32	-

7.3.5.4 Object 0x241F: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x241F	0	CustomerOverallGain	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> (0x3235) and the signal <SplActualValue> (0x6301).

⇨ Chapter "7.3.4.1 Object 0x6301: Actual value", page 155

7.3.6.1 Object 0x3237#1…3: Customer scaling external LVDT

This parameter is used to scale the input from the external LVDT to the actual main stage spool position.

$$\text{ExternalLVDTActualValue (0x3235)} = \frac{(\text{Input} + \text{ExternalLvdtOffset (0x3237\#3)}) \times \text{ExternalLvdtScaNumerator (0x3237\#1)}}{\text{ExternalLvdtScaDenominator (0x3237\#2)}}$$

ExternalLVDT							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3237	1	ExternalLvdtScaNumerator	INT16	rw	Y	INT16	16386
0x3237	2	ExternalLvdtScaDenominator	INT16	rw	Y	INT16	16386
0x3237	3	ExternalLvdtOffset	INT16	rw	Y	INT16	0

7.3.6.2 Object 0x3235: External LVDT Actual Value

This parameter holds the main stage spool position actual value.

ExternalLVDT							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3235	0	ExternalLVDTActualValue	INT16	ro	-	INT16	None

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 0x2149: 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> (0x2149) = 0 then <SplActualValue> (0x6301) is used, otherwise the output of the interface number <ActiveTransducerInterfaceMainStage> (0x2149).

ValveMainStageControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2149	0	ActiveTransducerInterfaceMainStage	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.

7.3.8.1 Object 0x215C: Main stage customer overall gain

The internal controller output will be multiplied with this gain to get the <ControllerOutput> (0x2158).

ValveMainStageControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x215C	0	MainStageCustomerOverallGain	FLOAT32	rw	Y	0.0…2.0	1.0

7.3.8.2 0x2170: Main stage controller proportional part output

This parameter holds the controller output of the P-part of the main stage spool position controller.

MainStageProportionalPart							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2170	0	MainStageProportionalPart	INT32	r	Y	INT32	-

7.3.8.3 0x2171: Main stage controller integral part output

This parameter holds the controller output of the I-part of the main stage spool position controller.

MainStageIntegralPart							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2171	0	MainStageIntegralPart	INT32	r	Y	INT32	-

7.3.8.4 0x2172: Main stage controller differential part output

This parameter holds the controller output of the D-part of the main stage spool position controller.

MainStageDifferentialPart							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2172	0	MainStageDifferentialPart	INT32	r	Y	INT32	-

7.3.8.5 Object 0x2158: Controller output

This parameter holds the main spool position controller output.

ValveMainStageControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2158	0	ControllerOutput	INT16	ro	-	INT16	None

7.4 Pressure setpoint conditioning / demand value generator
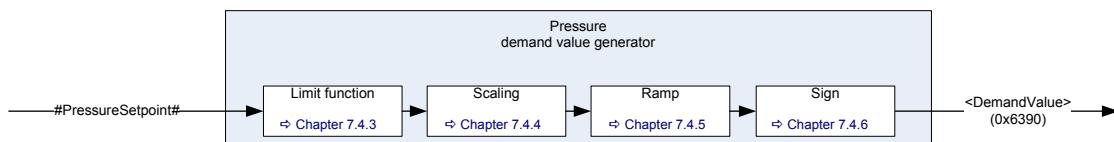
Figure 71: Pressure demand value generator

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 61

7.4.1 Object 0x6390: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6390	1	PrsDemandValue	INT16	ro	N	INT16	None
0x6390	2	Unit	UINT8	ro	-	UINT8	0
0x6390	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.4.2 Object 0x6391: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6391	1	PrsReferenceValue	INT16	ro	-	INT16	16384
0x6391	2	Unit	UINT8	ro	-	UINT8	0
0x6391	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

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 0x6041: Status word", page 52
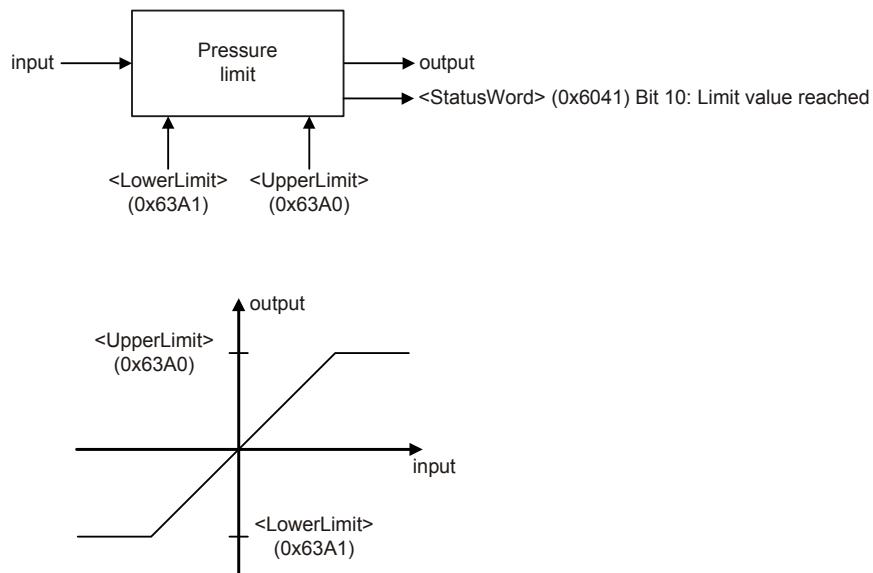
Figure 72: Limit function

> ⓘ The <UpperLimit> (0x63A0) must be greater than the <LowerLimit> (0x63A1).
> If the <LowerLimit> (0x63A1) is greater than the <UpperLimit> (0x63A0), the <UpperLimit> (0x63A0) will be set to the value of the <LowerLimit> (0x63A1).

7.4.3.1 Object 0x63A0: Upper Limit

ValvePositionControl_DemandValueGenerator_Limit							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63A0	1	UpperLimit	INT16	rw	Y	<LowerLimit> (0x63A1)…32767	32760
0x63A0	2	Unit	UINT8	ro	-	UINT8	0
0x63A0	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.4.3.2 Object 0x63A1: Lower Limit

ValvePositionControl_DemandValueGenerator_Limit							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63A1	1	LowerLimit	INT16	rw	Y	–32767…<UpperLimit> (0x63A0)	–32760
0x63A1	2	Unit	UINT8	ro	-	UINT8	0
0x63A1	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

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.


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

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

num: #Numerator#
den: #Denominator#
y_{off}: <Offset> (0x63A3)
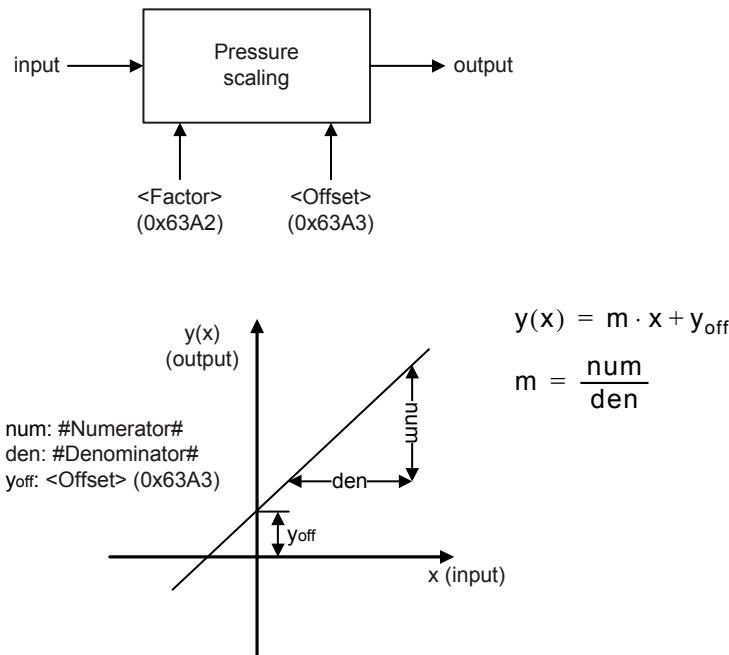
Figure 73: Scaling function

7.4.4.1 Object 0x63A2: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63A2	0	Factor	UINT32	rw	Y	UINT32	0x00010001

Value description

<Factor>			
Bit	31 16	15	0
Description	#Numerator#	#Denominator#	

Table 78: Data structure of the slope factor

7.4.4.2 Object 0x63A3: Offset

This parameter is the offset of the linear output function.

ValvePressureControl_DemandValueGenerator_Scaling							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63A3	1	Offset	INT16	rw	Y	INT16	0
0x63A3	2	Unit	UINT8	ro	-	UINT8	0
0x63A3	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.4.5 Ramp

The ramp function limits the slew rate of the input signal. The <Type> (0x63B0) 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> (0x6041) bits are set:

<StatusWord> (0x6041) 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 0x6041: Status word", page 52
⇨ Chapter "5.1.1 Object 0x604F: Local", page 43
⇨ Chapter "5.1.2 Object 0x6040: Control word", page 44
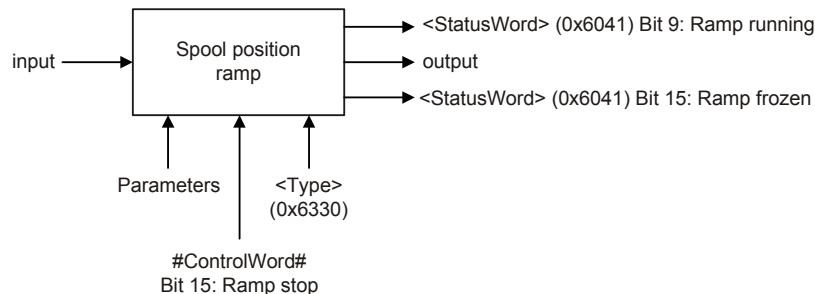⇨ Chapter "5.1.3 Object 0x4040: Local control word", page 45
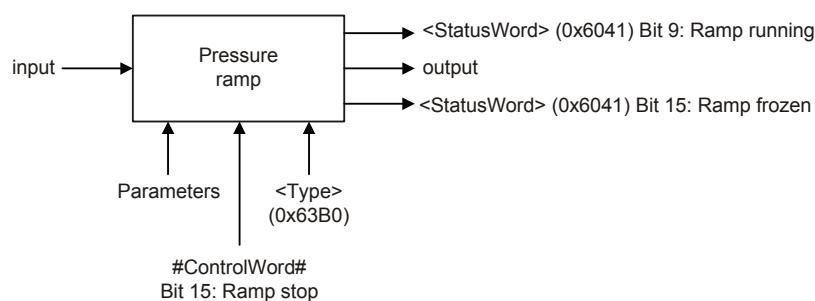
Figure 74: Ramp function

7.4.5.1 Object 0x63B0: Type

This parameter defines the progression of the ramp.

ValvePressureControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63B0	0	Type	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 79: Possible values of parameter <Type> (0x63B0)

7.4.5.2 One-quadrant ramp (ramp type 1)

This function limits the input signal's rate of change to the defined <AccelerationTime> (0x63B1).

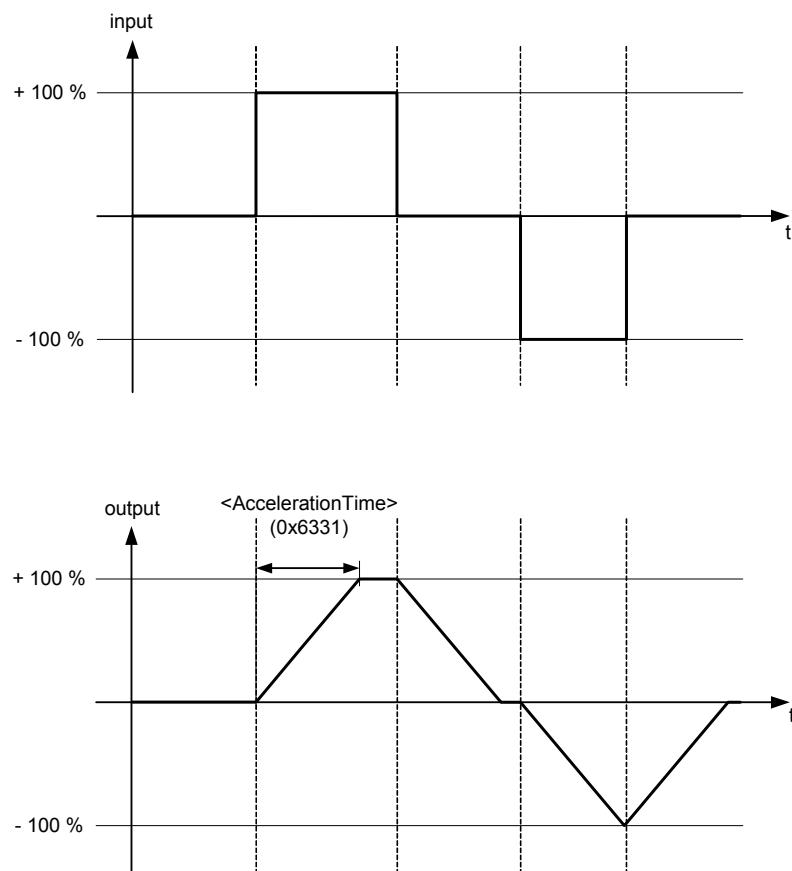This ramp type is active, if the parameter <Type> (0x63B0) is set to 1.
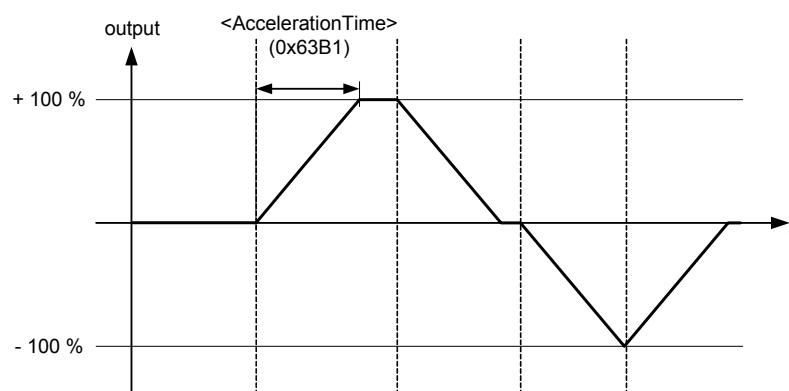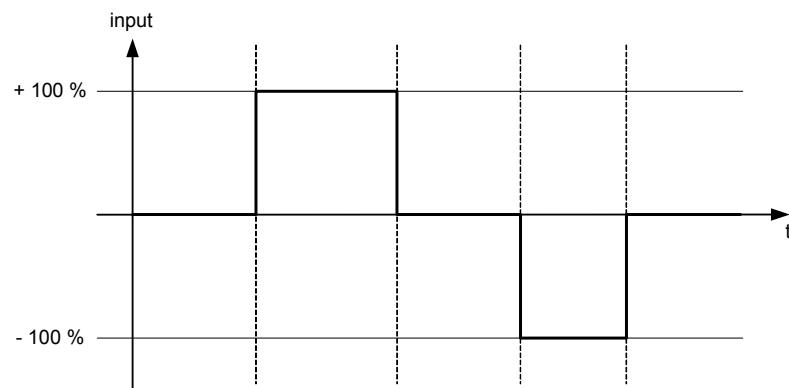
Figure 75: Ramp type 1

7.4.5.2.1 Object 0x63B1: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63B1	1	AccelerationTime	UINT16	rw	Y	UINT16	0
0x63B1	2	Unit	UINT8	ro	-	UINT8	3
0x63B1	3	AccelerationTime_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.4.5.3 Two-quadrant ramp (ramp type 2)

This function limits the input signal's rate of change to the defined <AccelerationTime> (0x63B1) and <DeccelerationTime> (0x63B4).

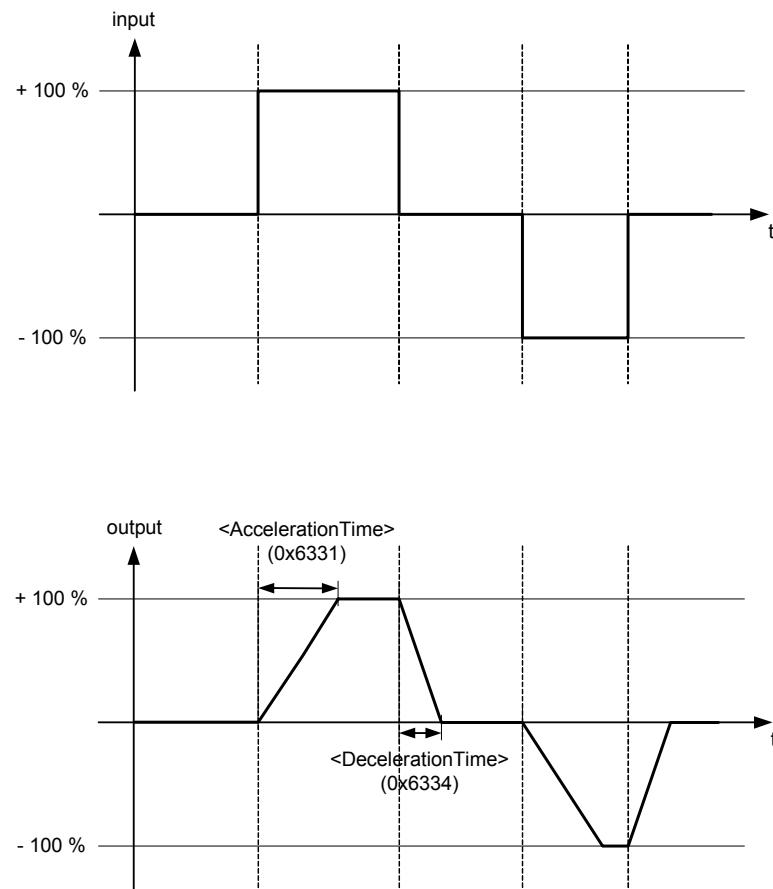This ramp type is active, if the parameter <Type> (0x63B0) is set to 2.
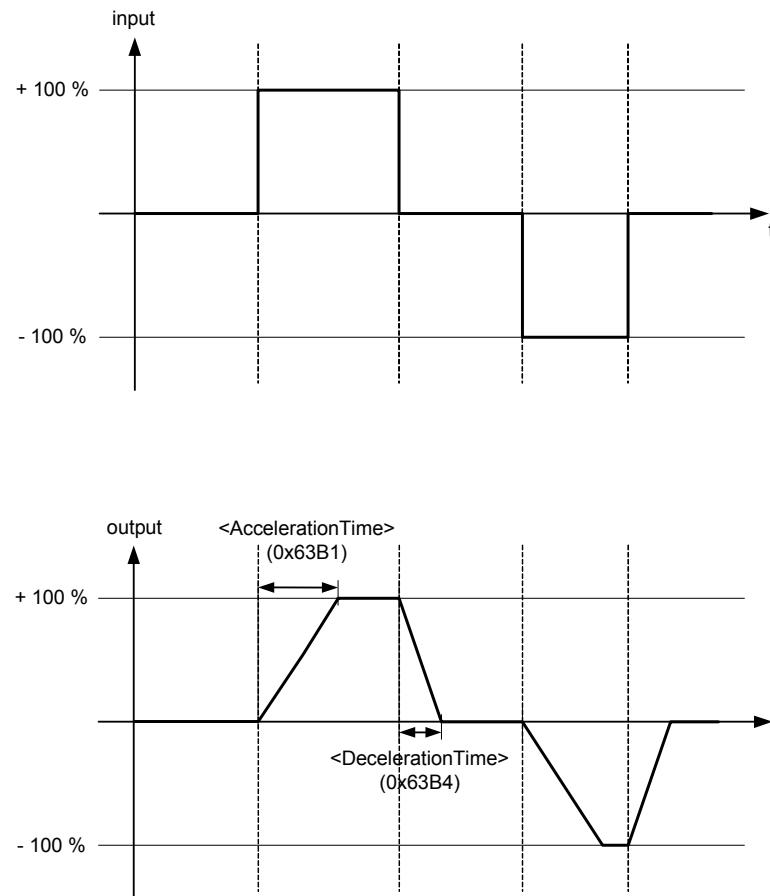
Figure 76: Ramp type 2

7.4.5.3.1 Object 0x63B1: Acceleration time

This parameter is the same as the acceleration time for ramp type 1.

⇨ Chapter "7.4.5.2.1 Object 0x63B1: Acceleration time", page 164

7.4.5.3.2 Object 0x63B4: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63B4	1	DecelerationTime	UINT16	rw	Y	UINT16	0
0x63B4	2	Unit	UINT8	ro	-	UINT8	3
0x63B4	3	DecelerationTime_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

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.

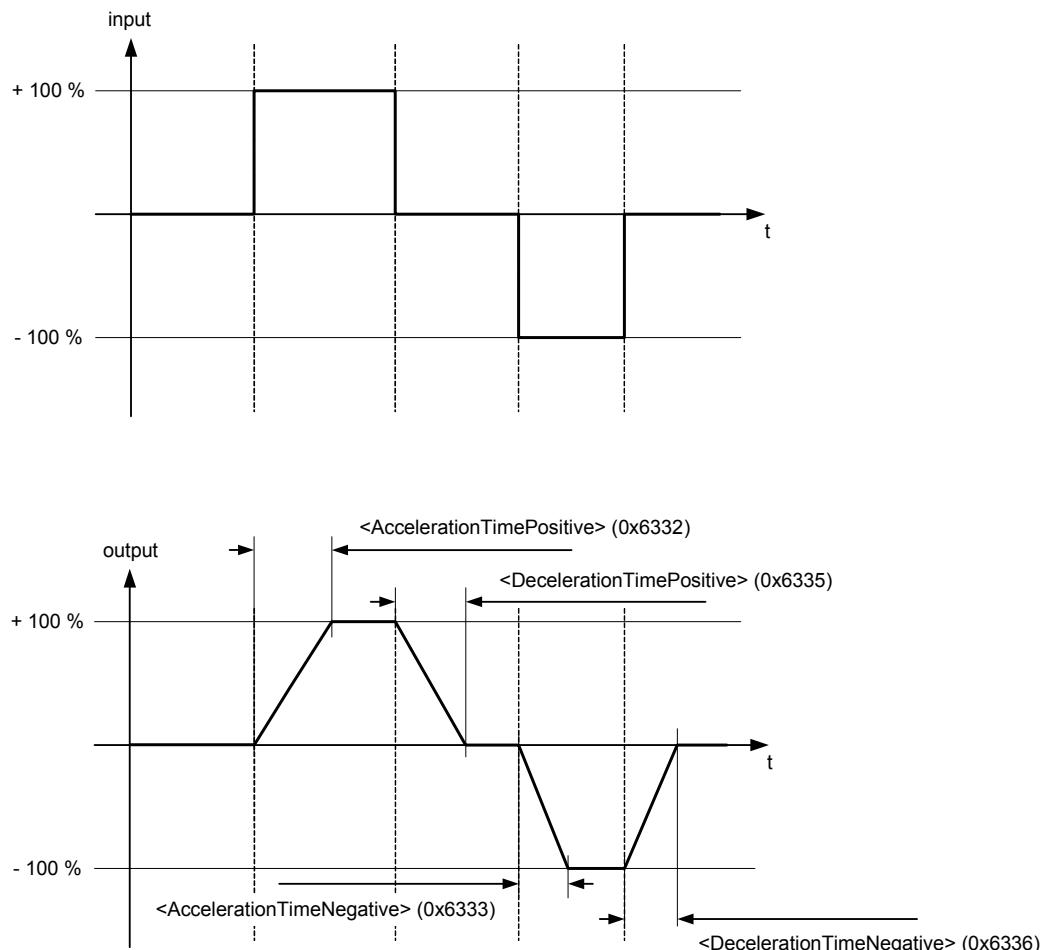This ramp type is active, if the parameter <Type> (0x63B0) is set to 3.
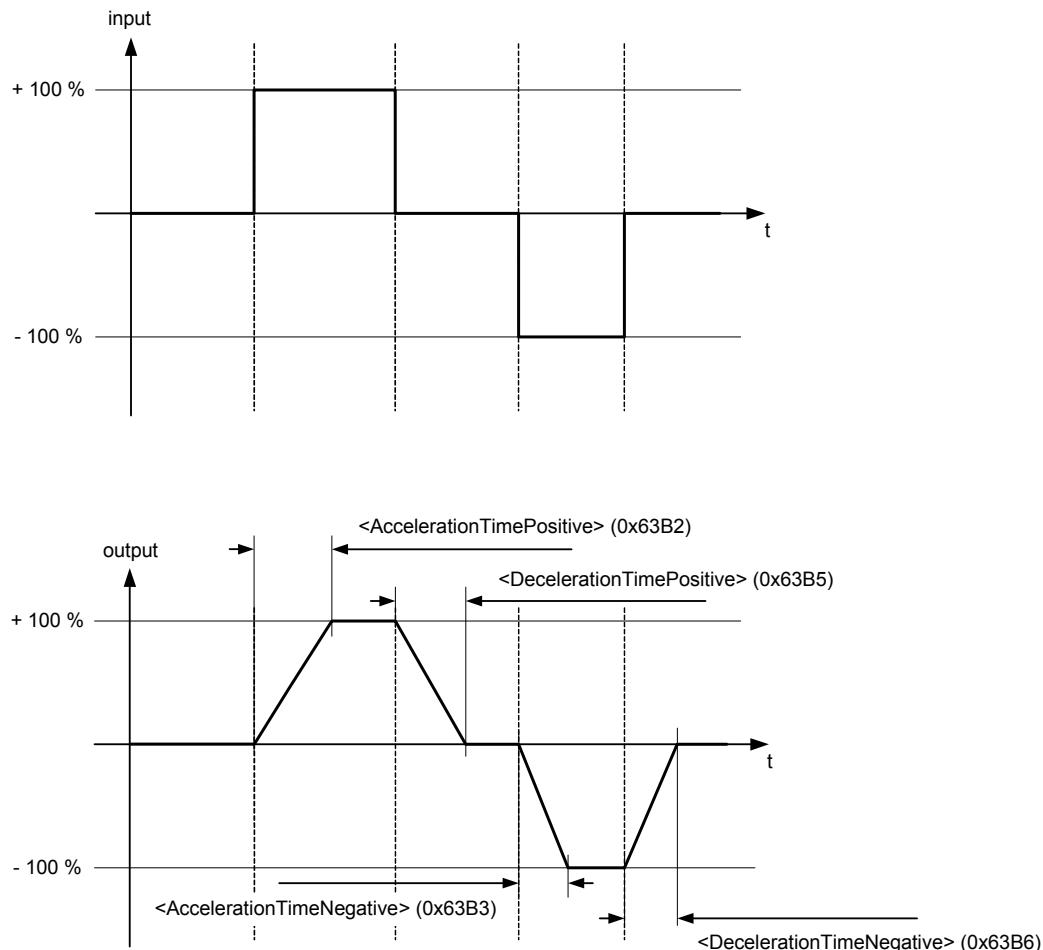
Figure 77: Ramp type 3

7.4.5.4.1 Object 0x63B2: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63B2	1	AccelerationTimePositive	UINT16	rw	Y	UINT16	0
0x63B2	2	Unit	UINT8	ro	-	UINT8	3
0x63B2	3	AccelerationTimePositive_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.4.5.4.2 Object 0x63B3: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63B3	1	AccelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x63B3	2	Unit	UINT8	ro	-	UINT8	3
0x63B3	3	AccelerationTimeNegative_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.4.5.4.3 Object 0x63B5: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63B5	1	DecelerationTimePositive	UINT16	rw	Y	UINT16	0
0x63B5	2	Unit	UINT8	ro	-	UINT8	3
0x63B5	3	DecelerationTimePositive_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.4.5.4.4 Object 0x63B6: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63B6	1	DecelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x63B6	2	Unit	UINT8	ro	-	UINT8	3
0x63B6	3	DecelerationTimeNegative_Prefix	INT8	rw	Y	−4…0	−3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.4.6 Pressure demand signal sign

In the pQ control mode, the <ControlMode> (0x6043) 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.
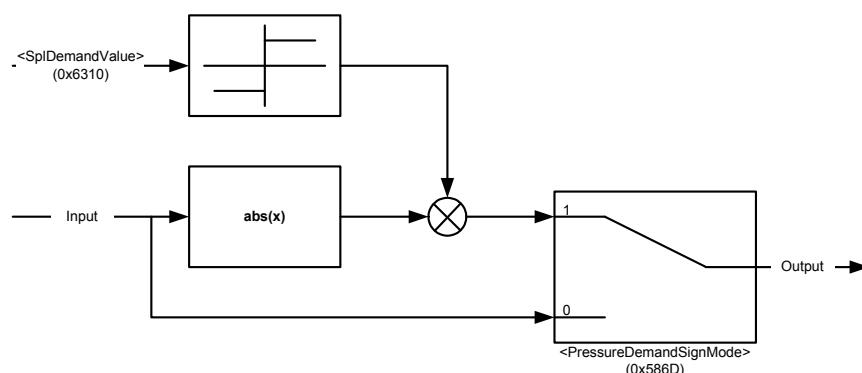
Figure 78: Pressure demand signal sign

7.4.6.1 Object 0x586D: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x586D	0	PressureDemandSignMode	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 80: Possible values of parameter <PressureDemandSignMode> (0x586D)

7.5 Pressure controller
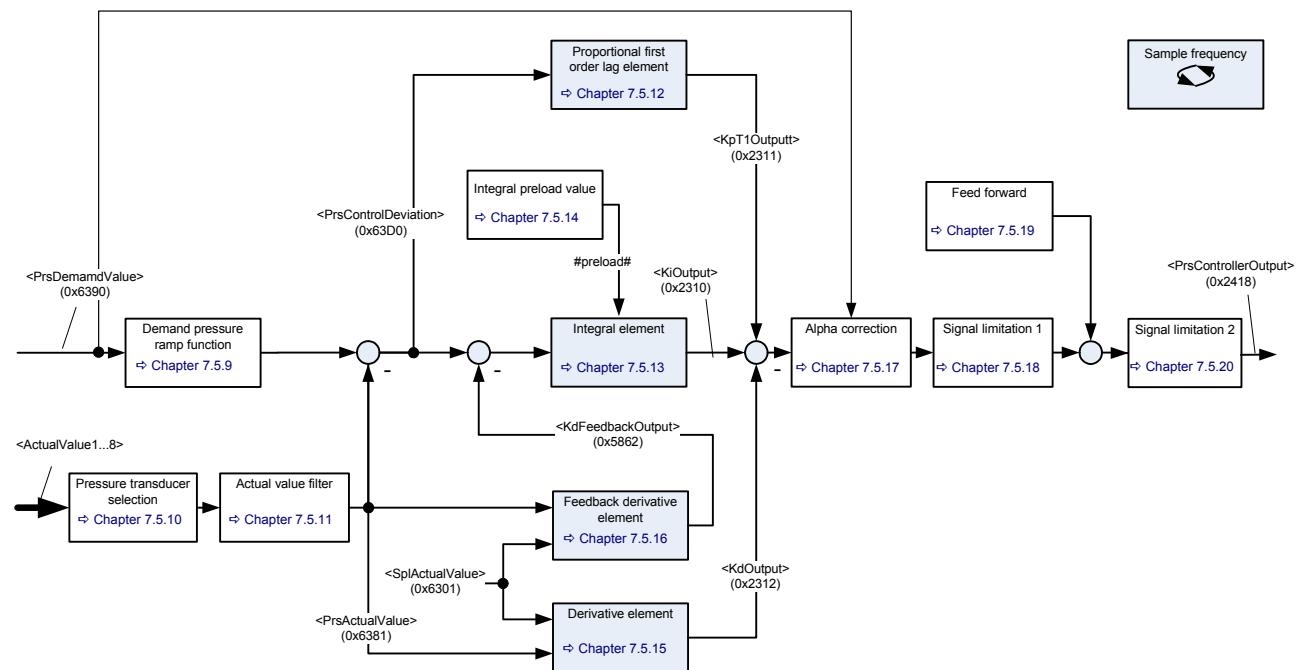
Figure 79: Pressure controller

⇨ Chapter "7.4.1 Object 0x6390: Demand value", page 159
⇨ Chapter "7.5.2 Object 0x63D0: Control deviation", page 169
⇨ Chapter "7.5.5 Object 0x2312: Kd output", page 169
⇨ Chapter "7.5.6 Object 0x5862: Kd feedback output", page 169
⇨ Chapter "7.5.4 Object 0x2310: Ki output", page 169
⇨ Chapter "7.5.3 Object 0x2311: Kp T1 output", page 169
⇨ Chapter "7.5.7 Object 0x2418: Controller output", page 170

Description of the feedback signal <ActualValue1…8> (0x6210…0x6217):
⇨ Chapter "6.3 Drive transducer interface", page 72

7.5.1 Object 0x6381: Actual value

The filter output <PrsActualValue> (0x6381) is the input of the pressure controller.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x6381	1	PrsActualValue	INT16	ro	-	INT16	None
0x6381	2	Unit	UINT8	ro	-	UINT8	0
0x6381	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.5.2 Object 0x63D0: Control deviation

This parameter holds the deviation between the ramped pressure setpoint value and the filtered pressure actual value.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x63D0	1	PrsControlDeviation	INT16	ro	-	INT16	None
0x63D0	2	Unit	UINT8	ro	-	UINT8	0
0x63D0	3	Prefix	INT8	ro	-	INT8	0

7.5.3 Object 0x2311: Kp T1 output

This parameter holds the output of the proportional element of the pressure controller.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2311	0	KpT1Output	FLOAT32	ro	-	FLOAT32	None

7.5.4 Object 0x2310: Ki output

This parameter holds the output of the integrator element of the pressure controller.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2310	0	KiOutput	FLOAT32	ro	-	FLOAT32	None

7.5.5 Object 0x2312: Kd output

This parameter holds the output of the first differential element of the pressure controller.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2312	0	KdOutput	FLOAT32	ro	-	FLOAT32	None

7.5.6 Object 0x5862: Kd feedback output

This parameter holds the output of the second differential element of the pressure controller.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5862	0	KdFeedbackOutput	FLOAT32	ro	-	FLOAT32	None

7.5.7 Object 0x2418: Controller output

This parameter holds the controller output of the pressure controller.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2418	0	ControllerOutput	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> (0x2350). 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.

Index	Object name
0x2303	Ramp slope
0x230D	Active transducer interface
0x230F	Transducer interface area B
0x2304	Proportional gain
0x230E	Proportional gain time constant
0x2305	Integrator gain
0x2306	Integrator factor
0x2307	Integrator control range
0x231A	Integrator upper output limit
0x231B	Integrator lower output limit
0x5861	Integrator proportional part P gain
0x2308	Differentiator gain
0x2309	Differentiator T1
0x5863	Differentiator gain 2
0x5864	Differentiator T1 2
0x230A	Upper output limit
0x230B	Lower output limit
0x5867	Feed forward gain
0x5868	Feed forward parameter
0x5870	Feed forward offset
0x5865	Upper controller output limit
0x5866	Lower controller output limit
0x586C	pQ switching mode
0x230C	Hydraulic capacity

Table 81: Pressure controller objects contained in a parameter set

The objects which are part of the parameter set are signed with a sub-index N (1…16) in brackets after the index number. E.g. (0x2304[N]).

7.5.8.1 Object 0x2350: Active parameter set number

This parameter is to select the active parameter set of the pressure controller.

The number of this parameter set is identical with the sub-index of the objects which are used for the pressure controller. If, for example, the <ActiveParameterSetNumber> (0x2350) is set to 3 all pressure controller objects within the set use the parameter with the sub-index 3 as active parameter.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2350	0	ActiveParameterSetNumber	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> (0x2303[N]). The ramp is only active, if the parameter is greater zero.
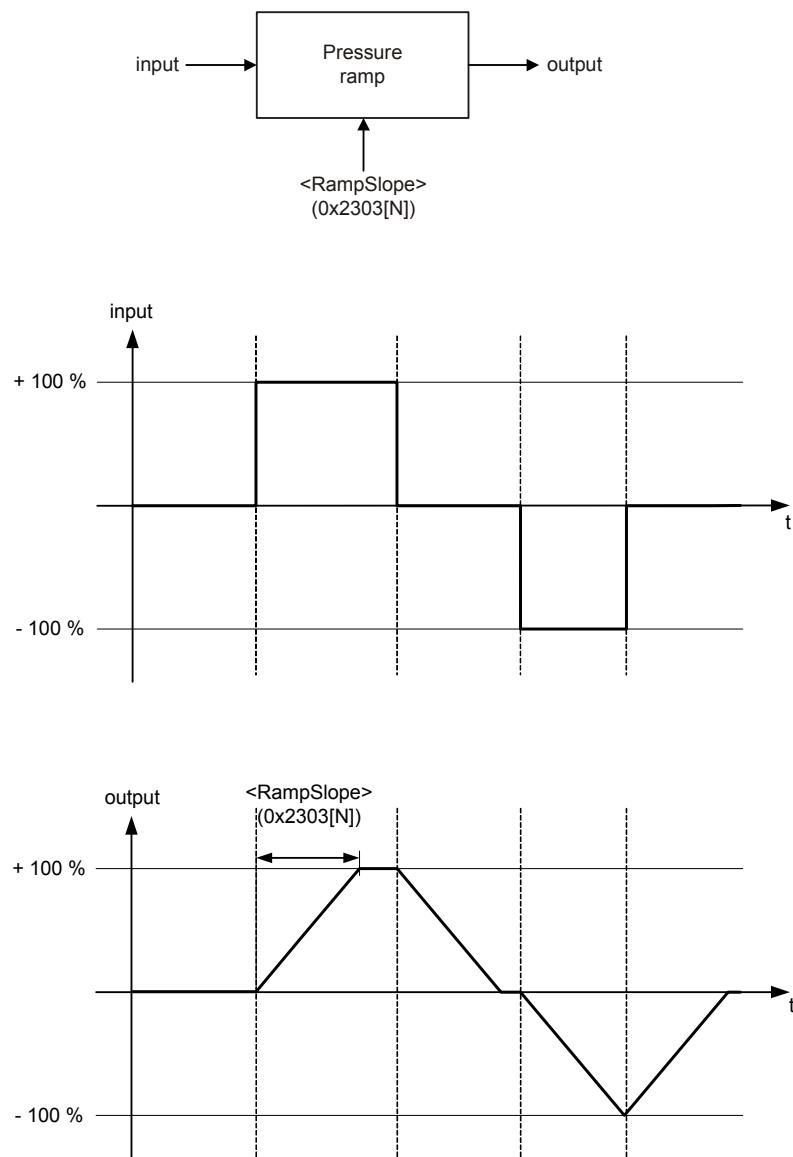
Figure 80: Demand pressure ramp function

7.5.9.1 Object 0x2303[N]: Ramp slope

If a 100 % step is set as input, the ramp output needs <RampSlope> (0x2303) milliseconds to reach the 100 % ramp output.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2303	N=1…16	RampSlope	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> (0x230D[N]) is fed to the pressure controller. The <ActiveTransducerInterfaceAreaB> (0x230F[N]) is set to zero.

- Differential pressure control
  Two pressure signals from the transducer interfaces are fed to the pressure controller.
  One from <ActiveTransducerInterfaceAreaA> (0x230D[N]) for the pressure in the servo valve port A and one from the transducer interface <ActiveTransducerInterfaceAreaB> (0x230F[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> (0x585F), <CylinderRodDiameterA> (0x585D) and <CylinderRodDiameterB> (0x585E) are used.
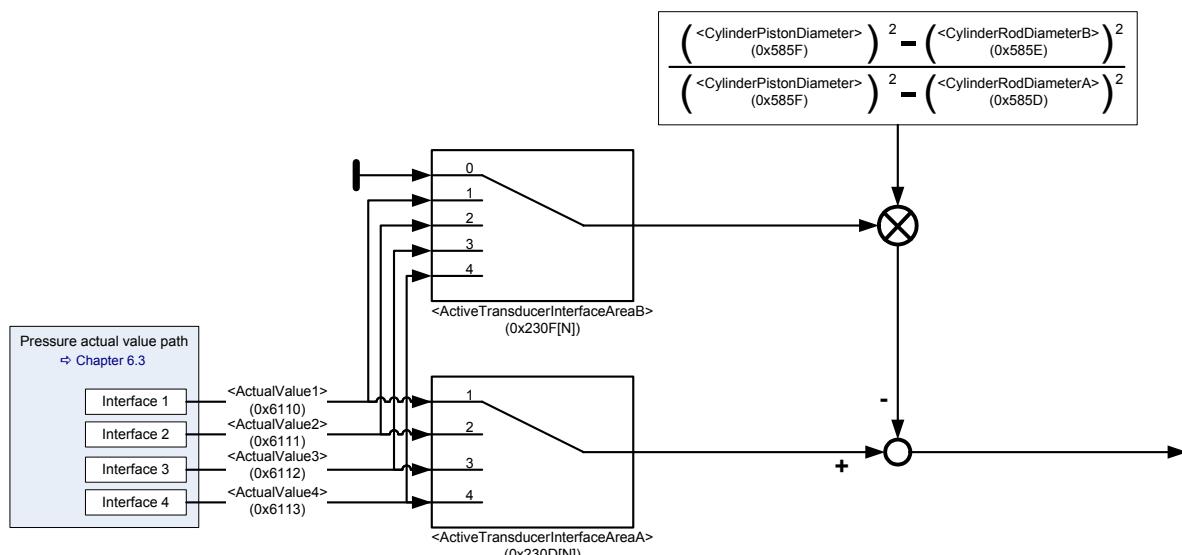
Figure 81: Pressure transducer selection

Set the parameter <ActiveTransducerInterfaceAreaB> (0x230F[N]) to 0 to switch off the differential pressure control.

7.5.10.1 Object 0x230D[N]: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x230D	N=1…16	ActiveTransducerInterfaceAreaA	INT8	rw	Y	1…4	1

7.5.10.2 Object 0x230F[N]: 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> (0x230F[N]) must be set to zero.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x230F	N=1…16	ActiveTransducerInterfaceAreaB	INT8	rw	Y	0…4	None

7.5.10.3 Object 0x585F[N]: 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> (0x585D) and <CylinderRodDiameterB> (0x585E).

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x585F	0	CylinderPistonDiameter	FLOAT32	rw	Y	0.0…+inf	1000000.0

7.5.10.4 Object 0x585D: 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> (0x585F) and <CylinderRodDiameterB> (0x585E).

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x585D	0	CylinderRodDiameterA	FLOAT32	rw	Y	0.0…<CylinderPistonDiameter> (0x585F)	0.0

7.5.10.5 Object 0x585E: 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> (0x585F) and <CylinderRodDiameterA> (0x585D).

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x585E	0	CylinderRodDiameterB	FLOAT32	rw	Y	0.0…<CylinderPistonDiameter> (0x585F)	0.0

7.5.11 Actual value filter

The parameters <ActualPressureFilterCutoffFrequency> (0x23F2) and <ActualPressureFilterOrder> (0x23F3) are used to set the behavior of the Butterworth filter. <ActualPressureFilterCutoffFrequency> (0x23F2) specifies the cutoff frequency of the filter in Hz. The order of the filter is set with the parameter <ActualPressureFilterOrder> (0x23F3) (possible values: 1…3).

<ActualPressureFilterCutoffFrequency>
(0x23F2)
<ActualPressureFilterOrder>
(0x23F3)
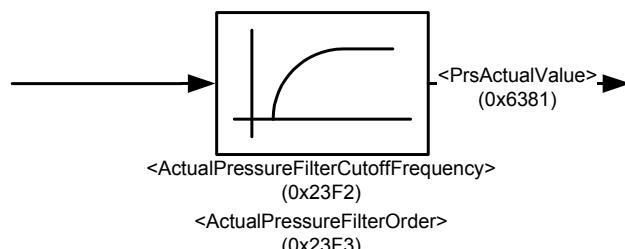
Figure 82: Actual value filter

7.5.11.1 Object 0x23F2: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x23F2	0	ActualPressureFilterCutoffFrequency	FLOAT32	rw	Y	0.0, 10.0…10000.0/3.0	None

7.5.11.2 Object 0x23F3: Actual pressure filter order

This parameter sets the order of the Butterworth filter.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x23F3	0	ActualPressureFilterOrder	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.
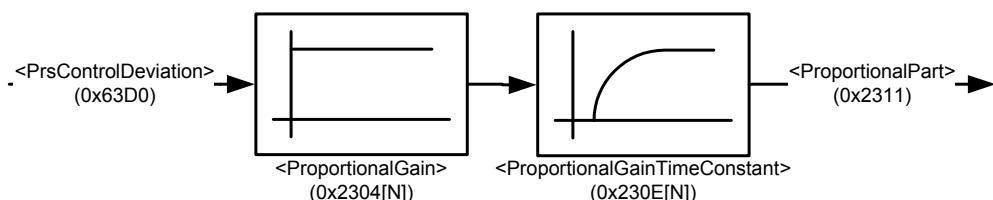
Figure 83: Proportional first order lag element (PPT1)

7.5.12.1 Object 0x2304[N]: Proportional Gain

This parameter sets the proportional gain.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2304	N=1…16	ProportionalGain	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.12.2 Object 0x230E[N]: Proportional gain time constant

This parameter sets the time constant in seconds of the first order lag element (PT1).

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x230E	N=1…16	ProportionalGainTimeConstant	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.
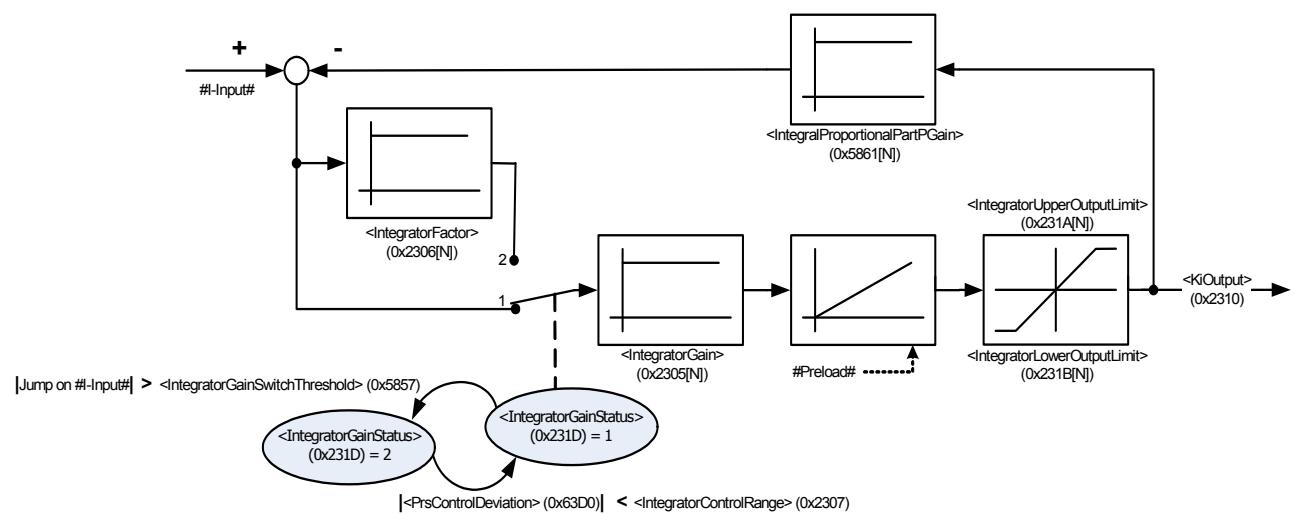
Figure 84: Integrator element (I)

7.5.13.1 Object 0x2305[N]: Integrator gain

This parameter contains the integrator gain.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x2305	N=1…16	IntegratorGain	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.13.2 Object 0x2306[N]: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x2306	N=1…16	IntegratorFactor	FLOAT32	rw	Y	0.0…+inf	0.1

7.5.13.3 Object 0x2307[N]: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x2307	N=1…16	IntegratorControlRange	INT16	rw	Y	0…16384	163

7.5.13.4 Object 0x231D: Integrator gain status

This parameter shows the state of the switch whether the <IntegratorFactor> (0x2306) is multiplied (value = 2) or not (value = 1).

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x231D	0	IntegratorGainStatus	UINT8	r	Y	1…2	1

7.5.13.5 Object 0x5857[N]: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5857	N=1…16	IntegratorGainSwitchThreshold	UINT32	rw	Y	UINT32	5000

7.5.13.6 Object 0x231A[N]: Integrator upper output limit

This parameter contains the upper limit of the integrator output.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x231A	N=1…16	IntegratorUpperOutputLimit	INT16	rw	Y	<IntegratorLowerOutputLimit> (0x231B[N])…32767	16384

7.5.13.7 Object 0x231B[N]: Integrator lower output limit

This parameter contains the lower limit of the integrator output.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x231B	N=1…16	IntegratorLowerOutputLimit	INT16	rw	Y	−32768… <IntegratorUpperOutputLimit> (0x231A[N])	−16384

7.5.13.8 Object 0x5861[N]: Integrator proportional part P gain

Using this feedback proportional gain, the integrator can be modified to a first order lag element.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5861	N=1…16	IntegralProportionalPartPGain	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> (0x586A).

If the <IntegratorPreloadMode> (0x586B) 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> (0x586B) 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.
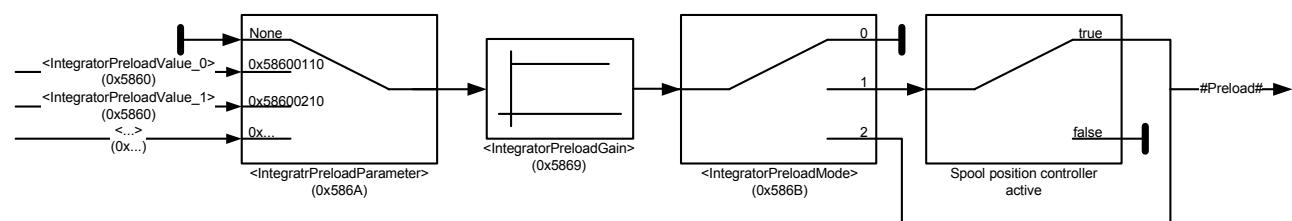
Figure 85: Integrator preload value

7.5.14.1 Object 0x586B: Integrator preload mode

This parameter is to select the integrator preload mode.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x586B	0	IntegratorPreloadMode	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 82: Possible values of parameter <IntegratorPreloadMode> (0x586B)

---

[1] Former value means the value of the <IntegratorPreloadMode> before setting the value to 2 (usually to default value 1).

7.5.14.2 Object 0x5869: Integrator preload gain

This parameter contains the integrator preload gain.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5869	0	IntegratorPreloadGain	FLOAT32	rw	Y	FLOAT32	0.0

7.5.14.3 Object 0x586A: Integrator preload parameter

With this parameter every INT16 application parameter can be mapped as preload input. Per default the pressure setpoint <PrsSetpoint> (0x6380) is mapped.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x586A	0	IntegratorPreloadParameter	UINT32	rw	Y	UINT32	0x63800110

Value description

<Parameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	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> (0x6380), sub-index 0x01 with a length of 16 bit (16=0x10).

7.5.14.4 Object 0x5860: Integrator preload values

This object contains two pre-calculated preload values.
These values can be mapped using the parameter <IntegratorPreloadParameter> (0x586A) to the integrator preload input.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5860	1	IntegratorPreloadValue_0	INT16	ro	-	INT16	None
0x5860	2	IntegratorPreloadValue_1	INT16	ro	-	INT16	None

Value description

<Parameter>	Behavior of preload output
<IntegratorPreloadValue1>	<SplDemandValue> (0x6310) minus <ProportionalPart> (0x2311)
<IntegratorPreloadValue2>	<SplDemandValue> (0x6310) minus <ProportionalPart> (0x2311) minus <FeedForwardOffset> (0x5870[N])

Table 83: Behavior of preload output

7.5.15 Derivative element (PD)

This element differentiates the pressure actual value <PrsActualValue> (0x6381) with a differentiator element including a first order filter with a time constant T1. Also a proportional gain element is implemented.
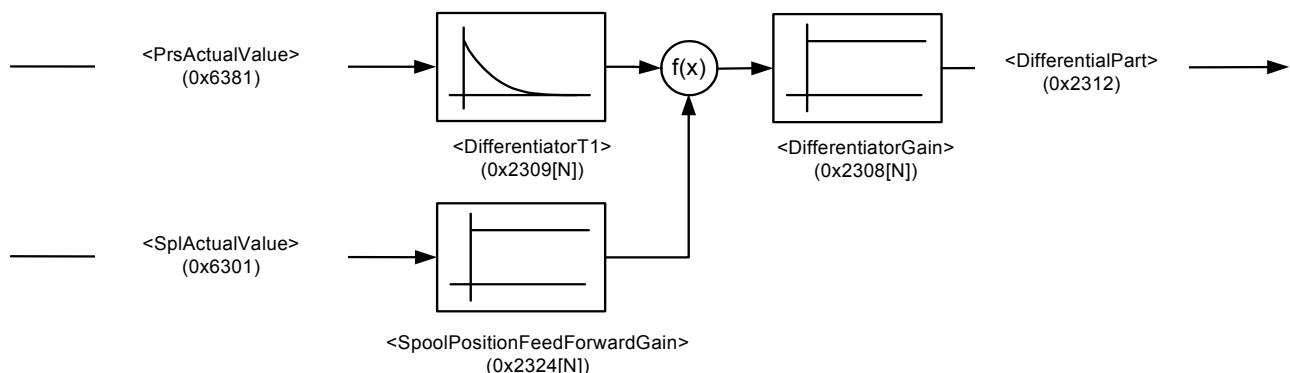
Figure 86: Proportional derivative element 1 (PD)

7.5.15.1 Object 0x2308[N]: Differentiator gain

This parameter contains the gain of the first differentiator.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2308	N=1…16	DifferentiatorGain	FLOAT32	rw	Y	–inf…+inf	0.0

7.5.15.2 Object 0x2309[N]: Differentiator T1

This parameter contains the time constant of the first differentiator in seconds.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2309	N=1…16	DifferentiatorT1	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.15.3 Object 0x2324[N]: Spool Position Feed Forward Gain

The gain controls the influence of the spool position to the derivative element (PD).

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2324	N=1…16	SpoolPositionFeedForwardGain	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.16 Feedback derivative element (PD)

This element differentiates the pressure actual value <PrsActualValue> (0x6381) with a real differentiator element including a first order filter with a time constant T1. A gain element is below this differentiator.
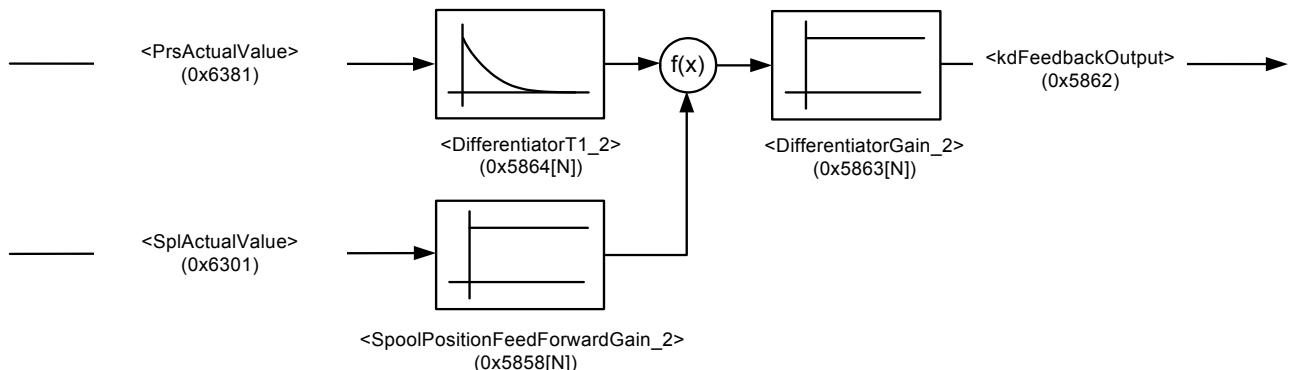
Figure 87: Proportional derivative element 2 (PD)

7.5.16.1 Object 0x5863[N]: Differentiator gain 2

This parameter contains the gain of the second differentiator.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5863	N=1…16	DifferentiatorGain_2	FLOAT32	rw	Y	–inf…+inf	0.0

7.5.16.2 Object 0x5864[N]: Differentiator T1 2

This parameter contains the time constant of the first differentiator in seconds.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5864	N=1…16	DifferentiatorT1_2	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.16.3 Object 0x5858N]: Spool Position Feed Forward Gain_2

This gain controls the influence of the spool position to the feedback derivative element (PD).

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5858	N=1…16	SpoolPositionFeedForwardGain_2	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> (0x6390) is greater than zero.

⇨ Chapter "7.5.10 Pressure transducer selection", page 172
⇨ Chapter "7.5.10.3 Object 0x585F[N]: Cylinder piston diameter", page 173
⇨ Chapter "7.5.10.4 Object 0x585D: Cylinder rod diameter A", page 173
⇨ Chapter "7.5.10.5 Object 0x585E: Cylinder rod diameter B", page 174

$$\frac{\left(\dfrac{\text{<CylinderPistonDiameter>}}{(0x585F)}\right)^2 - \left(\dfrac{\text{<CylinderRodDiameterB>}}{(0x585E)}\right)^2}{\left(\dfrac{\text{<CylinderPistonDiameter>}}{(0x585F)}\right)^2 - \left(\dfrac{\text{<CylinderRodDiameterA>}}{(0x585D)}\right)^2}$$
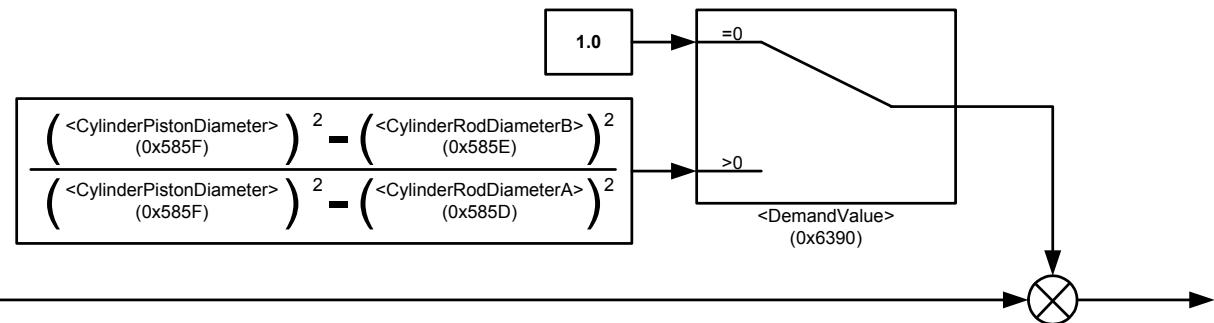
Figure 88: Alpha correction

7.5.18 Signal limitation 1

Signal limitation after the alpha correction.

<UpperOutputLimit>
(0x230A[N])

<LowerOutputLimit>
(0x230B[N])
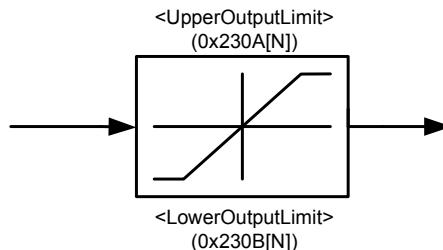
Figure 89: Signal limitation 1

7.5.18.1 Object 0x230A[N]: Upper output limit

This parameter contains the upper limit of the limiter.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x230A	N=1…16	UpperOutputLimit	INT16	rw	Y	<LowerOutputLimit> (0x230B)…32767	16384

7.5.18.2 Object 0x230B[N]: Lower output limit

This parameter contains the lower limit of the limiter.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x230B	N=1…16	LowerOutputLimit	INT16	rw	Y	32767… <UpperOutputLimit> (0x230A)	−16384

7.5.19 Feed forward

For some applications a feed forward control is advantageous. With the feed forward function a selectable sig-nal, 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.
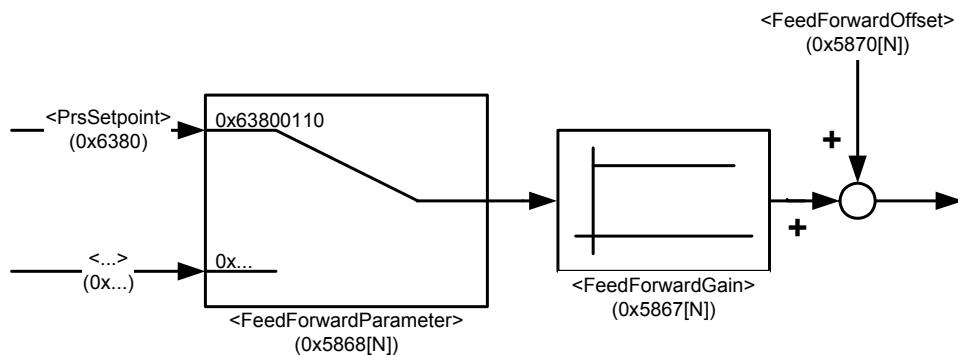
Figure 90: Feed forward

7.5.19.1 Object 0x5867[N]: Feed forward gain

This parameter contains the feed forward gain.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5867	N=1…16	FeedForwardGain	FLOAT32	rw	Y	FLOAT32	0.0

7.5.19.2 Object 0x5870[N]: Feed forward offset

This parameter contains the feed forward offset.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5870	N=1…16	FeedForwardOffset	INT16	rw	Y	INT16	0

7.5.19.3 Object 0x5868[N]: Feed forward parameter

With this parameter the source signal of the feed forward block will be selected. As default the pressure set-point value <PrsSetpoint> (0x6380) is mapped.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5868	N=1…16	FeedForwardParameter	UINT32	rw	Y	UINT32	0x63800110

Value description

<Parameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	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> (0x6380), sub-index 0x01 with a length of 16 bit (16=0x10).

7.5.20 Signal limitation 2

This block limits the controller output.
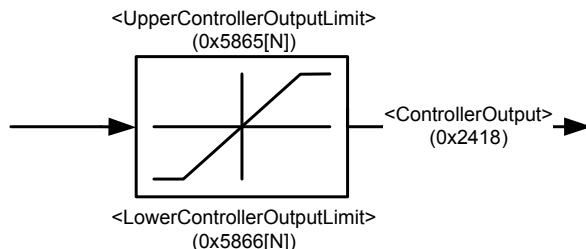
Figure 91: Signal limitation 2

7.5.20.1 Object 0x5865[N]: Upper controller output limit

This parameter contains the upper limit of the limitation.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5865	N=1…16	UpperControllerOutputLimit	INT16	rw	Y	<LowerControllerOutput-Limit> (0x5866[N])…32767	16384

7.5.20.2 Object 0x5866[N]: Lower controller output limit

This parameter contains the lower limit of the limitation.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5866	N=1…16	LowerControllerOutputLimit	INT16	rw	Y	32767… <UpperControllerOutput-Limit> (0x5865[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]
$C_H = \dfrac{V}{E_{Oil}}$	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 84: Parameters used in a linear plant model
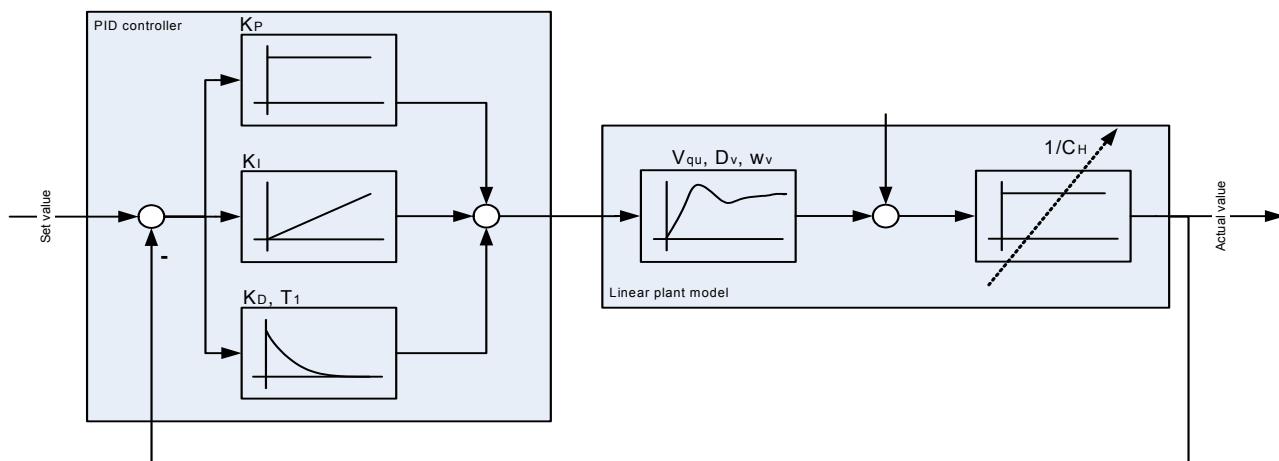
Figure 92: Parameterization of the pressure controller

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> (0x231C) must be set for the used pressure sensor interface. The <HydraulicCapacity> (0x230C) 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> (0x230C) to zero turns off the automatic parameter calculation.

7.5.21.1 Object 0x230C[N]: Hydraulic capacity

The parameter <HydraulicCapacity> (0x230C) is defined as $\dfrac{V}{E_{Oil}}$. The unit is defined as [10^{-6} l/bar].

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x230C	N=1…16	HydraulicCapacity	FLOAT32	rw	Y	0.0…+inf	0.0

7.5.21.2 Object 0x231C: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x231C	1	HydraulicCapacity	INT16	rw	N	INT16	400
0x231C	2	Unit	UINT8	ro	-	UINT8	0
0x231C	3	Prefix	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> (0x6043) 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 0x6043: Control mode", page 128
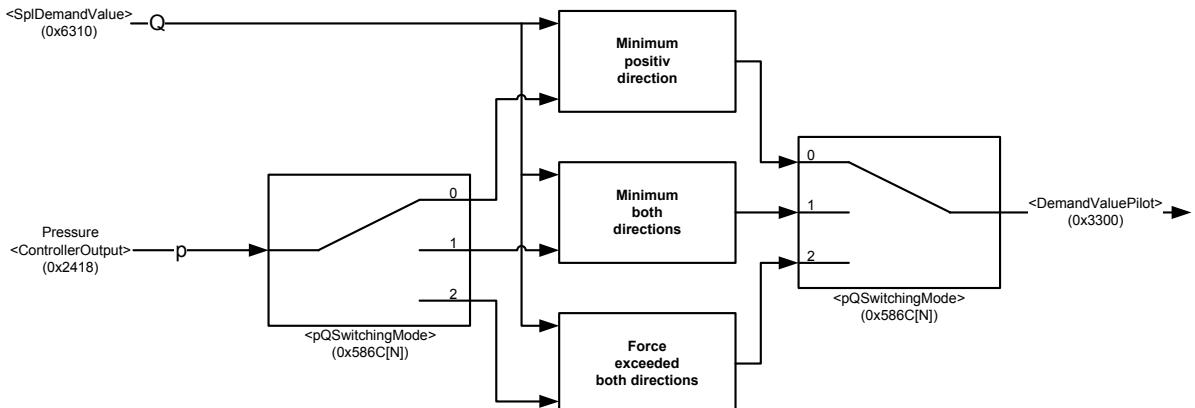
Figure 93: Spool position (Q) / pressure (P) switchover

7.6.1 Object 0x586C[N]: pQ switching mode

This parameter defines the p/Q switching mode.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x586C	N=1…16	pQSwitchingMode	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 85: Possible values of parameter <pQSwitchingMode> (0x586C)

7.6.1.1 Object 0x3300: Demand value pilot

This parameter contains the output signal of the p/Q switchover function.

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3300	0	DemandValuePilot	INT16	ro	-	INT16	None

7.6.2 Minimum criterion in positive direction (switching mode 0)

If the <pQSwitchingMode> (0x586C) 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> (0x3300) is used instead of <SplDemandValue> (0x6310).
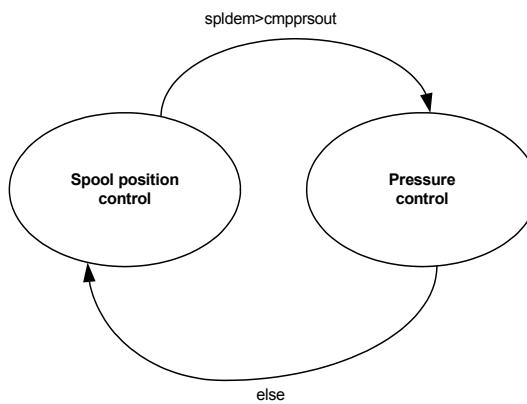
Figure 94: State machine used to switch between spool position control and pressure control

Value	Parameter
spldem	<SplDemandValue> (0x6310)
cmpprsout	<ControllerOutput> (0x2418)

7.6.3 Minimum criterion in both directions (switching mode 1)

If the <pQSwitchingMode> (0x586C) is set to 1 (minimum criterion in both directions), the following state machine is used to switch between spool position control and pressure control.
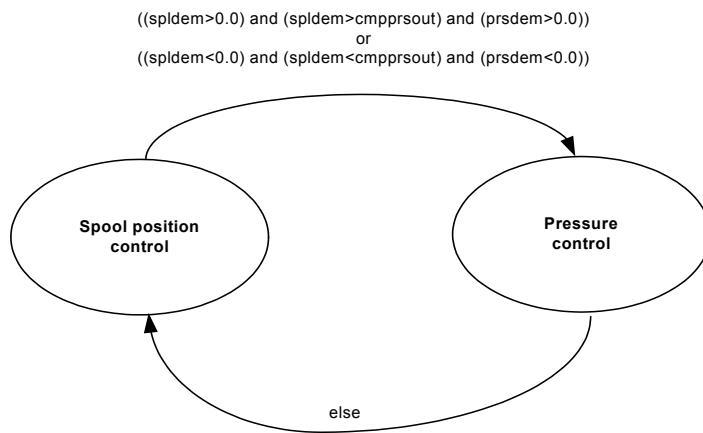
Figure 95: State machine used to switch between spool position control and pressure control

Value	Parameter
spldem	<SplDemandValue> (0x6310)
prsdem	<PrsDemandValue> (0x6390)
cmpprsout	<ControllerOutput> (0x2418)

7.6.4 Force exceeded in both directions (switching mode 2)

If the <pQSwitchingMode> (0x586C) is set to 2 (force exceeded in both directions), the following state machine is used to switch between spool position control and pressure control.
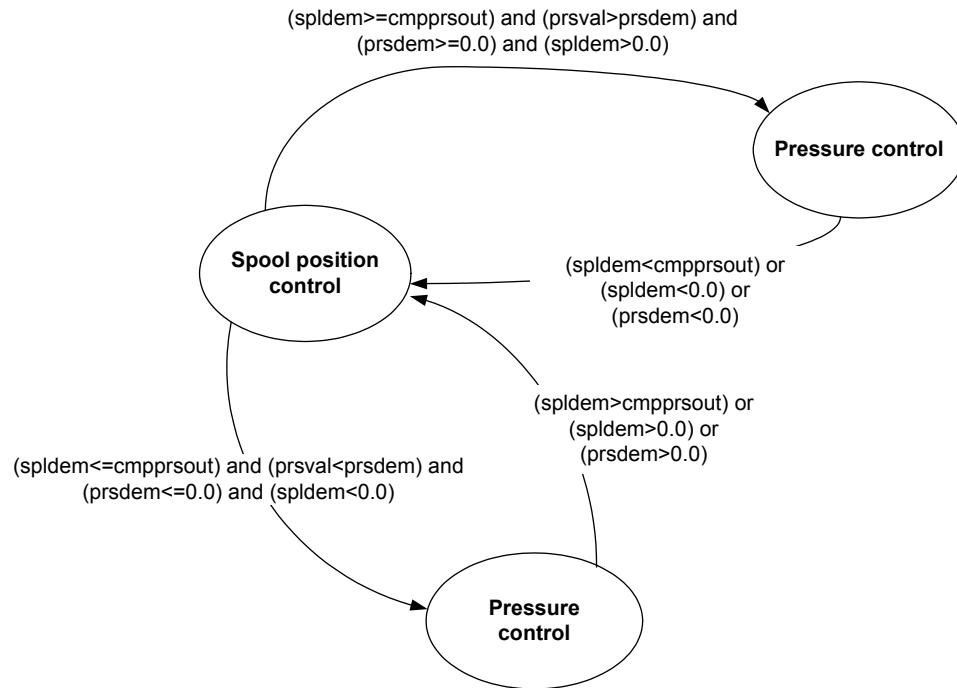
Figure 96: State machine used to switch between spool position control and pressure control

Value	Parameter
spldem	<SplDemandValue> (0x6310)
prsdem	<PrsDemandValue> (0x6390)
prsval	<PrsActualValue> (0x6381)
cmpprsout	<ControllerOutput> (0x2418)

7.7 Axis position setpoint conditioning / demand value generator
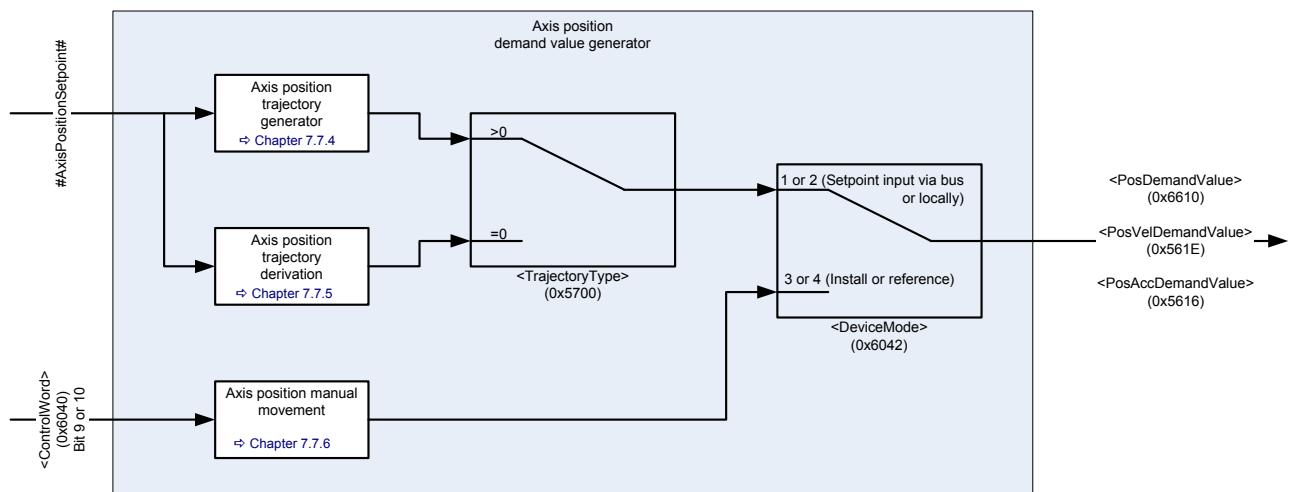
Figure 97: Axis position setpoint conditioning / demand value generator

> ℹ️ 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 63

7.7.1 Object 0x6610: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6610	1	PosDemandValue	INT32	ro	-	INT32	None
0x6610	2	Unit	UINT8	ro	-	UINT8	0
0x6610	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.7.2 Object 0x561E: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x561E	0	PosVelDemandValue	INT32	ro	-	INT32	None

7.7.3 Object 0x5616: 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

Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5616	0	PosAccDemandValue	INT32	ro	-	INT32	None

7.7.4 Axis position trajectory generator

The trajectory generator is active, if the <TrajectoryType> (0x5700) = 1 or 2. Then the trajectory generator allows a smooth <PosDemandValue> (0x6610) if the <PosSetpoint> (0x6600) jumps to another position.
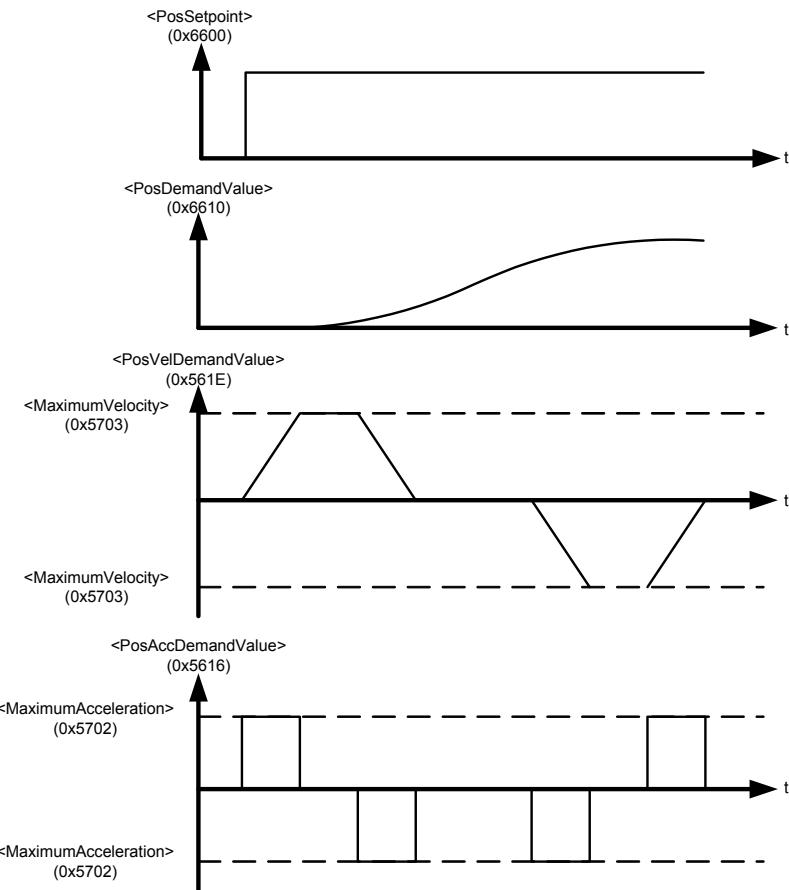
Figure 98: Axis position trajectory generator

The trajectory generator is switched off, if the <TrajectoryType> (0x5700) = 0.
In this case the <PosDemandValue> (0x6610) is equal to the <PosSetpoint> (0x6600).
The <PosVelDemandValue> (0x561E) is the derivation of the <PosDemandValue> (0x6610).
The <PosAccDemandValue> (0x5616) is the derivation of the <PosVelDemandValue> (0x561E).
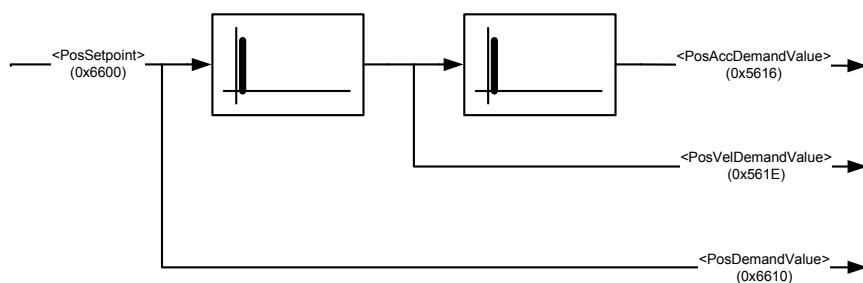
Figure 99: Axis position derivation

7.7.4.1 Object 0x5700: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5700	0	TractoryType	INT8	rw	Y	0…2	0

7.7.4.2 Object 0x5703: Maximum velocity

This parameter limits the maximal velocity of the axis.

DrivePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5703	0	MaximumVelocity	INT32	rw	Y	0…2147483647	100000

7.7.4.3 Object 0x5702: Maximum acceleration

This parameter limits the maximal acceleration of the axis.

DrivePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5702	0	MaximumAcceleration	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.
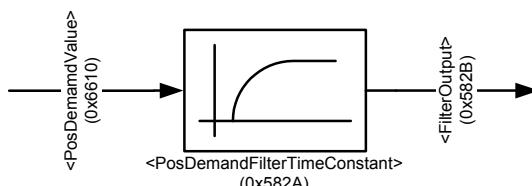
Figure 100: Axis position demand value filter

7.7.5.1 Object 0x582A: Demand value filter time constant

This parameter sets the time constant in seconds of the first order lag element (PT1).

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x582A	0	PosDemandFilterTimeConstant	FLOAT32	rw	Y	0…+inf	0.0

7.7.6 Axis position manual movement

In the <DeviceMode> (0x6042) = 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> (0x6040). If an incremental encoder is used, then the <DeviceMode> (0x6042) = 4 (reference mode) allows a homing position run using the Z pulse of the encoder. The maximal velocity is limited with the parameter <ManualMaxSpeed> (0x5614).

⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 58
⇨ Chapter "5.1.2 Object 0x6040: Control word", page 44
⇨ Chapter "6.6.4 Incremental Encoder", page 96
⇨ Chapter "7.7 Axis position setpoint conditioning / demand value generator", page 191

7.7.6.1 Object 0x5614: Manual maximum speed

This maximum speed limits the manual velocity in <DeviceMode> (0x6042) = 3 (install mode) or = 4 (reference mode).

DrivePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5614	0	ManualMaxSpeed	INT32	rw	Y	1000...2147483647	1000

7.8 Axis position controller
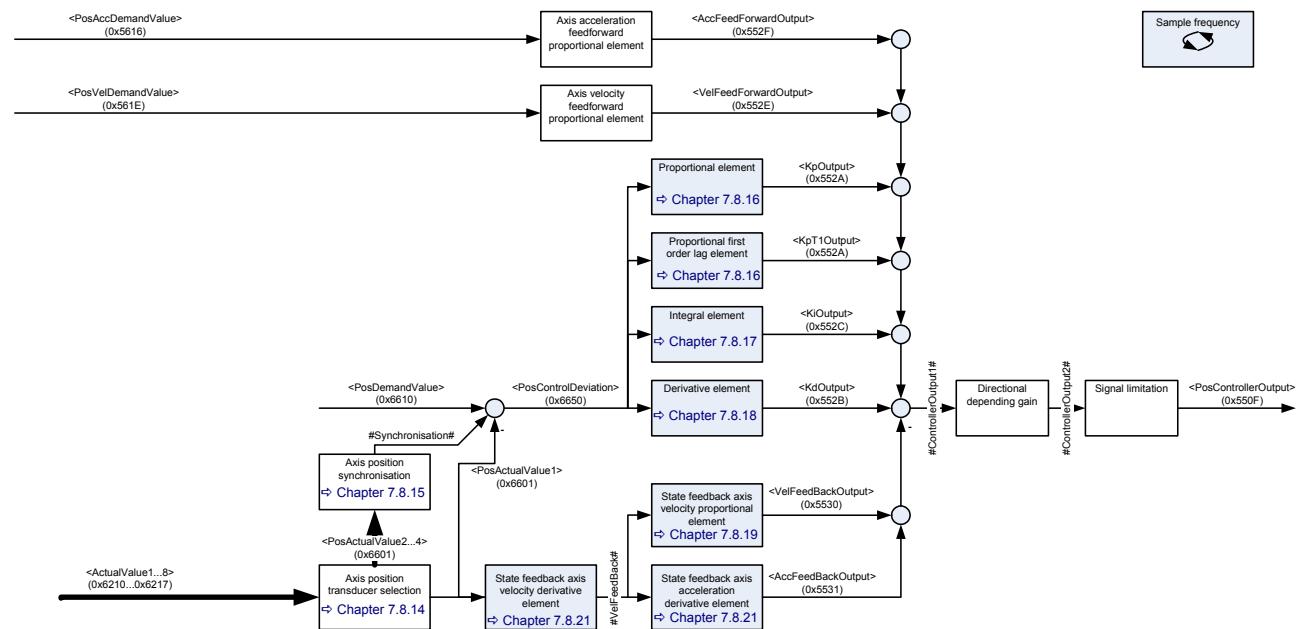
Figure 101: Axis position controller

7.8.1 Object 0x6601: Actual value

The actual position of the position controller has to be mapped to the parameter <PosActualValue1> (0x6601). 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> (0x6601). The mapping of the actual positions to the interface outputs is done with the transducer interface numbers <PosActualValueReference1...4> (0x6602).

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6601	1	PosActualValue1	INT32	ro	-	INT32	None
0x6601	2	PosActualValue2	INT32	ro	-	INT32	None
0x6601	3	PosActualValue3	INT32	ro	-	INT32	None
0x6601	4	PosActualValue4	INT32	ro	-	INT32	None

7.8.2 Object 0x582B: Demand filter output

This parameter holds the output value of the demand filter element.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x582B	0	PosDemandFilterOutput	FLOAT32	ro	-	FLOAT32	None

7.8.3 Object 0x6650: Control deviation

This parameter holds the deviation between the axis position demand filter value and the axis position actual value.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6650	1	PosControlDeviation	INT32	ro	-	INT32	None

7.8.4 Object 0x552A: Kp output

This parameter holds the output of the proportional element of the axis position controller.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x552A	1	KpOutput	FLOAT32	ro	-	FLOAT32	None

7.8.5 Object 0x552A: Kp T1 output

This parameter holds the output of the proportional first order lag element of the axis position controller.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x552A	2	KpT1Output	FLOAT32	ro	-	FLOAT32	None

7.8.6 Object 0x552C: Ki output

This parameter holds the output of the integral element of the axis position controller.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x552C	1	KiOutput	FLOAT32	ro	-	FLOAT32	None

7.8.7 Object 0x552B: Kd output

This parameter holds the output of the first differential element of the axis position controller.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x552B	1	KdOutput	FLOAT32	ro	-	FLOAT32	None

7.8.8 Object 0x552E: Velocity feedforward output

This parameter holds the output of the axis velocity feedforward proportional element of the axis position controller.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x552E	1	VelFeedForwardOutput	FLOAT32	ro	-	FLOAT32	None

7.8.9 Object 0x552F: Acceleration feedforward output

This parameter holds the output of the axis acceleration feedforward proportional element of the axis position controller.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x552F	1	AccFeedForwardOutput	FLOAT32	ro	-	FLOAT32	None

7.8.10 Object 0x5530: Velocity feedback output

This parameter holds the output of the axis velocity feedback proportional element of the axis position controller.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5530	1	VelFeedBackOutput	FLOAT32	ro	-	FLOAT32	None

7.8.11 Object 0x5531: Acceleration feedback output

This parameter holds the output of the axis acceleration feedback proportional element of the axis position controller.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5531	1	AccFeedBackOutput	FLOAT32	ro	-	FLOAT32	None

7.8.12 Object 0x550F: Controller output

This parameter holds the controller output of the axis position controller.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x550F	1	PosControllerOutput	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> (0x553F).

7.8.13.1 Object 0x553F: Sample frequency divider

This parameter holds the sample frequency divider.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x553F	0	PosSampleFrequencyDivider	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> (6602) is used to select the corresponding interfaces. The actual axis position of the controlled axis is expected on <PosActualValue1> (6601). The positions of the synchronised axis are expected on <PosActualValue2…4> (6601).
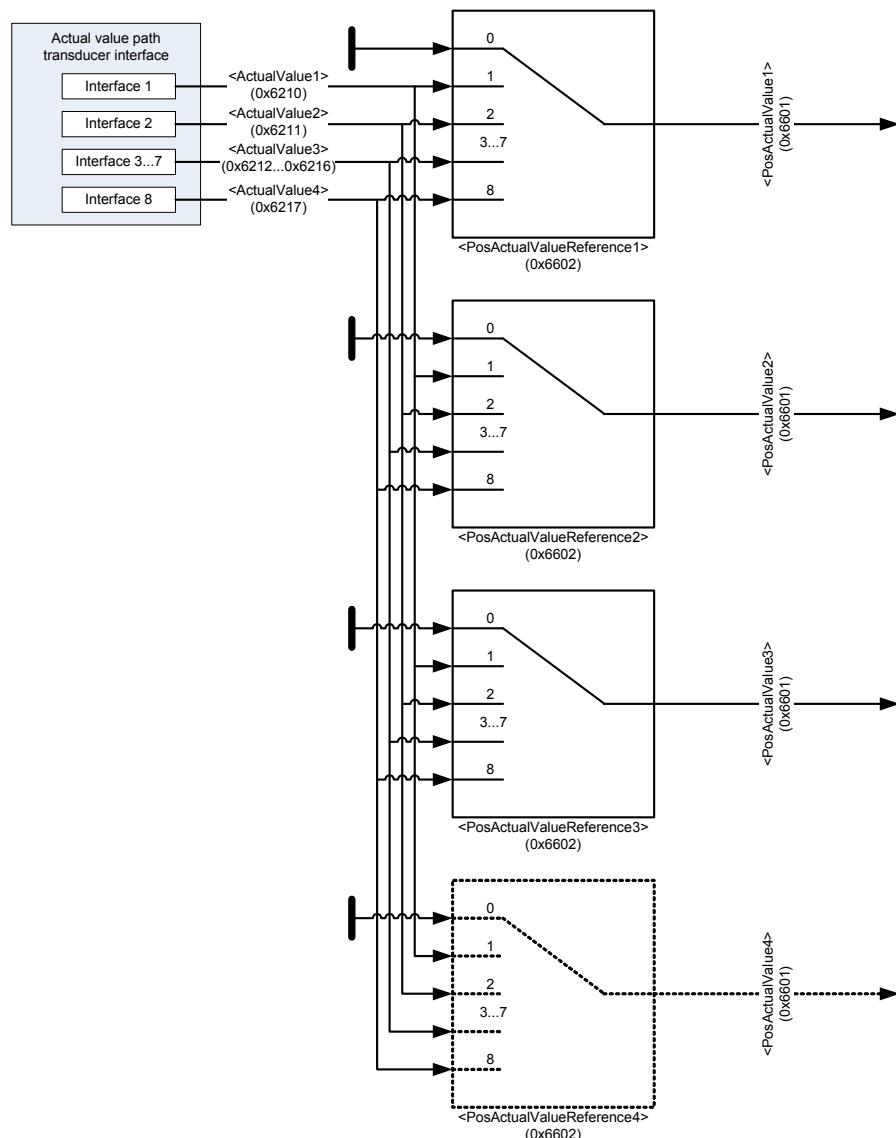
Figure 102: Axis position transducer selection

7.8.14.1 Object 0x6602: Actual value reference

This object allows to map four transducer interface inputs to the actual axis position inputs for the axis controller.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6602	1	PosActualValueReference1	UINT8	rw	Y	0…8	1
0x6602	2	PosActualValueReference2	UINT8	rw	Y	0…8	0
0x6602	3	PosActualValueReference3	UINT8	rw	Y	0…8	0
0x6602	4	PosActualValueReference4	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.
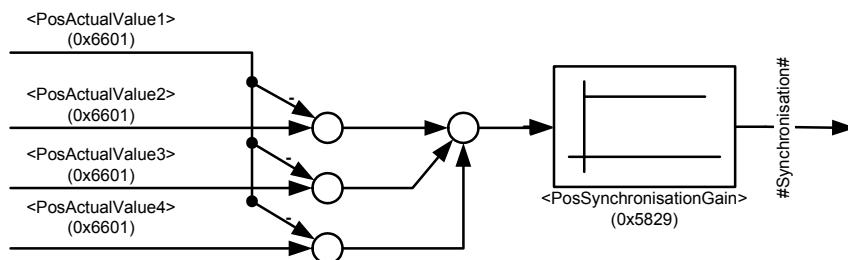
Figure 103: Axis position synchronisation

7.8.15.1 Object 0x5829: Synchronisation gain

This parameter holds the gain of the proportional element of the axis position synchronisation. Zero will switch off the synchronous function.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5829	0	PosSynchronisationGain	FLOAT32	rw	Y	FLOAT32	0.0

7.8.15.2 Object 0x5830: Number of axis to synchronize

This parameter holds the number of axis which should be synchronous.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5830	0	NumberOfAxisToSynchronize	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> (0x5501) plus the first order lag element with the gain <ProportionalGainKpT1> (0x5501) and the time constant <TimeConstantKpT> (0x5508). The input signal is the axis position control deviation.
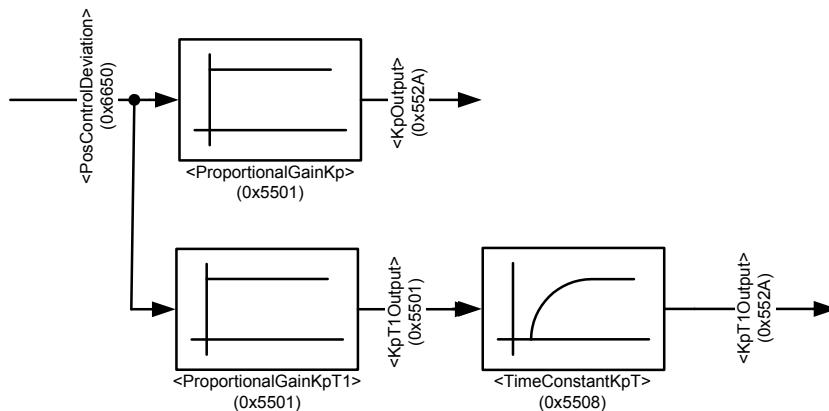
Figure 104: Proportional element with first order lag element

7.8.16.1 Object 0x5501: Proportional gain

This object contains two proportional gain parameters. <ProportionalGainKp> (0x5501) is the gain of the proportional part and <ProportionalGainKpT1> (0x5501) is the gain of the first order lag element.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5501	1	ProportionalGainKp	FLOAT32	rw	Y	FLOAT32	0.0
0x5501	2	ProportionalGainKpT1	FLOAT32	rw	Y	FLOAT32	0.0

7.8.16.2 Object 0x5508: Time constant

This parameter sets the time constant in seconds of the first order lag element (PT1).

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5508	1	TimeConstantKpT	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.
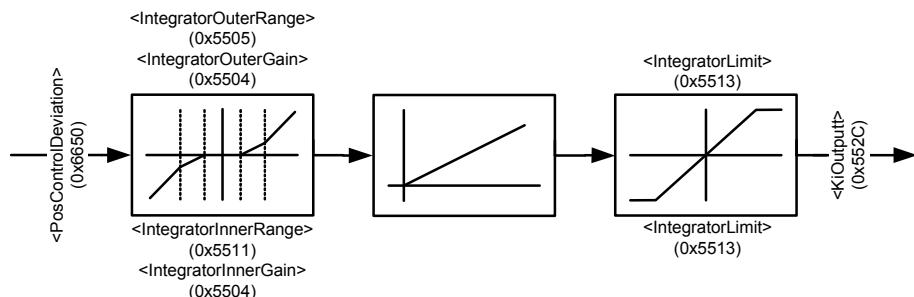
Figure 105: Integral element

7.8.17.1 Object 0x5504: Integral gain

The gain of the integral part depends on the value of the absolute input signal |Input|.

Zero range: |Input| less than <IntegralInnerRange> (0x5511). Output = Zero.
Inner range: |Input| between <IntegralInnerRange> (0x5511) and <IntegralOuterRange> (0x5505).

Output = <IntegralInnerGain> (0x5504) x Input

Outer range: |Input| greater than <IntegralOuterRange> (0x5505).

Output = <IntegralOuterGain> (0x5504) x Input

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5504	1	IntegralInnerGain	FLOAT32	rw	Y	FLOAT32	0.0
0x5504	2	IntegralOuterGain	FLOAT32	rw	Y	FLOAT32	0.0

7.8.17.2 Object 0x5511: Integral inner range

Border of the absolute input signal to switch between gain = 0.0 and gain = <IntegralInnerGain> (0x5504).

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5511	1	IntegralInnerRange	FLOAT32	rw	Y	0.0…2147483647	0.0

7.8.17.3 Object 0x5505: Integral outer range

Border of the absolute input signal to switch between gain = <IntegralInnerGain> (0x5504) and
gain = <IntegralOuterGain> (0x5504).

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5505	1	IntegralOuterRange	FLOAT32	rw	Y	0.0…2147483647	0.0

7.8.17.4 Object 0x5513: Integral limit

Limit of the output signal of the integral element.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5513	1	IntegralLimit	FLOAT32	rw	Y	0…+inf	16384000000

7.8.18 Derivative element
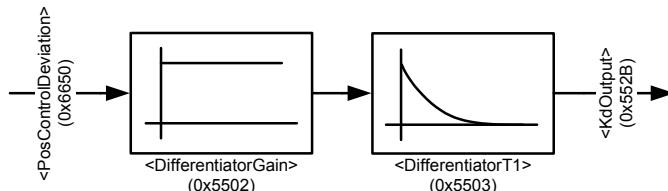
Figure 106: Derivative element

7.8.18.1 Object 0x5502: Differentiator gain

This parameter contains the gain of the first differentiator.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5502	1	DifferentiatorGain	FLOAT32	rw	Y	0…+inf	0.0

7.8.18.2 Object 0x5503: Differentiator T1

This parameter contains the time constant of the first differentiator in seconds.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5503	1	DifferentiatorT1	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.
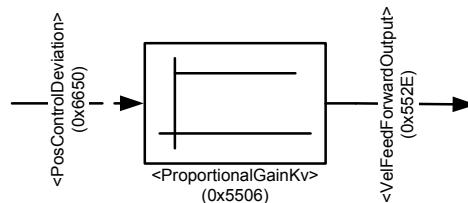
Figure 107: Axis velocity feed forward proportional element

7.8.19.1 Object 0x5506: Velocity feed forward proportional gain

This parameter contains the velocity feed forward gain.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5506	1	ProportionalGainKv	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.
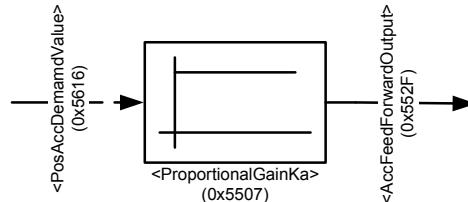
Figure 108: Axis acceleration feed forward proportional element

7.8.20.1 Object 0x5507: Acceleration feed forward proportional gain

This parameter contains the velocity feed forward gain.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5507	1	ProportionalGainKa	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.
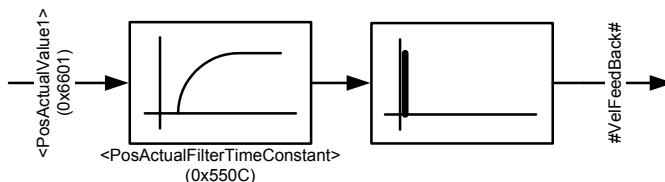
Figure 109: State feedback axis velocity and acceleration derivative element

7.8.21.1 Object 0x550C: Actual value filter time constant

This parameter sets the time constant in seconds of the first order lag filter (PT1).

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x550C	1	PosActualFilterTimeConstant	FLOAT32	rw	Y	0.0…+inf	0.0

7.8.21.2 Object 0x550B: 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.
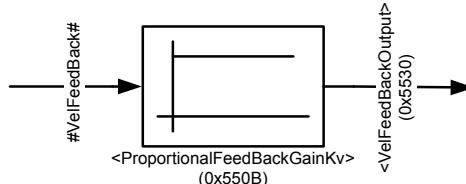
Figure 110: State feedback axis velocity proportional gain

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x550B	1	ProportionalFeedBackGainKv	FLOAT32	rw	Y	FLOAT32	0.0

7.8.21.3 Object 0x550A: 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> (0x550A) will be added to the other parts of the controller.
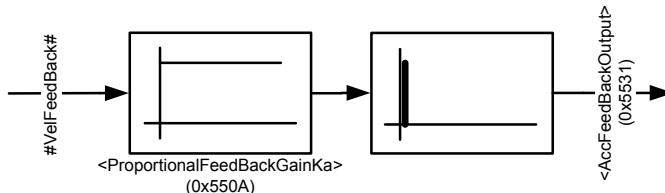
Figure 111: State feedback axis acceleration proportional gain

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x550A	1	ProportionalFeedBackGainKa	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.
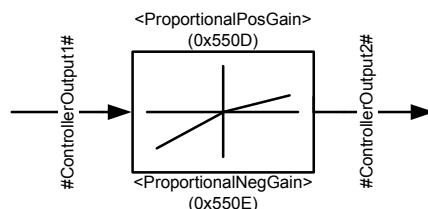
Figure 112: Directional depending gain

7.8.22.1 Object 0x550D: Directional depending positive proportional gain

This gain is used while the input signal is positive.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x550D	1	ProportionalPosGain	FLOAT32	rw	Y	FLOAT32	1.0

7.8.22.2 Object 0x550E: Directional depending negative proportional gain

This gain is used while the input signal is negative.

DrivePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x550E	1	ProportionalNegGain	FLOAT32	rw	Y	FLOAT32	1.0

7.8.23 Signal limitation

The output of the controller is limited to ±16384 (±100 %).
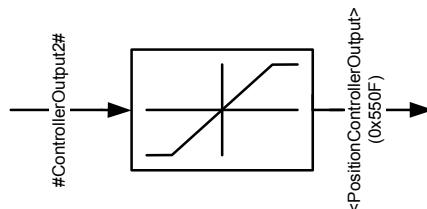
Figure 113: Signal limitation

7.9 Axis velocity setpoint conditioning / demand value generator
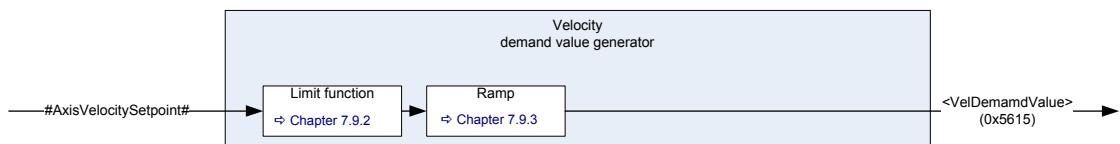
Figure 114: Axis velocity setpoint conditioning / demand value generator

> 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 67

7.9.1 Object 0x5615: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5615	1	VelDemandValue	INT32	ro	N	INT32	None
0x5615	2	Unit	UINT8	ro	-	UINT8	0
0x5615	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

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.
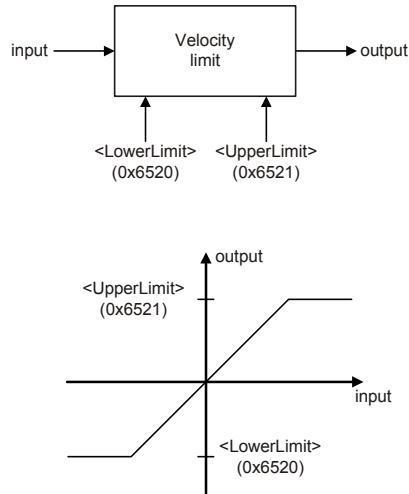
Figure 115: Limit function

ⓘ The <UpperLimit> (0x6520) must be greater than the <LowerLimit> (0x6521).
If <LowerLimit> (0x6521) will be set greater than the <UpperLimit> (0x6520), the <UpperLimit> (0x6520) will be set to the value of the <LowerLimit> (0x6521).

7.9.2.1 Object 0x6521: Upper limit

ValvePositionControl_DemandValueGenerator_Limit							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6521	1	UpperLimit	INT16	rw	Y	<LowerLimit> (0x6520)… 32767	16384

7.9.2.2 Object 0x6520: Lower limit

ValvePositionControl_DemandValueGenerator_Limit							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6520	1	LowerLimit	INT16	rw	Y	–32767… <UpperLimit> (0x5261)	–16384

7.9.3 Ramp

The ramp function limits the slew rate of the input signal.
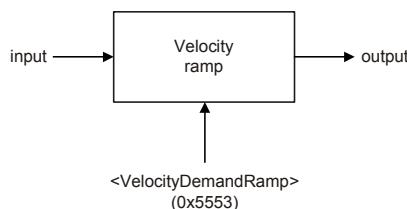
Figure 116: Ramp function

This function limits the output signal's rate of change to <VelocityDemandRamp> (0x5553) per 1/10000 second. The ramp function is disabled (output = input) if this parameter is zero.
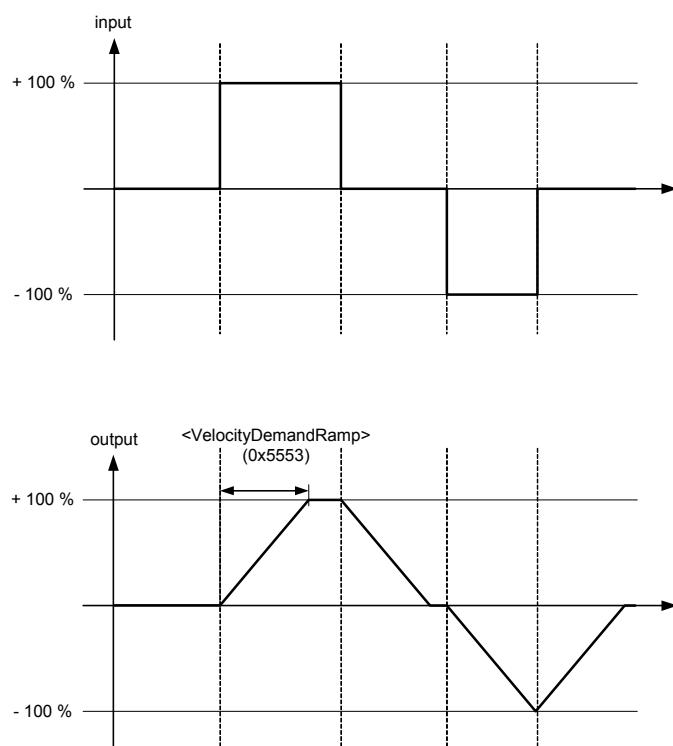
Figure 117: Velocity demand ramp

7.9.3.1 Object 0x5553: Velocity demand ramp

This parameter defines the demand signal's maximum rate of change.

Max rate of change per second = <VelocityDemandRamp> (0x5553) x 10000

ValveVelocityControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5553	1	VelocityDemandRamp	INT32	rw	Y	INT32	0

7.10 Axis velocity controller
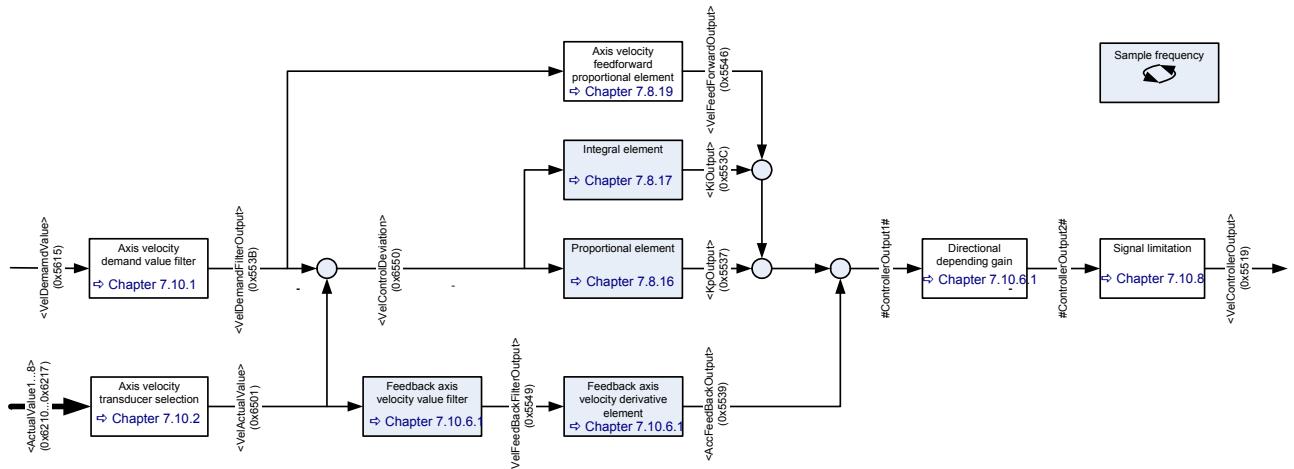
Figure 118: Axis velocity controller

7.10.1 Axis velocity demand value filter

The velocity demand value is filtered with a first order lag element (PT1).
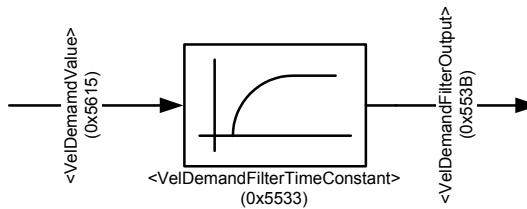
Figure 119: Axis velocity demand value filter

7.10.1.1 Object 0x5533: Demand value filter time constant

This parameter sets the time constant in seconds of the first order lag element (PT1).

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5533	1	VelDemandFilterTimeConstant	FLOAT32	rw	Y	0.0…+inf	0.0

7.10.1.2 Object 0x553B: Demand value filter output

This parameter holds the output of the demand filter element.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x553B	1	VelDemandFilterOutput	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> (0x5545) will select the used mode.
The parameter <VelActualValueReference> (0x5542) will select the interface in case the encoder supplies the velocity signal. Otherwise the <PosActualValue1> (0x6601) needs to be filtered and derived to get the actual velocity. In both modes, the velocity is multiplied with <VelActualGain> (0x5544) to get the <VelActualValue> (0x6501).
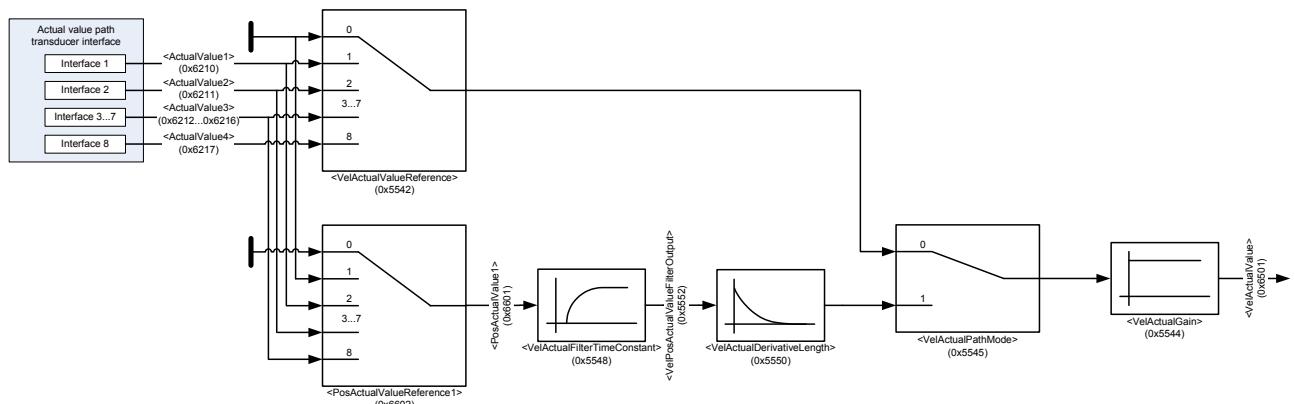
Figure 120: Axis velocity actual value

⇨ Chapter "7.8.14.1 Object 0x6602: Actual value reference", page 199
⇨ Chapter "7.8.1 Object 0x6601: Actual value", page 196
⇨ Chapter "7.10.3.3 Object 0x6550: Control deviation", page 212

7.10.2.1 Object 0x5542: Actual value reference

This parameter selects the transducer with the actual velocity input.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5542	1	VelActualValueReference	UINT8	rw	Y	0…8	1

7.10.2.2 Object 0x5548: Actual value filter time constant

This parameter sets the time constant in seconds of the first order lag element (PT1).

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5548	1	VelActualFilterTimeConstant	FLOAT32	rw	Y	0…+inf	0.0

7.10.2.3 Object 0x5552: Actual value filter output

This parameter holds the output of the axis velocity actual filter.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5552	1	VelPosActualValueFilterOutput	FLOAT32	ro	-	FLOAT32	None

7.10.2.4 Object 0x5550: 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> (0x5550) will give a smoother average signal but also a delay to the actual signal.
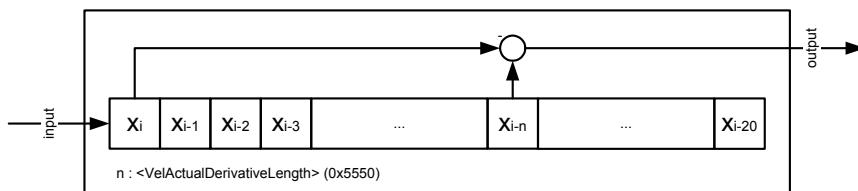
Figure 121: Actual derivative length

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5550	1	VelActualDerivativeLength	UINT8	rw	Y	1…20	1

7.10.2.5 Object 0x5545: Actual value path mode

This switch selects the path of the controller input for the actual velocity.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5545	1	VelActualPathMode	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 86: Possible values of parameter <VelActualPathMode> (0x5545)

7.10.2.6 Object 0x5544: Actual value proportional gain

The actual velocity or the derived actual position is multiplied with this gain to get the <VelActualValue> (0x6501).

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5544	1	VelActualGain	FLOAT32	rw	Y	FLOAT32	0.0

7.10.2.7 Object 0x6501: Axis velocity actual value

The velocity controller needs the actual velocity <VelActualValue> (0x6501) 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> (0x6601). The parameter <VelActualPathMode> (0x5545) will switch between the two variants.

⇨ Chapter "7.10.2 Axis velocity actual value", page 210

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6501	0	VelActualValue	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.
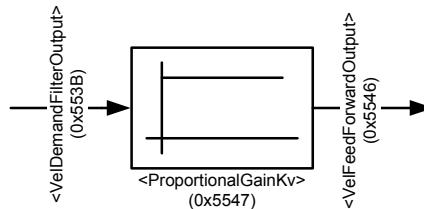
Figure 122: Axis velocity feed forward proportional element

7.10.3.1 Object 0x5547: Velocity feed forward proportional gain

This parameter contains the velocity feed forward gain.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5547	1	ProportionalGainKv	FLOAT32	rw	Y	FLOAT32	0.0

7.10.3.2 Object 0x5546: Velocity feed forward output

This parameter holds the output of the axis velocity feed forward proportional element.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5546	1	VelFeedForwardOutput	FLOAT32	ro	-	FLOAT32	None

7.10.3.3 Object 0x6550: Control deviation

This parameter holds the deviation between the axis velocity demand filter value and the axis velocity actual value.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6550	0	VelDemandFilterOutput	INT32	ro	-	INT32	None

7.10.4 Proportional element

The proportional part output is generated by the proportional gain parameter <ProportionalGainKp> (0x5514). The input signal is the axis position control deviation.
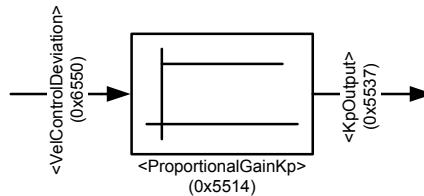
Figure 123: Proportional element

7.10.4.1 Object 0x5514: Proportional gain

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5514	1	ProportionalGainKp	FLOAT32	rw	Y	FLOAT32	0.0

7.10.4.2 Object 0x5537: Kp output

This parameter holds the output of the proportional element of the axis velocity controller.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5537	1	KpOutput	FLOAT32	ro	-	FLOAT32	None

7.10.5 Integral element

The integral element is composed of an input depending gain, the integrator and a limiter.
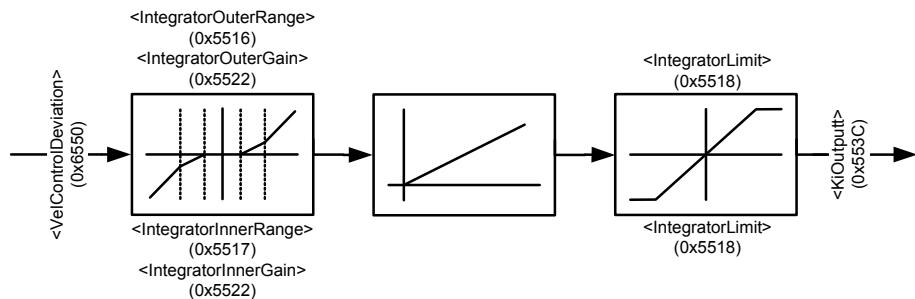
Figure 124: Integral element

7.10.5.1 Object 0x5522: Integral gain

The gain of the integral part depends on the value of the absolute input signal |Input|.

Zero range: |Input| less than <IntegralInnerRange> (0x5517). Output = Zero.
Inner range: |Input| between <IntegralInnerRange> (0x5517) and <IntegralOuterRange> (0x5516).

Output = <IntegralInnerGain> (0x5522) x Input

Outer range: |Input| greater than <IntegralOuterRange> (0x5516).

Output = <IntegralOuterGain> (0x5522) x Input

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5522	1	IntegralInnerGain	FLOAT32	rw	Y	FLOAT32	0.0
0x5522	2	IntegralOuterGain	FLOAT32	rw	Y	FLOAT32	0.0

7.10.5.2 Object 0x5517: Integral inner range

Border of the absolute input signal to switch between gain = 0.0 and gain = <IntegralInnerGain> (0x5522).

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5517	1	IntegralInnerRange	FLOAT32	rw	Y	0.0…2147483647	0.0

7.10.5.3 Object 0x5516: Integral outer range

Border of the absolute input signal to switch between gain = <IntegralInnerGain> (0x5522) and gain = <IntegralOuterGain> (0x5522).

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5516	1	IntegralOuterRange	FLOAT32	rw	Y	0.0…2147483647	0.0

7.10.5.4 Object 0x5518: Integral limit

Limit of the output signal of the integral element.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5518	1	IntegralLimit	FLOAT32	rw	Y	0…+inf	16384000000

7.10.5.5 Object 0x553C: Ki output

This parameter holds the output of the integral element of the axis velocity controller.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x553C	1	KiOutput	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 0x5548: Actual value filter time constant", page 210
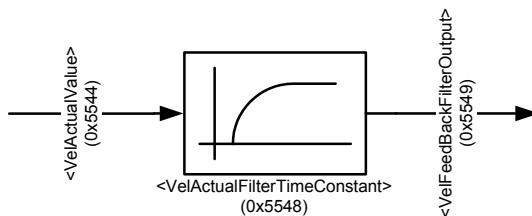
Figure 125: Feedback axis velocity value filter

7.10.6.2 Object 0x5549: Velocity feedback filter output

This is the filtered output of the axis velocity actual value.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5549	1	VelFeedBackFilterOutput	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 0x5550: Actual derivative length", page 211
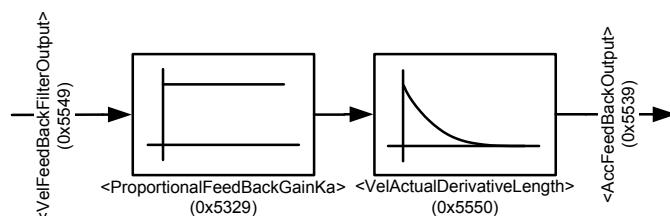
Figure 126: Feedback axis velocity derivative element

7.10.6.4 Object 0x5329: Feedback axis velocity derivative gain

Gain for the acceleration feedback.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5329	1	ProportionalFeedBackGainKa	FLOAT32	rw	Y	FLOAT32	0.0

7.10.6.5 Object 0x5539: Acceleration feedback output

Output of the acceleration feedback to the controller.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5539	1	AccFeedBackOutput	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.
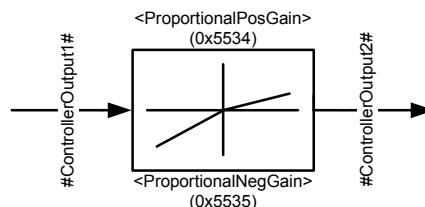
Figure 127: Directional depending gain

7.10.7.1 Object 0x5534: Directional depending positive proportional gain

This gain is used while the input signal is positive.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5534	1	ProportionalPosGain	FLOAT32	rw	Y	FLOAT32	1.0

7.10.7.2 Object 0x5535: Directional depending negative proportional gain

This gain is used while the input signal is positive.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5535	1	ProportionalNegGain	FLOAT32	rw	Y	FLOAT32	1.0

7.10.8 Signal limitation

The output of the controller is limited to ±16384 (±100 %).
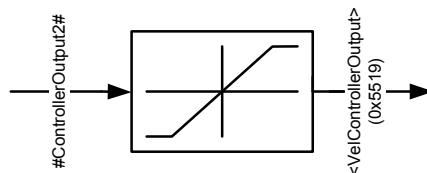
Figure 128: Signal limitation

7.10.8.1 Object 0x5519: Controller output

This parameter holds the controller output of the axis velocity controller.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5519	1	VelControllerOutput	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> (0x5540).
⇨ Chapter "7.1.13.1 Object 0x3030: Basic sample frequency", page 136

7.10.9.1 Object 0x5540: Sample frequency divider

This parameter holds the sample frequency divider.

DriveVelocityControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5540	1	VelSampleFrequencyDivider	UINT8	rw	Y	1…255	2

7.11 Flow setpoint conditioning / demand value generator

The flow controller does not have a demand value generator.
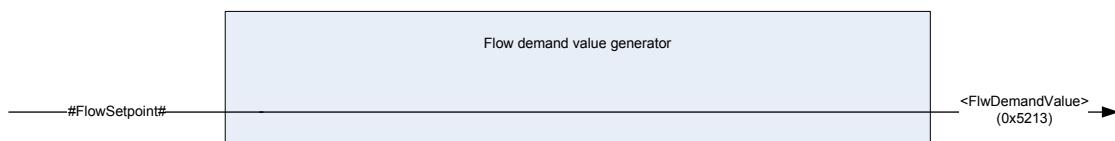
Figure 129: Flow setpoint conditioning / demand value generator

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.
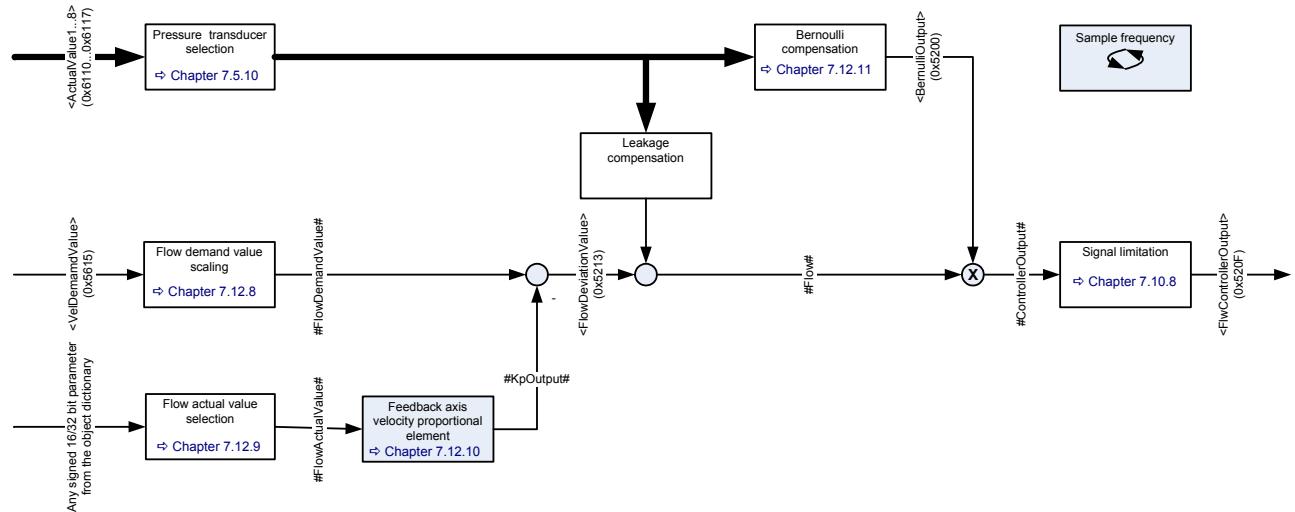
Figure 130: Flow control

⇨ Chapter "7.10.2 Axis velocity actual value", page 210

7.12.1 Object 0x5213: Deviation value

This parameter holds the deviation between the flow demand filter value and the flow feedback value.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5213	0	FlwDeviationValue	INT16	ro	-	INT16	None

7.12.2 Object 0x5200: Bernoulli output

This parameter holds the Bernoulli output of the flow position controller.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5200	0	BernoulliOutput	INT16	ro	-	INT16	None

7.12.3 Object 0x520F: Controller output

This parameter holds the controller output of the flow position controller.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x520F	0	FlwControllerOutput	INT16	ro	-	INT16	None

7.12.4 Sample frequency

The sample frequency is equal to the <BasicSampleFrequency> (0x3030).
⇨ Chapter "7.1.13.1 Object 0x3030: Basic sample frequency", page 136

7.12.5 Flow control mode

The <FlwControlMode> (0x5205) 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> (0x5205)
P-A	P-B	x	x	0
P-A	A-T	x		1
B-t	P-B		x	Not implemented
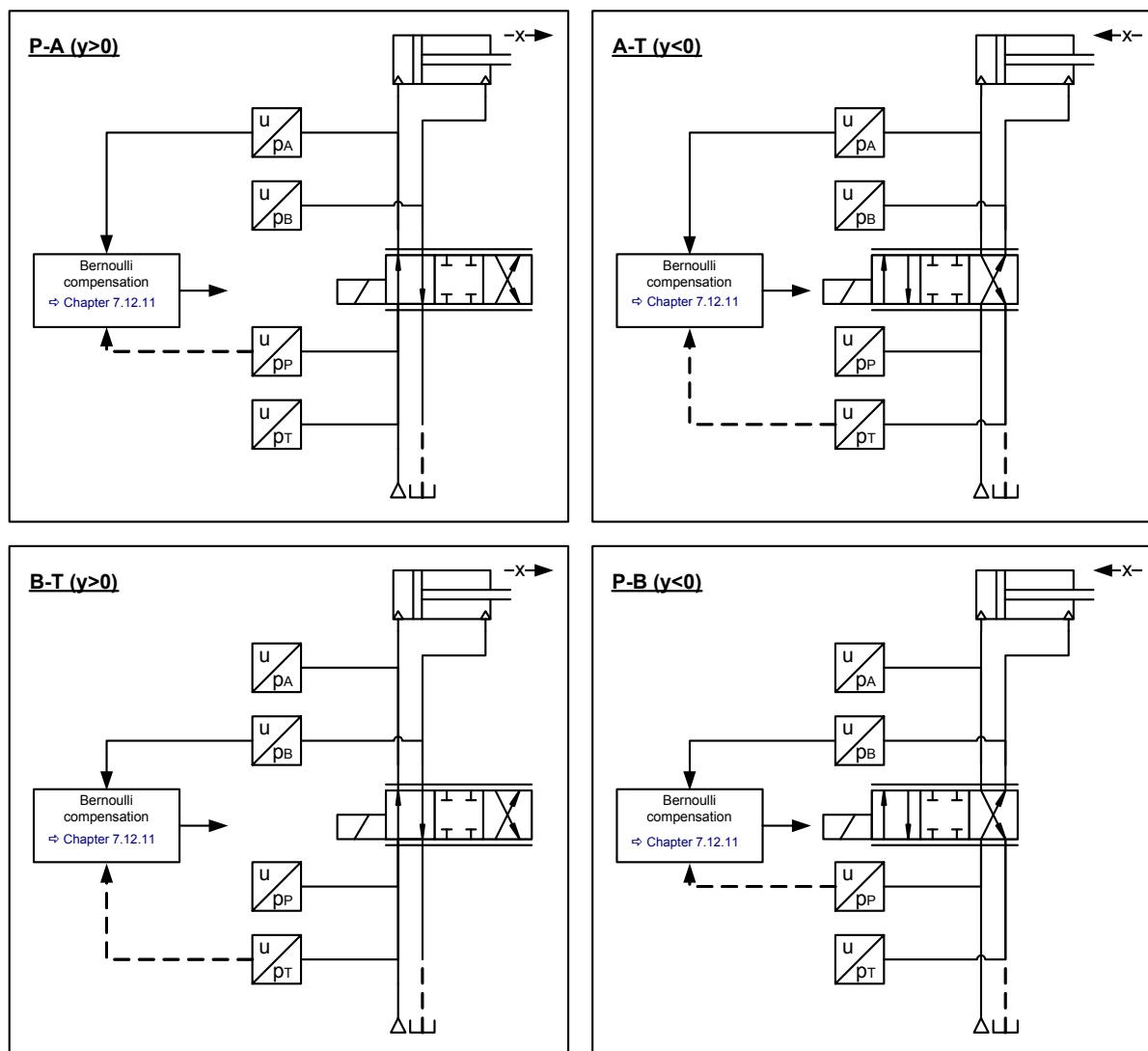
Figure 131: Flow control mode

7.12.5.1 Object 0x5205: Flow control mode

The flow control mode selects the pressure sensor configuration.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5205	0	FlwControlMode	UINT8	rw	Y	0…1	0

Value description

<FlwControlMode>	Description
0	P-A, P-B edge
1	P-A, A-T edge

Table 87: Possible values of parameter <FlwControlMode> (0x5205)

7.12.6 Flow transducer selection

The pressure actual values are read and scaled via the transducer interfaces. The parameter <ActiveTransducerInterfaceAreaA> (0x230D[N]) 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> (0x230F[N]), <ActiveTransducerInterfaceSystem> (0x2330) and <ActiveTransducerInterfaceTank> (0x5217). All actual pressure values are filtered with a low pass filter.

⇨ Chapter "7.5.10.1 Object 0x230D[N]: Active transducer interface area A", page 173
⇨ Chapter "7.5.10.2 Object 0x230F[N]: Active transducer interface area B", page 173

⇨ Chapter "7.5.11.1 Object 0x23F2: Actual pressure filter cutoff frequency", page 174
⇨ Chapter "7.5.11.2 Object 0x23F3: Actual pressure filter order", page 174
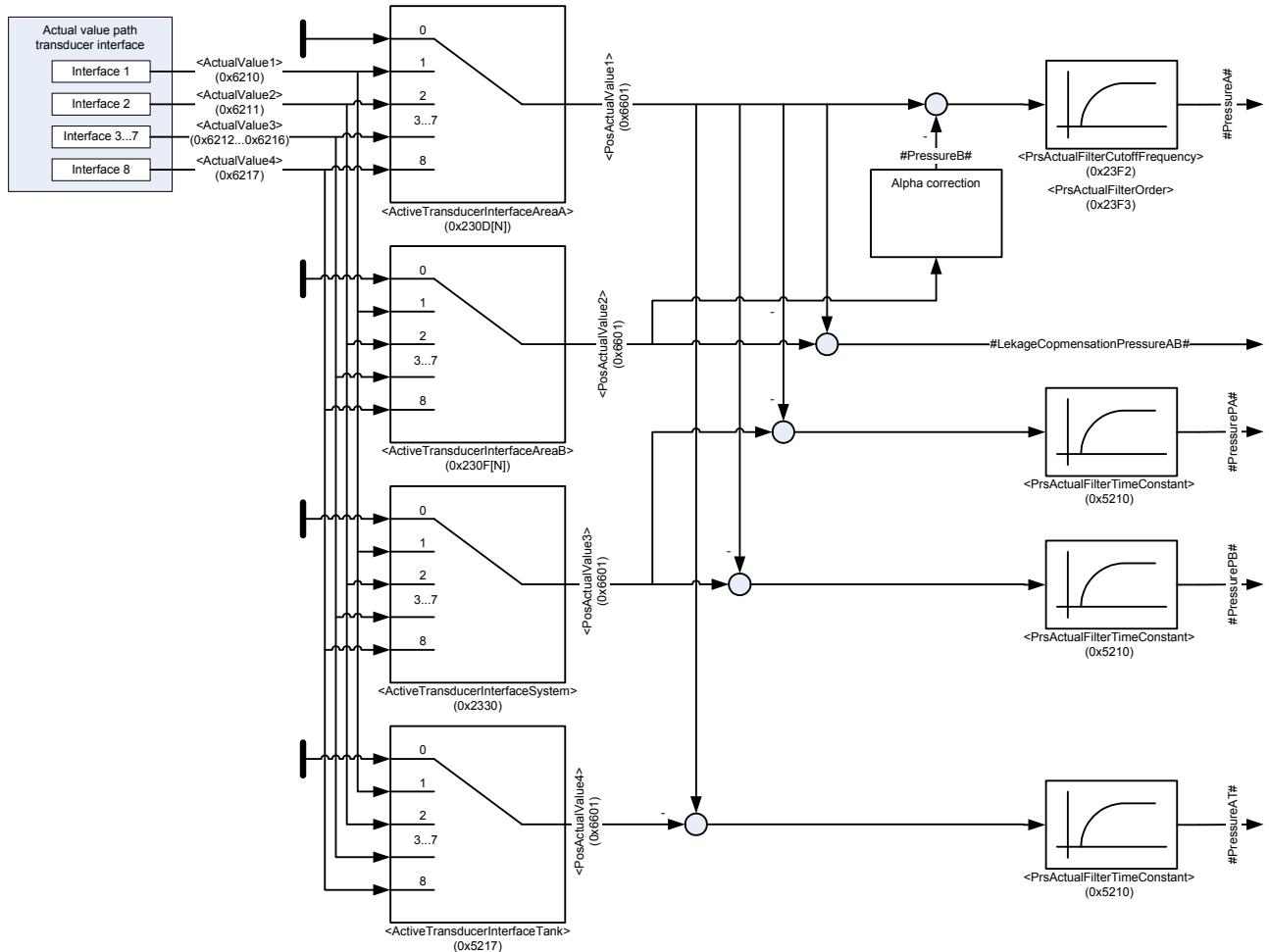
Figure 132: Flow transducer selection

7.12.6.1 Object 0x2330: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x2330	0	ActiveTransducerInterfaceSystem	INT8	rw	Y	0…8	0

7.12.6.2 Object 0x5217: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5217	0	ActiveTransducerInterfaceTank	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.

$$\frac{\left(\frac{<CylinderPistonDiameter>}{(0x585F)}\right)^2 - \left(\frac{<CylinderRodDiameterB>}{(0x585E)}\right)^2}{\left(\frac{<CylinderPistonDiameter>}{(0x585F)}\right)^2 - \left(\frac{<CylinderRodDiameterA>}{(0x585D)}\right)^2}$$

<FlwAlpha>
(0x5219)

<PosActualValue2>
(0x6601)

×
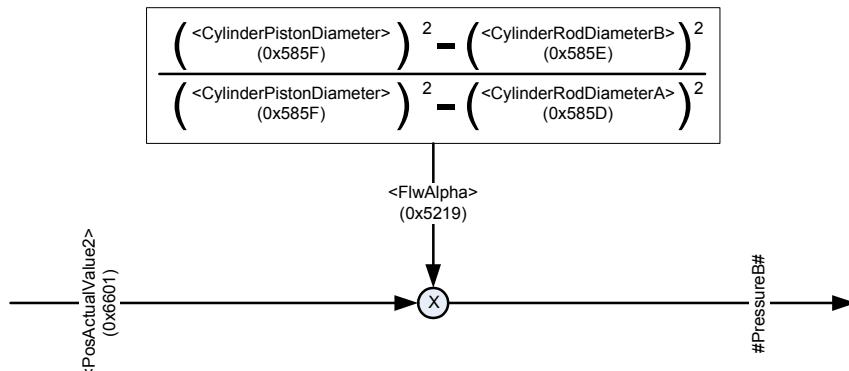
#PressureB#

Figure 133: Alpha correction

7.12.7.1 Object 0x5219: Flow alpha

This parameter is the multiplier for the pressure in B-side to get the alpha corrected actual pressure.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5219	0	FlwAlpha	FLOAT32	ro	-	FLOAT32	None

7.12.7.2 Object 0x5210: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5210	0	PrsActualFilterTimeConstant	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.
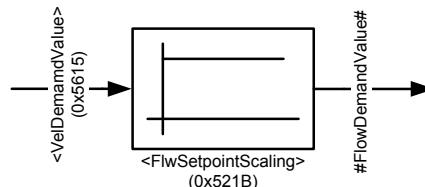
Figure 134: Flow demand value scaling

7.12.8.1 Object 0x521B: Demand value scaling

This parameter sets the demand value scaling.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x521B	0	FlwSetpointScaling	FLOAT32	rw	Y	−1.0…1.0	1.0

7.12.9 Flow actual value selection
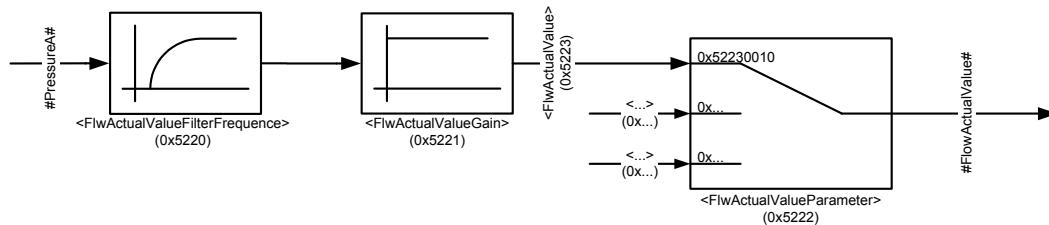
Figure 135: Flow actual value selection

7.12.9.1 Object 0x5223: Actual value

This parameter sets the demand value scaling.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5223	0	FlwActualValue	INT16	ro	-	INT16	None

7.12.9.2 Object 0x5220: Actual value high pass filter frequency

This parameter holds the cut off frequency in Hertz of the high pass filter.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5220	0	FlwActualValueFilterFrequence	FLOAT32	rw	Y	FLOAT32	20.0

7.12.9.3 Object 0x5221: Actual value proportional gain

This parameter holds the proportional gain for the flow actual value.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5221	0	FlwActualValueGain	FLOAT32	rw	Y	FLOAT32	1.0

7.12.9.4 Object 0x5222: Actual value parameter

By default this parameter is pointing to <FlwActualValue> (0x5223) as feedback input for the axis velocity controller. But also each other signal can be mapped to the feedback input.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5222	0	FlwActualValueParameter	UINT32	rw	Y	UINT32	0x52230010

Value description

<FlwActualValueParameter>			
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-Index	Parameter bit length:0x10
Default	0x52	0x23	0x01	0x10

Table 88: Possible values of parameter <FlwActualValueParameter> (0x5222)

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> (0x5223), 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.
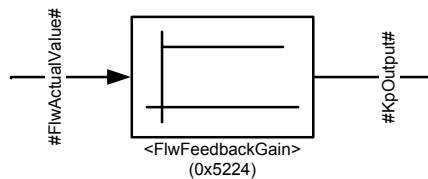
Figure 136: Feedback axis velocity proportional element

7.12.10.1 Object 0x5224: Feedback axis velocity proportional gain

This parameter holds the proportional gain for the flow controller.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5224	0	FlwFeedbackGain	FLOAT32	rw	Y	FLOAT32	0.0

7.12.11 Bernoulli compensation

The Bernoulli compensator compensates the influence of the load on the hydraulic cylinder.
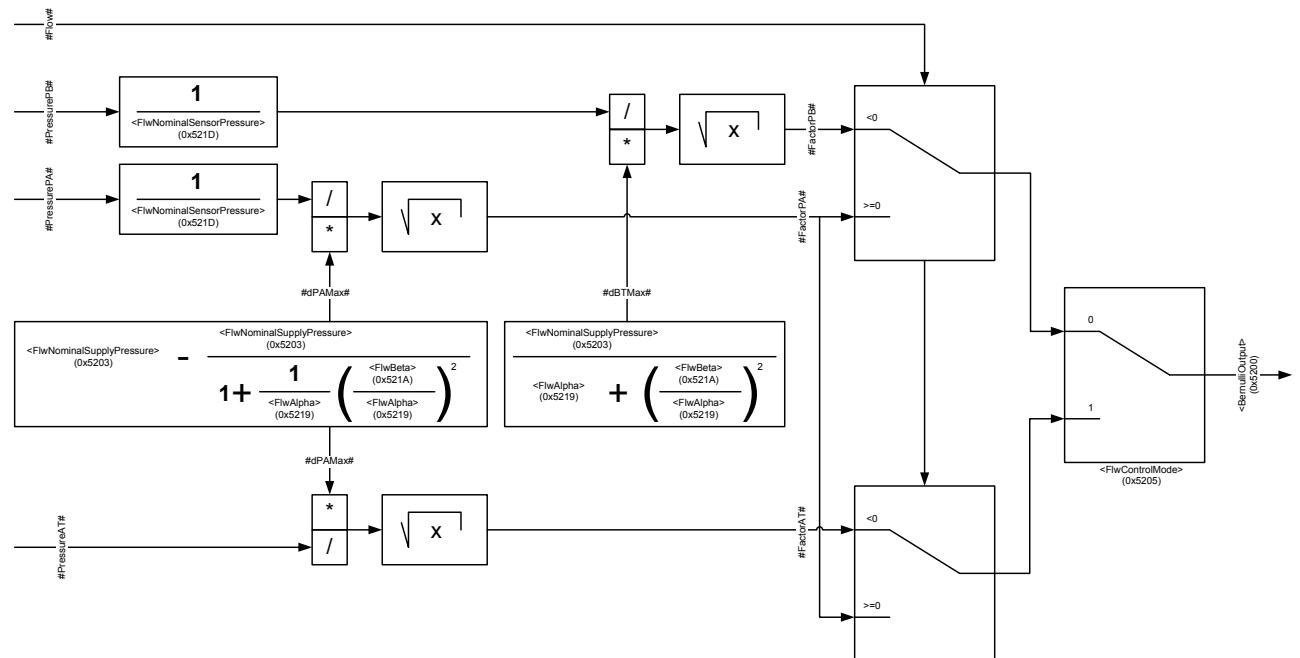
Figure 137: Bernoulli compensator

7.12.11.1 Object 0x5215: Nominal flow A side

Maximal flow on port A of the servo valve.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5215	0	FlwNominalFlowASide	FLOAT32	rw	Y	0.01…+inf	0.0

7.12.11.2 Object 0x5216: Nominal flow B side

Maximal flow on port B of the servo valve.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5216	0	FlwNominalFlowBSide	FLOAT32	rw	Y	0.01…+inf	0.0

7.12.11.3 Object 0x521A: Flow beta

This is a calculated servo valve depending factor. It equals the <FlwNominalFlowBSide> (0x5216) divided by <FlwNominalFlowASide> (0x5215) (at the same system pressure).

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x521A	0	FlwBeta	FLOAT32	ro	-	FLOAT32	0.0

7.12.11.4 Object 0x5203: Nominal supply pressure

The nominal supply pressure of the servo valve is necessary to normalize the command value.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5203	0	FlwNominalSupplyPressure	INT16	rw	Y	1…32767	400

7.12.11.5 Object 0x521D: 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							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x521D	0	FlwNominalSensorPressure	INT16	ro	-	INT16	None

7.12.11.6 Object 0x521F: Maximal flow A side

For the flow control the maximal flow on A-side depending on the nominal pressure on A-side is necessary.
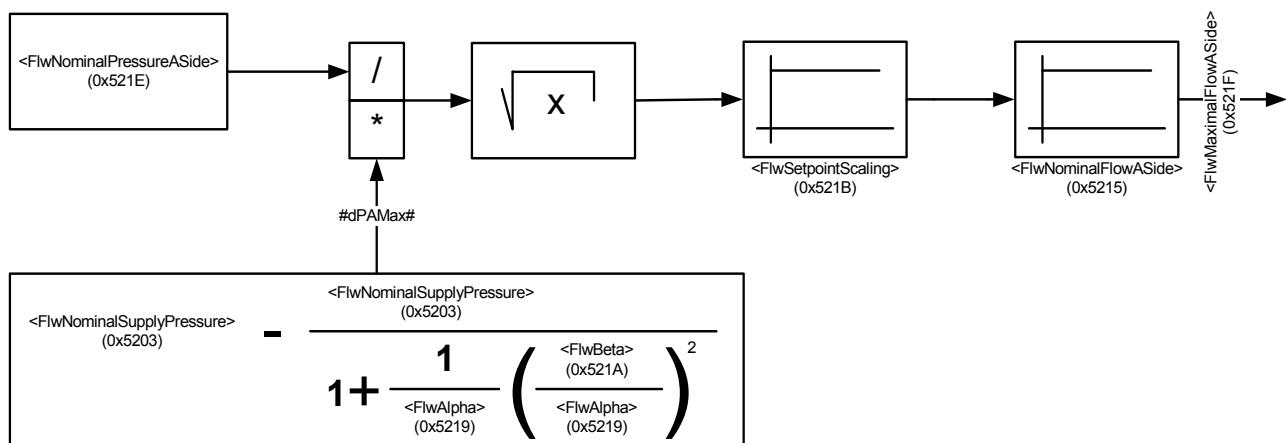
Figure 138: Maximal flow A side

This parameter contains the result of the calculation of the maximal flow.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x521F	0	FlwMaximalFlowASide	FLOAT32	ro	-	FLOAT32	None

7.12.11.7 Object 0x521E: Nominal pressure A side

For the calculation of <FlwMaximalFlowASide> (0x521F), the nominal pressure is necessary (the pressure used to measure the nominal flow <FlwNominalFlowASide> (0x5215)).

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x521E	0	FlwNominalPressureASide	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).
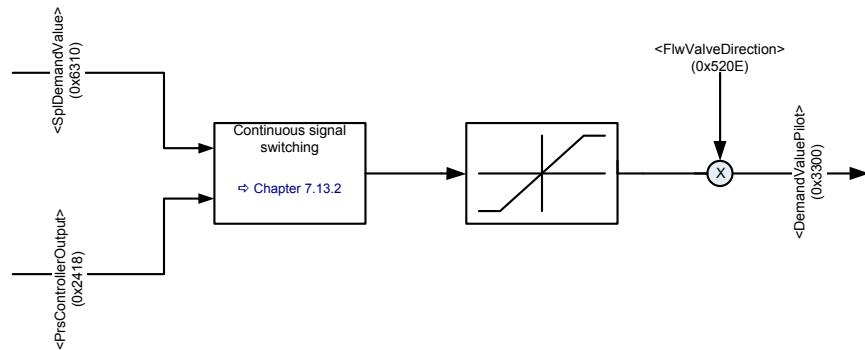
Figure 139: Flow / pressure switchover

7.13.1 Object 0x520E: Flow valve direction

Signal to negate the output to the valve. Thus, the direction of the axis movement can be inverted.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x520E	0	FlwValveDirection	INT8	rw	Y	−1…1	1

7.13.2 Continuous signal switching

The switch-over criteria between the control modes are the PressureDemandValue (<PrsDemandValue> (0x6390)), the PressureActualValue (<PrsActualValue> (0x6381)), the output of the PressureControllerOutput (<ControllerOutput> (0x2418)) and the output of the FlowControllerOutput (<ControllerOutput> (0x520F)). 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.

((PressureDemandValue > 0.0) and (PressureControllerOutput > FlowControllerOutput))
or
((PressureDemandValue <= 0.0) and (PressureControllerOutput < FlowControllerOutput))
or
Change sign of FlowControllerOutput

((PressureDemandValue) > 0.0) and (PressureActualValue > PressureDemandValue))
or
((PressureDemandValue < 0.0) and (PressureActualValue < PressureDemandValue))
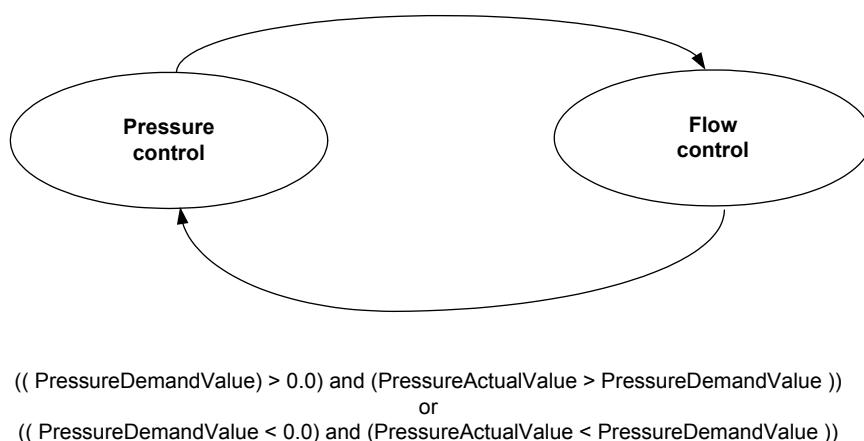
Figure 140: Continuous signal switching

7.13.2.1 Object 0x5204: 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> (0x2418)) to FlowControllerOutput (<ControllerOutput> (0x520F)) and back.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5204	1	SwitchingTimeConstantFlow2Pressure	FLOAT32	rw	Y	0.0…+inf	30.0
0x5204	2	SwitchingTimeConstantPressure2Flow	FLOAT32	rw	Y	0.0…+inf	3.0

7.14 Axis status

7.14.1 Object 0x561D: Axis status word

Drive							
Index	Sub-index	Parameter name	Data type	Access	Per- sis- tence	Value range	Default
0x561D	0	AxisStatusWord	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> (0x6610) – <ActualValue1> (0x6601)|) is less than <PositionWindow1> (0x570C) and the velocity |<ActualValue> (0x6501)| is less than <VelocityLimitWindow> (0x570D).
Bit 2 0x0004	Velocity window reached	Active while the velocity difference (|<ActualValue> (0x6501) – <VelocityLimit> (0x570E)|) is less than <VelocityLimitWindow> (0x570D).
Bit 3 0x0008	Velocity stopped	Active while the actual velocity (|<ActualValue> (0x6501) | is less than <VelocityLimitWindow> (0x570D).
Bit 4 0x0010	Force reached	Active while the pressure difference (|<prsActualvalue> (0x6381) – <PrsDemandValue> (0x6390)|) is less than <PressureWindow> (0x5710).
Bit 5 0x0020	Position window 2 reached	Active while the position difference (|<DemandValue> (0x6610) – <ActualValue1> (0x6601) |) is less than <PositionWindow2> (0x570C).
Bit 6 0x0040	Maximum force reached	Active while the actual pressure <prsActualvalue> (0x6381) is greater than the <MaximumPressureWindow> (0x5711).
Bit 7 0x0080	Axis limit touched	Not implemented.
Bit 8 0x0100	Install or reference mode active	Active while the <DeviceMode> (0x6042) = 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 89: Possible values of parameter <AxisStatusWord> (0x561D)

7.14.2 Object 0x561C: Axis status and device status word

This 32 bit status word is a combination of the 16 bit <AxisStatusWord> (0x561D) and the 16 bit <StatusWord> (0x6041).

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 52

Drive							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x561C	0	AxisStatusAndDeviceStatusWord	UINT32	ro	-	UINT32	None

Value description

<AxisStatusAndDeviceStatusWord>		
Bit	Word	Description
0…15	<AxisStatusWord> (0x561D)	Axis status word.
16…31	<StatusWord> (0x6041)	Device status word.

Table 90: Possible values of parameter <AxisStatusAndDeviceStatusWord> (0x561C)

7.14.3 Object 0x570C: Axis position deviation window

The position deviation window is used to set bit 1 and bit 5 in the <AxisStatusWord> (0x561D).

Bit 1 is active while the absolute position difference (|<DemandValue> (0x6610) – <ActualValue1> (0x6601)|) is less than <PositionWindow1> (0x570C) and the velocity |<ActualValue> (0x6501)| is less than <VelocityLimitWindow> (0x570D).

Bit 5 is active while the absolute position difference (|<DemandValue> (0x6610) – <ActualValue1> (0x6601)|) is less than <PositionWindow2> (0x570C).

Drive							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x570C	1	PositionWindow1	INT32	rw	Y	0…2147483647	0
0x570C	2	PositionWindow2	INT32	rw	Y	0…2147483647	0

7.14.4 Object 0x570D: Axis velocity actual value window

The velocity actual value window is used to set bit 1 and bit 3 in the <AxisStatusWord> (0x561D).

Bit 1 is active while the absolute position difference (|<DemandValue> (0x6610) – <ActualValue1> (0x6601)|) is less than <PositionWindow1> (0x570C) and the velocity |<ActualValue> (0x6501)| is less than <VelocityLimitWindow> (0x570D).

Bit 3 is active while the absolute velocity |<ActualValue> (0x6501)| is less than the <VelocityLimitWindow> (0x570D).

Drive							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x570D	0	VelocityLimitWindow	INT32	rw	Y	0…2147483647	0

7.14.5 Object 0x570E: Axis velocity deviation window

The velocity deviation window is used to set bit 2 in the <AxisStatusWord> (0x561D).

Bit 2 is active while the velocity difference (|<ActualValue> (0x6501) – <VelocityLimit> (0x570E|) is less than the <VelocityLimitWindow> (0x570D).

Drive							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x570E	0	VelocityLimit	INT32	rw	Y	INT32	0

7.14.6 Object 0x5710: Pressure deviation window

The pressure deviation window is used to set bit 4 in the <AxisStatusWord> (0x561D).

Bit 4 is active while the absolute pressure difference (|<prsActualvalue> (0x6381) – <PrsDemandValue> (0x6390)|) is less than the <PressureWindow> (0x5710).

Drive							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5710	0	PressureWindow	INT16	rw	Y	0…32767	0

7.14.7 Object 0x5711: Pressure actual value limit

The Pressure limit window used to set bit 6 in the <AxisStatusWord> (0x561D).

Bit 6 is active while the actual pressure <prsActualvalue> (0x6381) is greater than the <MaximumPressureWindow> (0x5711).

Drive							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5711	0	MaximumPressureWindow	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> (0x6043) 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> (0x6041).

7.15.1 Spool position control deviation monitoring
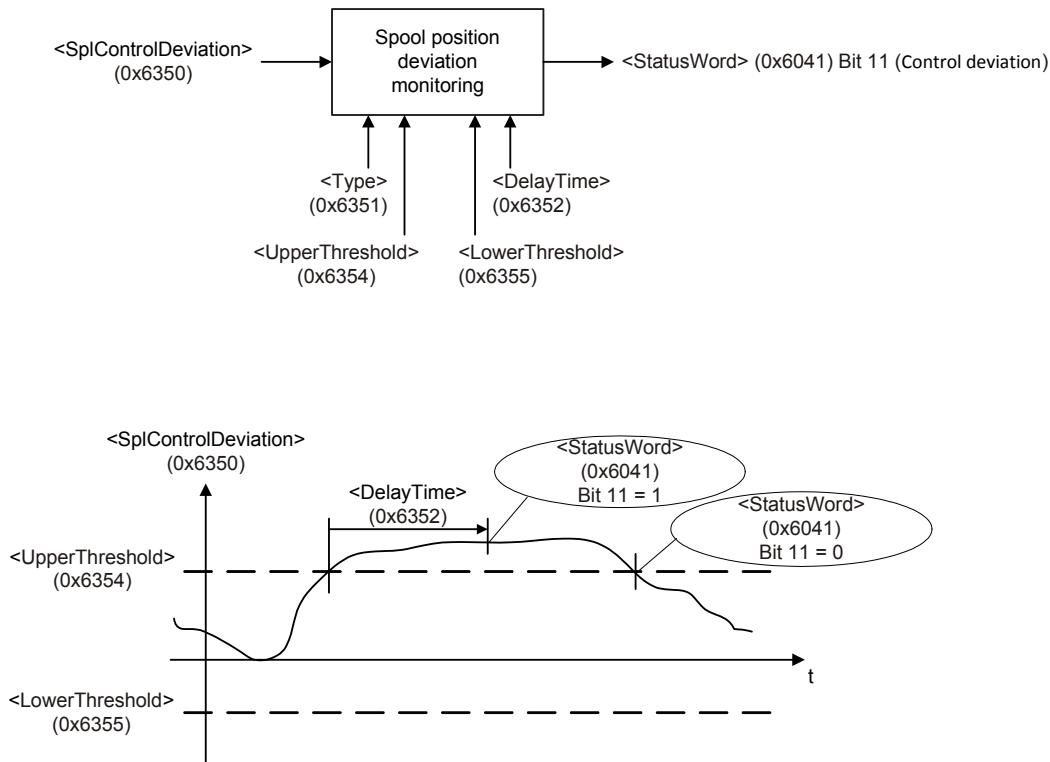
Figure 141: Spool position control deviation monitoring

⇨ Chapter "7.3.5.1 Object 0x6350: Control deviation", page 156

7.15.1.1 Object 0x6351: Type

The parameter <Type> (0x6351) is used to activate or deactivate the standard spool position control deviation monitoring function.

ValvePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6351	0	Type	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 91: Possible values of parameter <Type> (0x6351)

7.15.1.2 Object 0x6352: 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> (0x6351), <UpperThreshold> (0x6354), <LowerThreshold> (0x6355) or <DelayTime> (0x6352).

ValvePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6352	1	DelayTime	UINT16	rw	Y	UINT16	30
0x6352	2	Unit	UINT8	ro	-	UINT8	3
0x6352	3	Prefix	INT8	ro	-	INT8	–3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.15.1.3 Object 0x6354: Upper threshold

This parameter defines the upper threshold of the control deviation's tolerance band.

ValvePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6354	1	UpperThreshold	INT16	rw	Y	INT16	512
0x6354	2	Unit	UINT8	ro	-	UINT8	0
0x6354	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.15.1.4 Object 0x6355: Lower threshold

This parameter defines the lower threshold of the control deviation's tolerance band.

ValvePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6355	1	LowerThreshold	INT16	rw	Y	INT16	–512
0x6355	2	Unit	UINT8	ro	-	UINT8	0
0x6355	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.15.2 Pressure control deviation monitoring
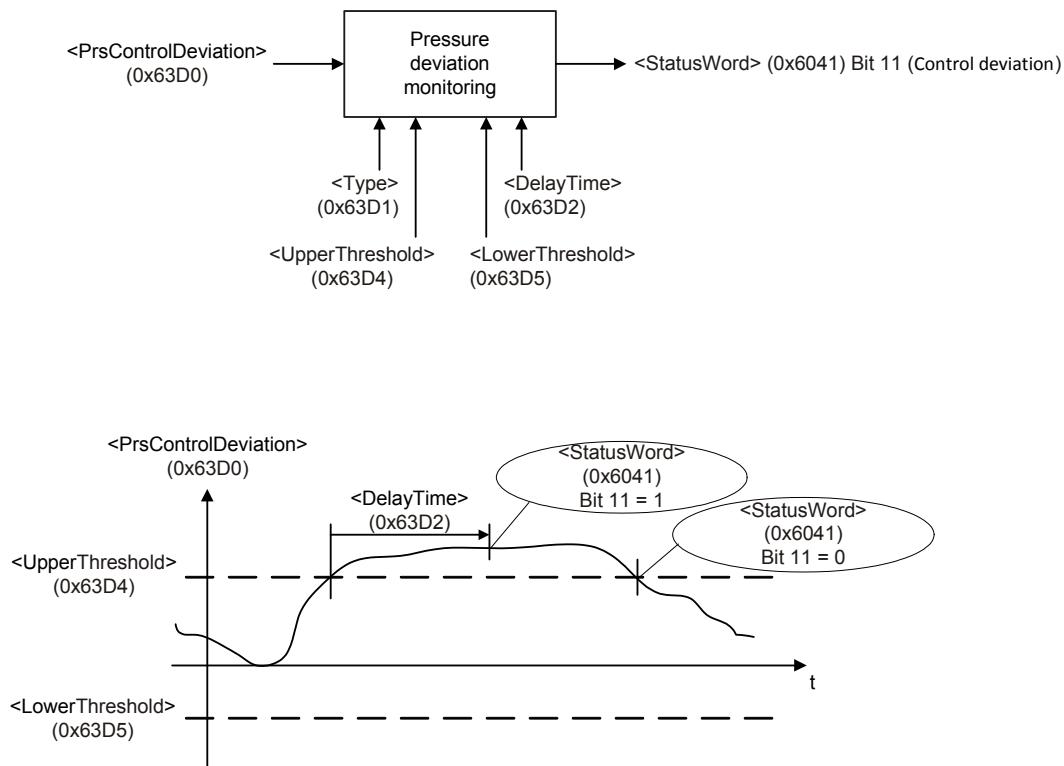
Figure 142: Pressure control deviation monitoring

7.15.2.1 Object 0x63D1: Type

The parameter <Type> (0x63D1) is used to activate or deactivate the standard pressure control deviation monitoring function.

ValvePressureControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x63D1	0	Type	INT8	rw	Y	0…1	0

Value description

<Type>	Description
0	Pressure control deviation monitoring off.
1	Pressure control deviation monitoring on.

Table 92: Possible values of parameter <Type> (0x63D1)

7.15.2.2 Object 0x63D2: 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> (0x63D1), <UpperThreshold> (0x63D4), <LowerThreshold> (0x63D5) or <DelayTime> (0x63D2).

ValvePressureControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x63D2	1	DelayTime	UINT16	rw	Y	UINT16	30
0x63D2	2	Unit	UINT8	ro	-	UINT8	3
0x63D2	3	Prefix	INT8	ro	-	INT8	–3

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.15.2.3 Object 0x63D4: Upper threshold

This parameter defines the upper threshold of the control deviation's tolerance band.

ValvePressureControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x63D4	1	UpperThreshold	INT16	rw	Y	INT16	512
0x63D4	2	Unit	UINT8	ro	-	UINT8	0
0x63D4	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.15.2.4 Object 0x63D5: Lower threshold

This parameter defines the lower threshold of the control deviation's tolerance band.

ValvePressureControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x63D5	1	LowerThreshold	INT16	rw	Y	INT16	–512
0x63D5	2	Unit	UINT8	ro	-	UINT8	0
0x63D5	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.15.3 Axis position control deviation monitoring
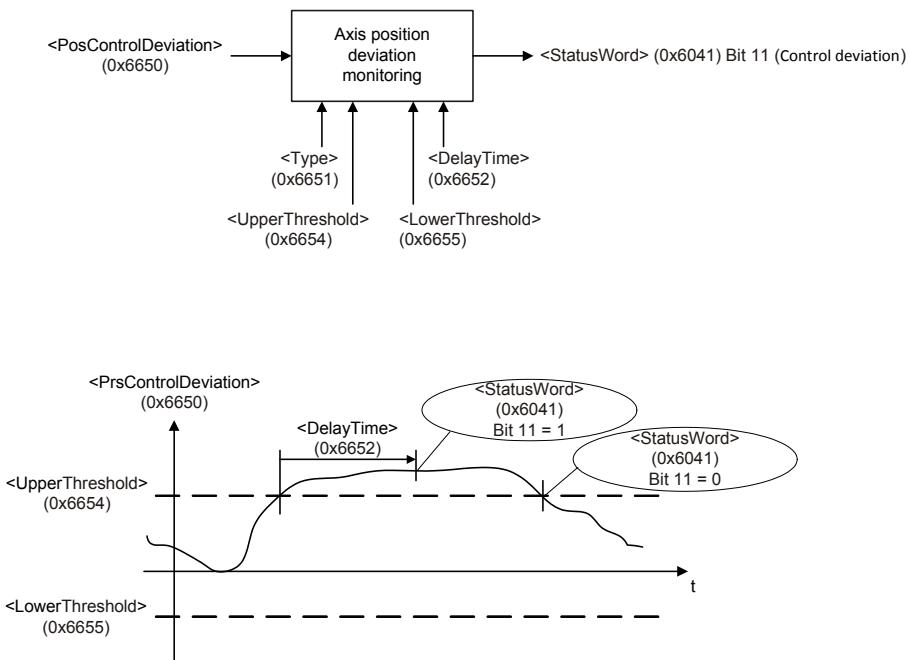
Figure 143: Axis position control deviation monitoring

⇨ Chapter "7.8.3 Object 0x6650: Control deviation", page 196

7.15.3.1 Object 0x6651: Type

This parameter is used to activate or deactivate the standard pressure control deviation monitoring function.

DrivePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6651	0	Type	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 93: Possible values of parameter <Type> (0x6651)

7.15.3.2 Object 0x6652: 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> (0x6651)
- <UpperThreshold> (0x6654)
- <LowerThreshold> (0x6655)
- <DelayTime> (0x6652)

DrivePositionControl_ControlMonitoring								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6652	1	DelayTime	UINT16	rw	Y	UINT16	30

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.15.3.3 Object 0x6654: Upper threshold

This parameter defines the upper threshold of the control deviation's tolerance band.

DrivePositionControl_ControlMonitoring								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6654	1	UpperThreshold	INT16	rw	Y	INT16	512

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.15.3.4 Object 0x6655: Lower threshold

This parameter defines the lower threshold of the control deviation's tolerance band.

DrivePositionControl_ControlMonitoring								
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6655	1	LowerThreshold	INT16	rw	Y	INT16	–512

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.15.4 Axis velocity control deviation monitoring

Figure 144: Axis velocity control deviation monitoring

7.15.4.1 Object 0x6551: Type

This parameter is used to activate or deactivate the standard pressure control deviation monitoring function.

DriveVelocityControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6551	0	Type	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 94: Possible values of parameter <Type> (0x6651)

7.15.4.2 Object 0x6552: 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> (0x6551)
- <UpperThreshold> (0x6554)
- <LowerThreshold> (0x6555)
- <DelayTime> (0x6552)

DriveVelocityControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6552	1	DelayTime	UINT16	rw	Y	UINT16	30

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.15.4.3 Object 0x6554: Upper threshold

This parameter defines the upper threshold of the control deviation's tolerance band.

DriveVelocityControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6554	1	UpperThreshold	INT16	rw	Y	INT16	512

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

7.15.4.4 Object 0x6555: Lower threshold

This parameter defines the lower threshold of the control deviation's tolerance band.

DriveVelocityControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6555	1	LowerThreshold	INT16	rw	Y	INT16	–512

⇨ Chapter "2.4.3 Units and prefix parameter", page 12

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> (0x2421) and a <LowerLimit> (0x2422), which are set by the factory. If the failsafe spool position monitoring function is enabled by setting the parameter <DigitalOutputType1> (0x2420) to 1 (failsafe spool position monitoring on), the servo valve monitors the failsafe position of the spool position <ActualValue> (0x6301) and sets the digital output 1.

⇨ Chapter "6.8.2 Object 0x5E41: Digital output configuration", page 101

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!
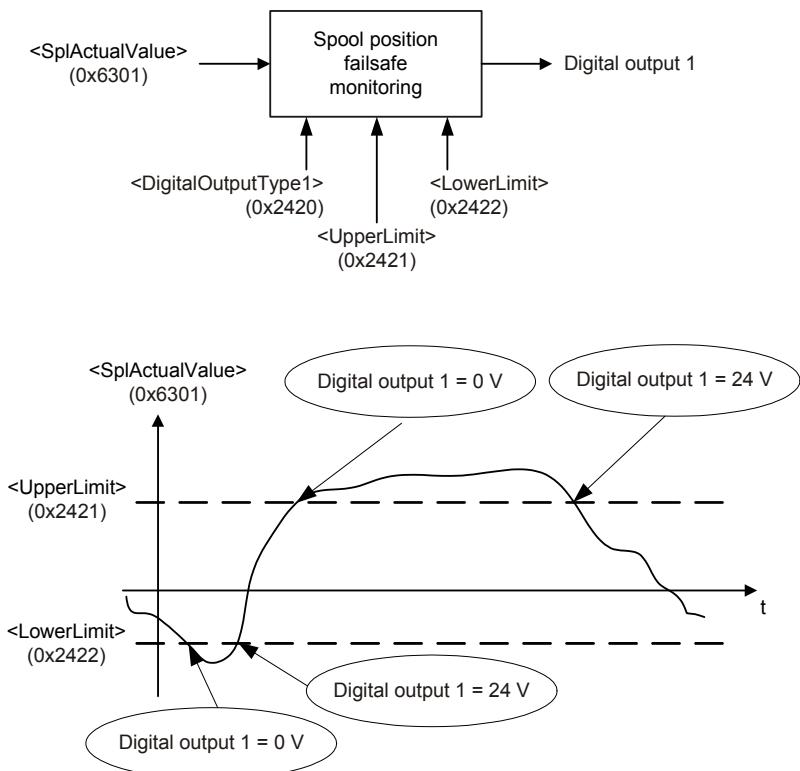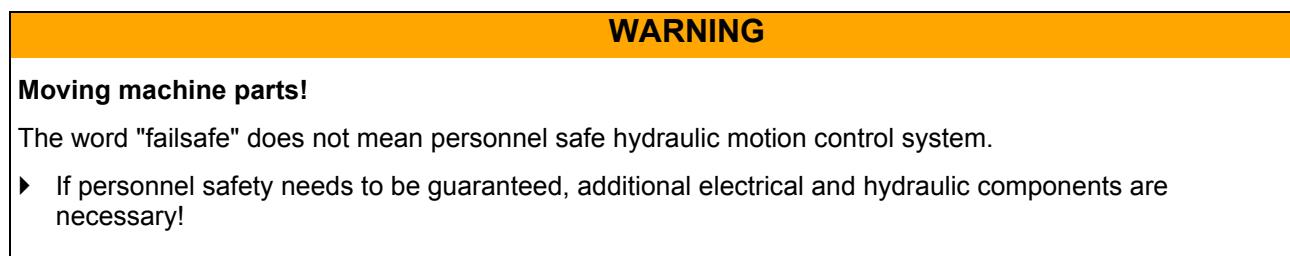
Figure 145: Failsafe monitoring

Electrical signal

Spool position <SplActualValue> (0x6301)	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> (0x6041)

<StatusWord> (0x6041) (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> (0x2420) is set to 1 (failsafe spool position monitoring on).

7.15.5.1 Object 0x2421: Upper limit

The parameter <UpperLimit> (0x2421) shows the upper limit of the spool position failsafe monitoring window.

ValveFailSafeWindowMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2421	0	UpperLimit	INT16	ro	-	<LowerLimit> (0x2422)…32767	16384

7.15.5.2 Object 0x2422: Lower limit

The parameter <LowerLimit> (0x2422) shows the lower limit of the spool position failsafe monitoring window.

ValveFailSafeWindowMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2422	0	LowerLimit	INT16	ro	-	−32768… <UpperLimit> (0x2421)	−16384

7.15.5.3 Object 0x3307: 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> (0x3307) and <SpringPositionMaximum> (0x3308).

SpringPositionMinimum							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3307	0	SpringPositionMinimum	INT16	ro	Y	INT16	0

7.15.5.4 Object 0x3308: 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> (0x3307) and <SpringPositionMaximum> (0x3308).

SpringPositionMaximum							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3308	0	SpringPositionMaximum	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 <AnaInMonitorCurrent2> (0x3217) is set to 0 (fault code 33)	Yes if <AnaInMonitorCurrent2> (0x3217) is set to 1 (fault code 33)
Analog input 3	Yes (fault code 29)	No	Yes if <AnaInMonitorCurrent3> (0x3228) is set to 0 (fault code 34)	Yes if <AnaInMonitorCurrent3> (0x3228) is set to 1 (fault code 34)
Analog input 4	Yes (fault code 30)	No	Yes if <AnaInMonitorCurrent4> (0x3227) is set to 0 (fault code 35)	Yes if <AnaInMonitorCurrent4> (0x3227) is set to 1 (fault code 35)

Table 95: 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.

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 96: Possible fault codes

7.15.8.1 Object 0x3217: 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							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3217	0	AnaInMonitorCurrent2	UINT8	rw	Y	0…1	0

7.15.8.2 Object 0x3228: 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							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3228	0	AnaInMonitorCurrent3	UINT8	rw	Y	0…1	0

7.15.8.3 Object 0x3227: 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							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3227	0	AnaInMonitorCurrent4	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.
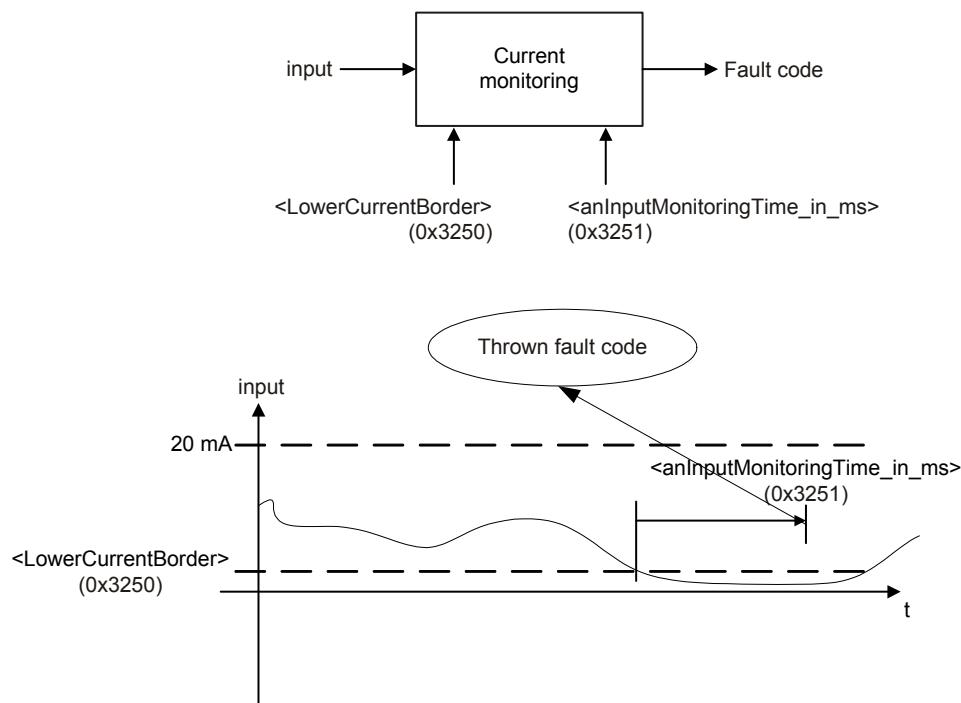
Figure 146: 4…20 mA analog input signal monitoring

If the current is below <LowerCurrentBorder> (0x3250) for <anInputMonitoringTime_in_ms> (0x3251) 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 0x3250: Lower current border

This parameter contains the lower current boarder.

AnalogInput							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3250	0	LowerCurrentBorder	FLOAT32	rw	Y	2.2…20.0	3.0

7.15.8.5 Object 0x3251: Analog input monitoring time

This parameter contains the delay time (in ms) before generating the fault code.

AnalogInput							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3251	0	anInputMonitoringTime_in_ms	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 0x2803: 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: $3.1 \text{ V} \leq \text{CpuSupplyVoltage} \leq 3.5 \text{ V}$

Hardware_DiagnosticData							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2803	0	CpuSupplyVoltage	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 97: Fault codes

The power supply voltage should be in the range of 18…32 V to ensure proper operation.

7.15.10.2 Object 0x2804: 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: $17 \text{ V} \leq \text{PowerSupplyVoltage} \leq 32.5 \text{ V}$

Hardware_DiagnosticData							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2804	0	PowerSupplyVoltage	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 98: Fault codes

7.15.10.3 Object 0x2805: 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							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2805	0	PcbTemperature	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 99: 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 0x2809: 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							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2809	0	PcbMaxTemperatureCustomer	INT16	rw	Y	INT16	0

7.15.10.5 Object 0x280D: Operating time

The parameter <PowerOnTime> (0x280D) contains the power on time (in minutes) since production of the servo valve.

The parameter <OperatingTime> (0x280D) contains the time (in minutes) the servo valve is in the device state machine (DSM) states 'HOLD', 'FAULT HOLD' or 'ACTIVE'.

Hardware_DiagnosticData							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x280D	1	PowerOnTime	UINT32	ro	-	UINT32	0
0x280D	2	OperatingTime	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"
- 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 0x2901: Event expression 1

This parameter contains the expression string of the first event handler.

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2901	0	EventExpression_1	STRING	rw	Y	None	""

7.16.1.2 Object 0x2902: Event expression 2

This parameter contains the expression string of the second event handler.

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2902	0	EventExpression_2	STRING	rw	Y	None	""

7.16.1.3 Object 0x2903: Event expression 3

This parameter contains the expression string of the third event handler.

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2903	0	EventExpression_3	STRING	rw	Y	None	""

7.16.1.4 Object 0x2904: Event expression 4

This parameter contains the expression string of the fourth event handler.

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2904	0	EventExpression_4	STRING	rw	Y	None	""

7.16.1.5 Object 0x2905: Event expression 5

This parameter contains the expression string of the fifth event handler.

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2905	0	EventExpression_5	STRING	rw	Y	None	""

7.16.1.6 Object 0x2906: Event expression 6

This parameter contains the expression string of the sixth event handler.

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2906	0	EventExpression_6	STRING	rw	Y	None	""

7.16.1.7 Object 0x2907: Event expression 7

This parameter contains the expression string of the seventh event handler.

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2907	0	EventExpression_7	STRING	rw	Y	None	""

7.16.1.8 Object 0x2908: Event expression 8

This parameter contains the expression string of the eighth event handler.

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2908	0	EventExpression_8	STRING	rw	Y	None	""

7.16.1.9 Object 0x2909: Event enable

These parameters switch the event handler on or off.

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2909	1	EventEnable_1	UINT8	rw	Y	0…1	0
0x2909	2	EventEnable_2	UINT8	rw	Y	0…1	0
0x2909	3	EventEnable_3	UINT8	rw	Y	0…1	0
0x2909	4	EventEnable_4	UINT8	rw	Y	0…1	0
0x2909	5	EventEnable_5	UINT8	rw	Y	0…1	0
0x2909	6	EventEnable_6	UINT8	rw	Y	0…1	0
0x2909	7	EventEnable_7	UINT8	rw	Y	0…1	0
0x2909	8	EventEnable_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 122

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> (0x2901):
prsval>10000?ctlmod=4;evtena[0]=0;evtena[1]=1

Expression 2 <EventExpression_2> (0x2902):
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.

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.
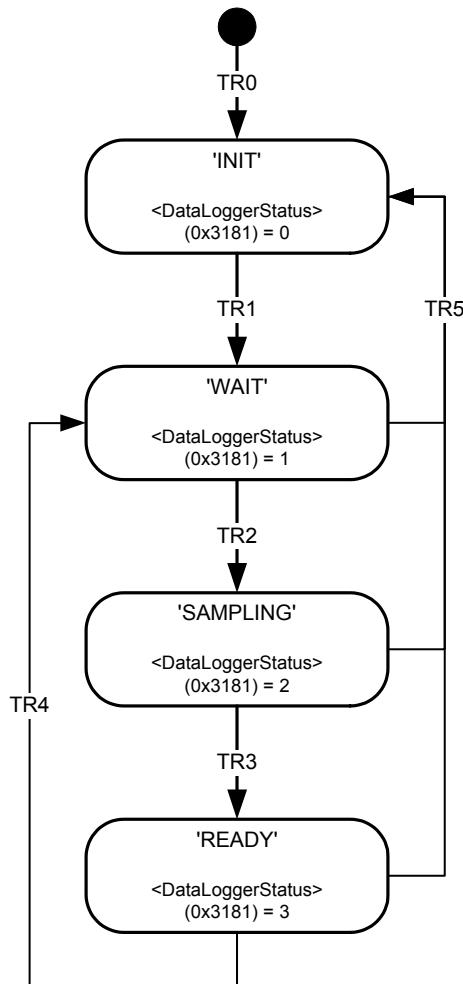
Figure 147: Data logger state machine

	<DataLoggerTriggerType> (0x3188)		
State	0 (Free)	1 (Normal)	2 (Single)
'INIT'	Initialize data logger.		
'WAIT'	The trigger condition is always true. If the <Control> (0x3180) is set to 1 (enable data logger), the state changes to 'SAMPLE'.	If the <Control> (0x3180) 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> (0x3187) in the ring buffer memory <Memory> (0x3186) 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> (0x3180) is set to 1 (enable data logger), the state changes to 'WAIT'.	State stays in 'READY' until <Control> (0x3180) is set to 1 (enable data logger).	

Table 100: States of the data logger state machine

Transition (TR)	Description
TR0	Start program.
TR1	Triggered by setting <Control> (0x3180) to 1 (enable data logger).
TR2	Trigger condition is active.
TR3	Data logger ring buffer is full.
TR4	Triggered by setting <Control> (0x3180) to 1 (enable data logger).
TR5	One of the following parameters has changed: <Divider> (0x3182) <EnableChannel1…4> (0x3184) <ChannelParameter1…4> (0x3185) <TriggerType> (0x3188) <TriggerParameter> (0x3189)

Table 101: Transitions of the data logger state machine

7.17.1.1 Object 0x3180: Control

This parameter enables the data logger.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3180	0	Control	UINT8	rw	N	0…1	None

Value description

<Control>	Description
0	Disable data logger.
1	Enable data logger.

Table 102: Possible values of parameter <Control> (0x3180)

7.17.1.2 Object 0x3181: Status

State of the data logger state machine.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3181	0	Status	UINT8	ro	-	0…3	None

Value description

<Status>	Description
0	'INIT'
1	'WAIT'
2	'SAMPLING'
3	'READY'

Table 103: Possible values of parameter <Status> (0x3181)

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 0x3185: Channel parameter

The parameters <ChannelParameter1…4> define the parameters which shall be sampled. The values are composed of index, sub-index and bit length of the chosen parameter.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3185	1	ChannelParameter1	UINT32	rw	Y	UINT32	0x63100110
0x3185	2	ChannelParameter2	UINT32	rw	Y	UINT32	0x63010110
0x3185	3	ChannelParameter3	UINT32	rw	Y	UINT32	0x63900110
0x3185	4	ChannelParameter4	UINT32	rw	Y	UINT32	0x63810110

Value description

<ChannelParameter1…4>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	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> (0x6310), sub-index 0x01 with a length of 16 bit (16=0x10).

7.17.2.2 Object 0x3184: Enable channel

Any channel can be switched on or off with this parameter.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3184	1	EnableParameter1	UINT8	rw	Y	0…1	0
0x3184	2	EnableParameter2	UINT8	rw	Y	0…1	0
0x3184	3	EnableParameter3	UINT8	rw	Y	0…1	0
0x3184	4	EnableParameter4	UINT8	rw	Y	0…1	0

Value description

<EnableParameter1…4>	Description
0	Channel disabled.
1	Channel enabled.

Table 104: Possible values of parameter <EnableParameter> (0x3184)

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> (0x3182).

$$\text{New sample frequency} = \frac{\text{Maximum sample frequency}}{\text{<Divider> (0x3182)}}$$

The measuring time is increased by the factor <Divider> (0x3182).

7.17.3.1 Object 0x3182: Divider

This parameter contains an integer number to reduce the sampling frequency.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3182	0	Divider	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 105: Possible values of parameter <Divider> (0x3182)

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 0x3189: Trigger parameter

The <TriggerParameter> (0x3189) defines the parameter which is used as trigger signal.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3189	0	TriggerParameter	UINT32	rw	Y	UINT32	0x63100110

Value description

<TriggerParameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	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> (0x6310), sub-index 0x01 with a length of 16 bit (16=0x10).

7.17.4.2 Object 0x3188: Trigger type

This parameter contains the trigger type of the data logger.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3188	0	TriggerType	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 106: Possible values of parameter <TriggerType> (0x3188)

7.17.4.3 Object 0x318C: Trigger level or bitmask

This parameter contains the trigger level if the <TriggerCoupling> (0x318A) is set to 0 or 1.

This parameter contains the BITMASK if the <TriggerCoupling> (0x318A) is set to 2. The BITMASK selects the bits to be compared with the trigger signal.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x318C	0	TriggerLevelOrBitmask	INT32	rw	Y	INT32	0

7.17.4.4 Object 0x318A: Trigger coupling

This parameter contains the trigger coupling type of the data logger.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x318A	0	TriggerCoupling	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 107: Possible values of parameter <TriggerCoupling> (0x318A)

7.17.4.5 Object 0x318B: Trigger slope

The <TriggerSlope> (0x318B) defines the edge of the signal which starts the sampling procedure.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x318B	0	TriggerSlope	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 108: Possible values of parameter <TriggerSlope> (0x318B)

7.17.4.6 Object 0x318D: Trigger position

The <TriggerPosition> (0x318D) is provided as number of samples which shifts the starting point in the ring buffer.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x318D	0	TriggerPosition	INT32	rw	Y	INT32	0

Value description

<TriggerPosition>	Description
> 0	Post trigger.
= 0	No delay.
< 0	Pre trigger.

Table 109: Possible values of parameter <TriggerPosition> (0x318D)

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> (0x3187) until the byte 2047 and continue from byte 0 to byte <SampleStartOffset> (0x3187) – 1.
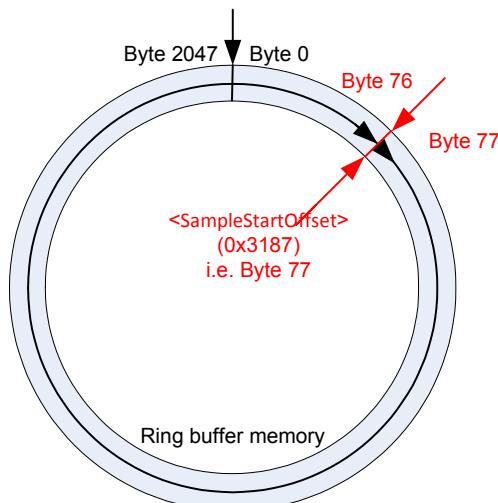
Figure 148: Data memory organization

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> (0x3182) is set to 1 (10000 samples per second), the memory is filled in 292/10000 = 29.2 ms.
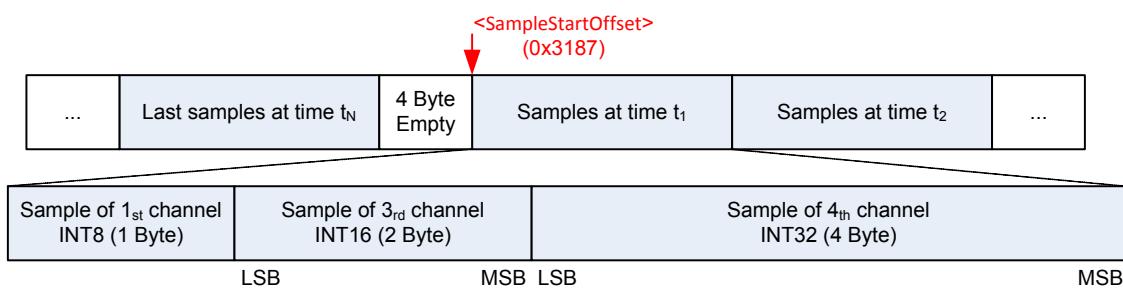
Figure 149: Data memory - mixed channel data

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> (0x3182) is set to 1 (10000 samples per second), the memory is filled in 2048/10000 = 204.8 ms.
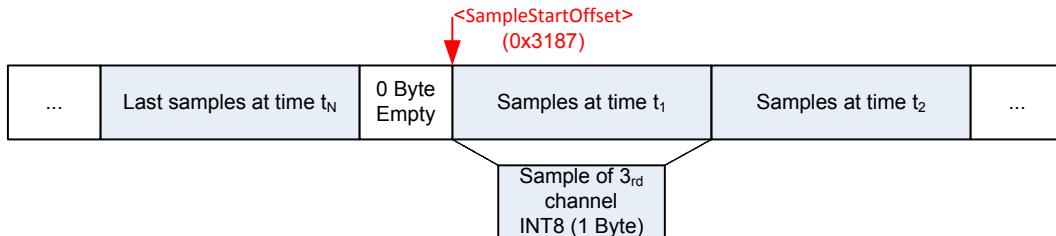
Figure 150: Data memory - one channel with INT8 parameter

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> (0x3182) is set to 1 (10000 samples per second), the memory is filled in 128/10000 = 12.8 ms.
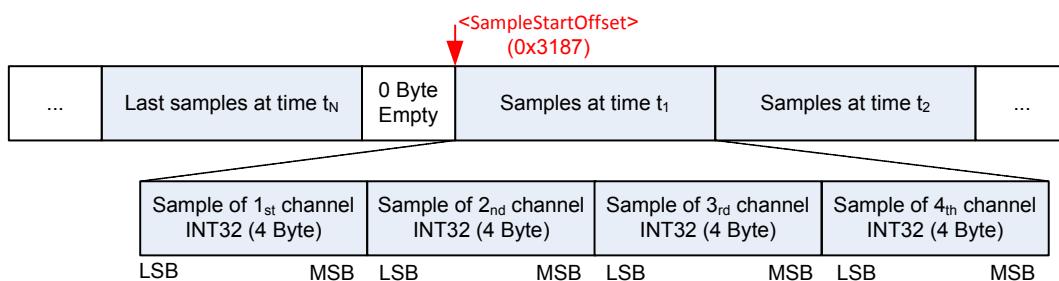
Figure 151: Data memory - four channels with INT32 parameters

7.17.5.1 Object 0x3186: Memory

The parameter <Memory> (0x3186) contains the sampled information of the four channels. The parameter is an array of UINT8 with 2048 entries/bytes.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3186	0	Memory	DOMAIN	ro	-	None	None

7.17.5.2 Object 0x3187: Sample start offset

The <SampleStartOffset> (0x3187) contains the byte position where the recorded data start. It indicates the position of the first sample point.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3187	0	SampleStartOffset	UINT32	ro	-	UINT32	None

7.17.5.3 Object 0x3183: Number of samples

The parameter <NumberOfSamples> (0x3183) contains the number of sample points. On each sample point the data of all active channels are recorded.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3183	0	NumberOfSamples	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 0x3101: Output signal", page 264
- 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 0x3102: Square output (Trigger signal)", page 265

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)
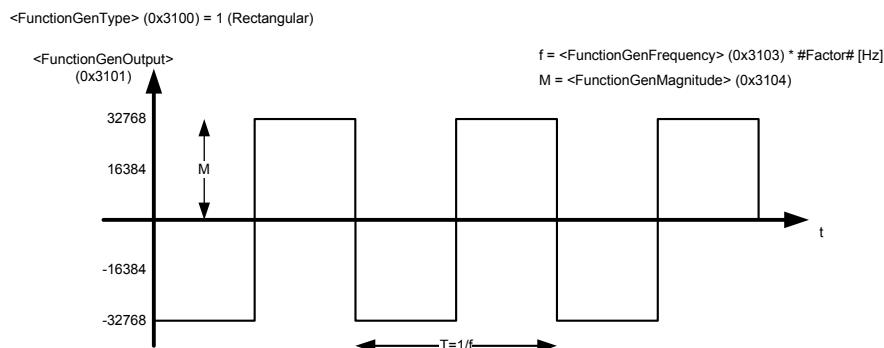
Figure 152: Rectangular output signal (type 1)

7.18.1.2 Triangle output signal (type 2)
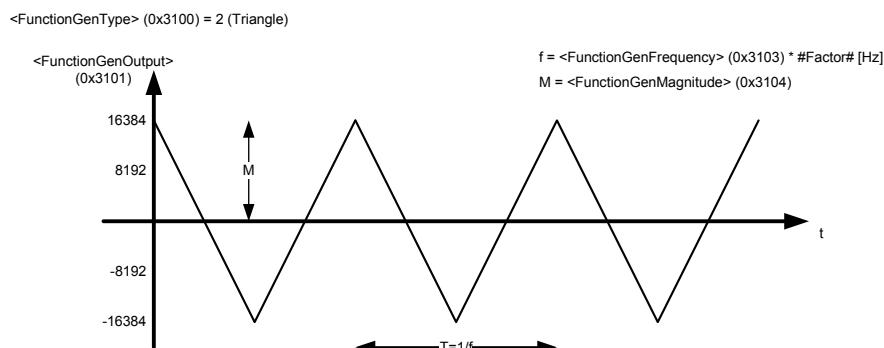
Figure 153: Triangle output signal (type 2)

7.18.1.3 Sawtooth signal (type 3)
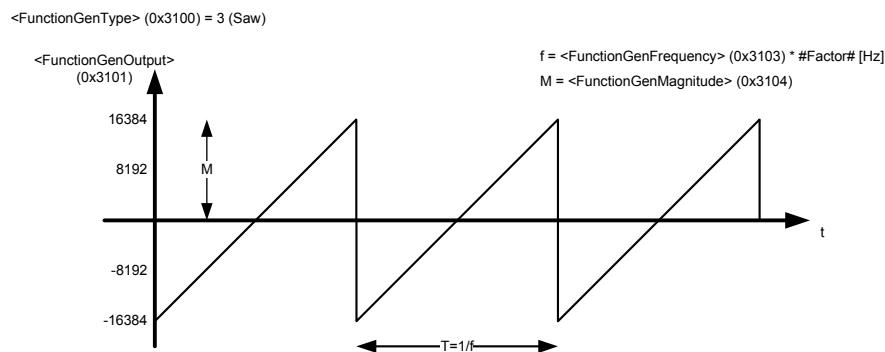
Figure 154: Saw signal (type 3)

ℹ️ The output signal can be inverted by setting the parameter <FunctionGenSign> (0x3107) to –1.
⇨ Chapter "7.18.1.9 Object 0x3107: Sign", page 263

7.18.1.4 Trapezoid signal (type 4)
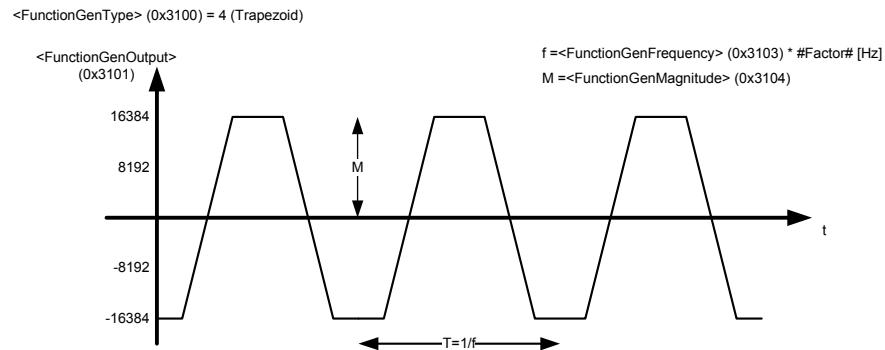
Figure 155: Trapezoid signal (type 4)

7.18.1.5 Sine signal (type 5)
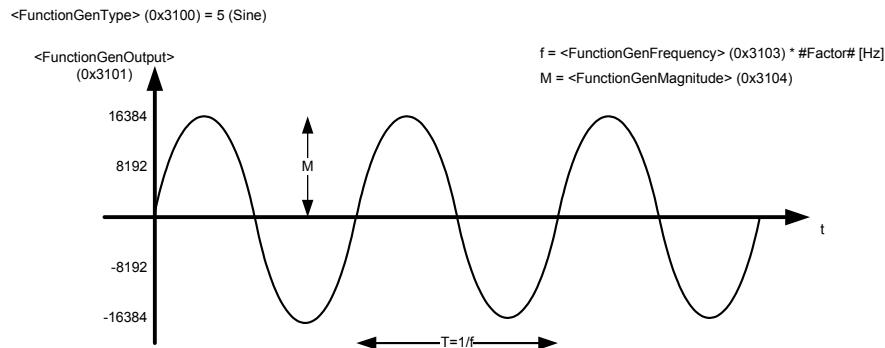
Figure 156: Sine signal (type 5)

7.18.1.6 Object 0x3100: Type

This parameter defines the function generator output signal shape.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3100	0	FunctionGenType	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 110: Possible values of parameter <Type> (0x3100)

7.18.1.7 Object 0x3104: Magnitude

This parameter is the magnitude of the function generator output signal in increments. To configure e.g. a setpoint value of ±100 %, 16384 has to be set as magnitude.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3104	0	FunctionGenMagnitude	INT16	rw	N	0…32767	0

7.18.1.8 Object 0x3105: Offset

This parameter is the offset of the function generator output signal in increments.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3105	0	FunctionGenOffset	INT16	rw	N	INT16	0

7.18.1.9 Object 0x3107: Sign

This parameter is the sign of the function generator output signal.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3107	0	FunctionGenSign	INT8	rw	Y	−1…1	1

7.18.2 Function generator output signal frequency

The function generator output frequency is defined as:

f = <FunctionGenFrequency> (0x3103) · #Factor#

The #Factor# is dependent on the frequency prefix parameter <FunctionGenFrequencyPrefix> (0x3108). The Frequency unit is Hertz [Hz] or [1/s].

7.18.2.1 Object 0x3103: 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							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3103	0	FunctionGenFrequency	UINT16	rw	Y	1…10000	10

7.18.2.2 Object 0x3108: Frequency prefix

This parameter defines a factor which is multiplied with the parameter <FunctionGenFrequency> (0x3103) to get the function generator output signals frequency in Hertz [Hz] or [1/s].

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3108	0	FunctionGenFrequencyPrefix	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 111: Possible values of parameter <FunctionGenFrequencyPrefix> (0x3108)

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> (0x3320)
  ⇨ Chapter "6.2.3.2 Object 0x3320: Setpoint parameter", page 60

- **Pressure setpoint value:** <PrsSetpointParameter> (0x3310)
  ⇨ Chapter "6.2.4.2 Object 0x3310: Setpoint parameter", page 62

7.18.3.1 Object 0x3101: Output signal

This is the function generator output signal.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3101	0	FunctionGenOutput	INT16	ro	-	INT16	None

7.18.3.2 Object 0x3102: 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							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3102	0	FunctionGenSquareOutput	INT16	ro	-	INT16	None

Value description
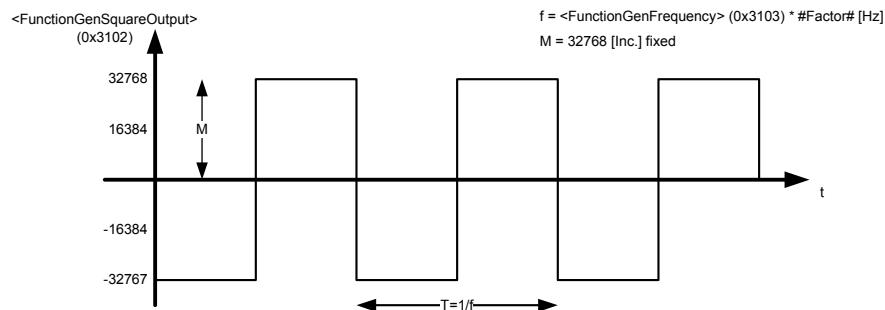
Figure 157: Trigger signal

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> (0x2831) and <FaultRetainStatus> (0x2834) parameters in bit coded form. If no fault reaction for this fault is defined within the parameter <FaultReactionType> (0x2830) 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 <Error-Register> (0x1001) 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> (0x1003) which holds the last eight thrown faults. Afterwards the error description string is saved in the array <FaultReactionDescription> (0x2832). The <FaultHistoryNumber> (0x2833) 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.

8.1.1 Fault reaction flow chart
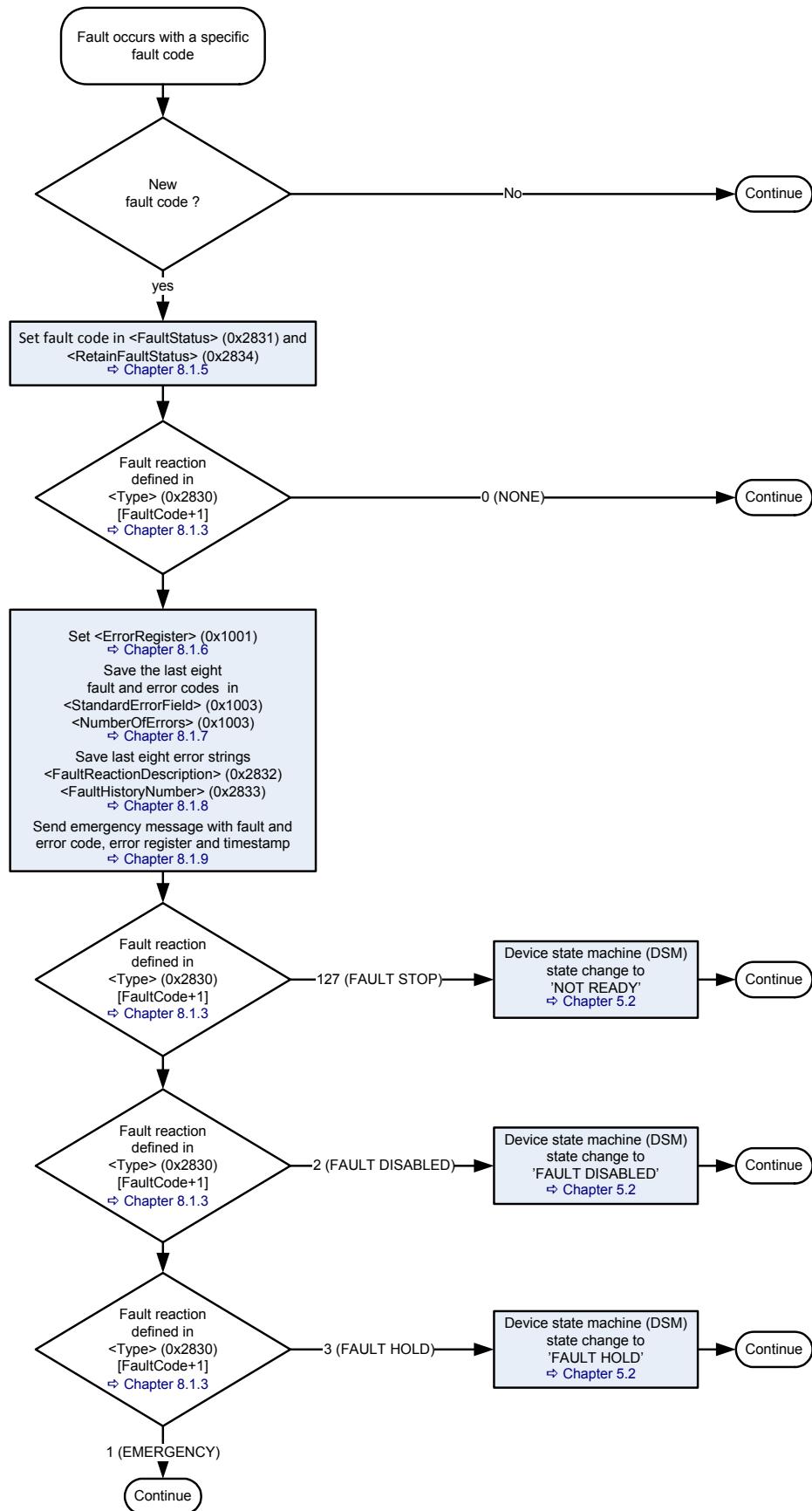
Figure 158: Fault reaction flow chart

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 sub-index of the fault reaction type <FaultReactionType> (0x2830), the fault code must be incremented by one.

Fault code			Dword		<FaultReactionType> (0x2830)		
Dec.	Hex.	Fault description	Num	Bit	Sub-Index	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 112: Possible fault codes (part 1 of 4)

Fault code			Dword		<FaultReactionType> (0x2830)		
Dec.	Hex.	Fault description	Num	Bit	Sub-Index	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 112: Possible fault codes (part 2 of 4)

Fault code			Dword		<FaultReactionType> (0x2830)		
Dec.	Hex.	Fault description	Num	Bit	Sub-Index	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 112: Possible fault codes (part 3 of 4)

Fault code			Dword		<FaultReactionType> (0x2830)		
Dec.	Hex.	Fault description	Num	Bit	Sub-Index	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 112: Possible fault codes (part 4 of 4)

8.1.3 Fault reaction type

The fault reaction parameter <FaultReactionType> (0x2830) can be used to configure the fault behavior for each fault code.

8.1.3.1 Object 0x2830: 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 ndex of the fault reaction type <FaultReactionType> (0x2830), the fault code must be incremented by 101.

FaultReaction							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2830	1…128	FaultReactionType	INT8	rw	Y	⇨ Table 112, page 268	

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 113: 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 232

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 114: 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 114: 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 114: 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 0x2831: Fault status

Actual reported faults in bit coded form.

FaultReaction							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2831	1	FaultStatus (fault code 1…31)	UINT32	ro	-	UINT32	None
0x2831	2	FaultStatus (fault code 32…63)	UINT32	ro	-	UINT32	None
0x2831	3	FaultStatus (fault code 64…95)	UINT32	ro	-	UINT32	None
0x2831	4	FaultStatus (fault code 96…128)	UINT32	ro	-	UINT32	None

8.1.5.2 Object 0x2834: Fault retain status

All reported faults in bit coded form since powering on the servo valve.

FaultReaction							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2834	1	FaultRetainStatus (fault code 1…31)	UINT32	rw	N	UINT32	None
0x2834	2	FaultRetainStatus (fault code 32…63)	UINT32	rw	N	UINT32	None
0x2834	3	FaultRetainStatus (fault code 64…95)	UINT32	rw	N	UINT32	None
0x2834	4	FaultRetainStatus (fault code 96…128)	UINT32	rw	N	UINT32	None

8.1.6 Error register

The <ErrorRegister> (0x1001) displays the error information about the last reported fault in bit-coded form. Bit 0 of the <ErrorRegister> (0x1001) is set as soon as an error occurs on the servo valve.

ⓘ The error codes of older faults are stored in the <PredefinedErrorField> (0x1003).
⇨ Chapter "8.1.7 Last eight fault codes and error codes", page 276

8.1.6.1 Object 0x1001: Error register

Device							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1001	0	ErrorRegister	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 115: Possible values of parameter <ErrorRegister> (0x1001)

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> (0x1003) parameter array. The <StandardErrorField> (0x1003) parameter array contains a list of up to 8 entries. This error code provides information about the reason of the error. The parameter <NumberOfErrors> (0x1003) holds information about the number of errors currently recorded. Every new error is stored in the first element of the parameter array <StandardErrorField> (0x1003), 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> (0x1003), sub-index 0, deletes the entire error code entries.

ⓘ Only the first eight elements of the parameter array <StandardErrorField> (0x1003), sub-index 1…8 are used.

8.1.7.1 Object 0x1003: Predefined error field

This object contains the last eight error codes, fault codes and the number of recorded errors.

Device							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1003	0	NumberOfErrors	UINT32	rw	N	UINT32	0
0x1003	1…16	StandardErrorField	UINT32	ro	-	UINT32	0

Value description

Parameter	Description
<NumberOfErrors>	Number of actual recorded errors.
<StandardErrorField>	Array of recorded errors.

Table 116: Possible values of parameter <PreDefinedErrorField> (0x1003)

<StandardErrorField>				
Byte	3	2	1	0
Description	Additional information		Error code	
	Reserved	Fault code ⇨ Chapter "8.1.2 Possible fault codes", page 268	⇨ Chapter "8.1.4 Error codes depending on fault codes", page 272	

Example

The parameter <StandardErrorField> (0x1003), sub-index 1 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> (0x2832). The parameter <FaultHistoryNumber> (0x2833) selects one of the last eight error description strings. The newest error description string is shown if the <FaultHistoryNumber> (0x2833) is set to zero and the oldest saved error description string is shown if the <FaultHistoryNumber> (0x2833) is set equal to the parameter <NumberOfErrors> (0x1003).

8.1.8.1 Object 0x2832: Fault reaction description

The parameter <FaultReactionDescription> (0x2832) contains the fault reaction description string depending on the <FaultHistoryNumber> (0x2833) 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							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2832	0	FaultReactionDescription	STRING	ro	-	None	None

8.1.8.2 Object 0x2833: Fault history number

The parameter <FaultHistoryNumber> (0x2833) selects the fault description shown in the parameter <FaultReactionDescription> (0x2832).

FaultReaction							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2833	0	FaultHistoryNumber	UINT8	rw	N	0…7	None

8.1.9 Emergency message

Every time a configured error occurs on the servo valve, it sends an emergency message with error register, error code and timestamp to the master. The emergency 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 "2.8.2 Emergency (EMCY) protocol (COB-ID: 0x080+Node-ID)", page 17
⇨ Chapter "8.1.4 Error codes depending on fault codes", page 272

The coding of the emergency message is as follows:

Byte	7	6	5	4	3	2	1	0
Description	Moog-specific error code					Error register	Emergency error code ⇨ Chapter "8.1.4 Error codes depending on fault codes", page 272	
	Power on time in minutes				Fault code ⇨ Chapter "8.1.2 Possible fault codes", page 268			

ⓘ The displayed byte order of the transmitted emergency message is depending on the field bus master.

Example

In the example below a simple reproducible fault is described. The fault code 0x05 (power supply voltage too low) will occur if the power supply voltage is less than 17 V. The corresponding error code is 0x3412 and the <ErrorRegister> (0x1001) is set to 0x4. The time since power on of the servo valve until the fault occurred is 1000 or in hex 0x03E8 minutes.

Byte	7	6	5	4	3	2	1	0
Description	Moog-specific error code					Error register	Emergency error code	
	Power on time in minutes				Fault code			
Content	0x00	0x00	0x03	0xE8	0x05	0x04	0x34	0x12
Result	0x000003E805043412							

8.1.10 Fault disappears

If all faults has disappeared the <ErrorRegister> (0x1001) and the <FaultStatus> (0x2831) 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).

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.

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).

> ⓘ 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> (0x2830).

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 0x2822: Internal error code

This object contains the first five occurred internal error codes since the firmware reset.

ErrorHandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2822	1…5	InternalErrorCode	UINT32	ro	-	UINT32	0

8.2.2 Object 0x2823: 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							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2823	1…5	InternalErrorTime	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 117: 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 237

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 239

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 235

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 233

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 102
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 102
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 102
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 102
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.
	⇨ Chapter "2.8.3 Process data object (PDO) protocol", page 18
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 297
	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.
	⇨ Chapter "2.8.3 Process data object (PDO) protocol", page 18
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.
	⇨ Chapter "2.8.3 Process data object (PDO) protocol", page 18
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.
	⇨ Chapter "2.8.3 Process data object (PDO) protocol", page 18
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. ⇨ Chapter "2.8.3 Process data object (PDO) protocol", page 18
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. ⇨ Chapter "2.9.8 Node guarding (COB-ID: 0x700, RTR:1)", page 36
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. ⇨ Chapter "2.8.1 Synchronization (SYNC) protocol (COB-ID: 0x080)", page 16
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. ⇨ Chapter "2.8.1 Synchronization (SYNC) protocol (COB-ID: 0x080)", page 16 ⇨ Chapter "2.8.3 Process data object (PDO) protocol", page 18
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.

(i) 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 296
- 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 297

ⓘ   Parameters are stored and restored in accordance with the procedure described in the common device profile CiA 301.

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 118: 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 0x1010: 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							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1010	1	SaveAllParameters	UINT32	rw	N	UINT32	None
0x1010	2	SaveCommunicationParameters	UINT32	rw	N	UINT32	None
0x1010	3	SaveApplicationParameters	UINT32	rw	N	UINT32	None
0x1010	4	SaveManufacturerDefinedParameters	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 119: Possible values of parameter 0x1010

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 0x1011: Restore default parameters

The factory settings can be restored by writing the signature 0x64616F6C ("load") to one the following parameters.

StoreParameters							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1011	1	RestoreAllDefaultParameters	UINT32	rw	N	UINT32	None
0x1011	2	RestoreCommunicationDefaultParameters	UINT32	rw	N	UINT32	None
0x1011	3	RestoreApplicationDefaultParameters	UINT32	rw	N	UINT32	None
0x1011	4	RestoreManufacturerDefinedDefaultParameters	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 120: Possible values of parameter 0x1011

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 46
⇨ Chapter "2.9 Network management state machine (NMT state machine)", page 32

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 121: 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.

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x0002	0	Y	dums08	CiA 301	DataType, DummyDataS08 DummyDataS08	INT8	rw	N	INT8	-
0x0003	0	Y	dums16	CiA 301	DataType, DummyDataS16 DummyDataS16	INT16	rw	N	INT16	-
0x0004	0	Y	dums32	CiA 301	DataType, DummyDataS32 DummyDataS32	INT32	rw	N	INT32	-
0x0005	0	Y	dumu08	CiA 301	DataType, DummyDataU08 DummyDataU08	UINT8	rw	N	UINT8	-
0x0006	0	Y	dumu16	CiA 301	DataType, DummyDataU16 DummyDataU16	UINT16	rw	N	UINT16	-
0x0007	0	Y	dumu32	CiA 301	DataType, DummyDataU32 DummyDataU32	UINT32	rw	N	UINT32	-
0x0008	0	Y	dumf32	CiA 301	DataType, DummyDataF32 DummyDataF32	FLOAT32	rw	N	FLOAT32	-
0x0009	0	N	dumchr	CiA 301	DataType, DummyDataVisibleString DummyDataVisibleString	CHAR[64]	rw	N	-	-
0x1000	0	N	devtyp	CiA 301	Device, DeviceType DeviceType	UINT32	ro	-	UINT32	408
0x1001	0	Y	errreg	CiA 301	Device, ErrorRegister ErrorRegister	UINT8	ro	-	UINT8	-
0x1002	0	Y	manststsreg	CiA 301	Device, ManufacturerStatusRegister ManufacturerStatusRegister	UINT32	ro	-	UINT32	-
0x1003	0	N	preerrnum	CiA 301	Device, NumberOfErrors NumberOfErrors	UINT32	rw	N	UINT32	0
0x1003	1…16	N	preerrfld[0…15]	CiA 301	Device, StandardErrorField StandardErrorField	UINT32	ro	-	UINT32	-
0x1005	0	N	sncmsgcob	CiA 301	CAN, CobIdSyncMessage CobIdSyncMessage	UINT32	rw	Y	1…0x400007FF	<NodeID> (0x100B) + 0x0001
0x1006	0	N	comcycper	CiA 301	CAN, CommunicationCyclePeriod CommunicationCyclePeriod	UINT32	rw	Y	UINT32	-

Table 122: Object dictionary (part 1 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name / Parameter name	Data type	Access	Persistence	Value range	Default
0x1007	0	N	sncwinlen	CiA 301	CAN, SynchronousWindowLength / SynchronousWindowLength	UINT32	rw	Y	UINT32	-
0x1008	0	N	mandevnam	CiA 301	Device, ManufacturerDeviceName / ManufacturerDeviceName	CHAR[64]	ro	-	-	-
0x1009	0	N	manhdwver	CiA 301	Device, ManufacturerHardwareVersion / ManufacturerHardwareVersion	CHAR[64]	ro	-	-	-
0x100A	0	N	mansfwver	CiA 301	Device, ManufacturerSoftwareVersion / ManufacturerSoftwareVersion	CHAR[64]	ro	-	-	B99224-DV018-D-211a
0x100B	0	N	nodide	CiA 301	CAN, NodeId / NodeId	UINT32	ro	-	UINT32	0x7F
0x100C	0	N	grdtim	CiA 301	CAN, GuardTime / GuardTime	UINT16	rw	Y	UINT16	-
0x100D	0	N	liftimfct	CiA 301	CAN, LifeTimeFactor / LifeTimeFactor	UINT8	rw	Y	UINT8	-
0x1010	1	N	stopar[0]	CiA 301	Device, StoreParameters / SaveAllParameters	UINT32	rw	N	UINT32	1
0x1010	2	N	stopar[1]	CiA 301	Device, StoreParameters / SaveCommunicationParameters	UINT32	rw	N	UINT32	1
0x1010	3	N	stopar[2]	CiA 301	Device, StoreParameters / SaveApplicationParameters	UINT32	rw	N	UINT32	1
0x1010	4	N	stopar[3]	CiA 301	Device, StoreParameters / SaveManufacturerDefinedParameters	UINT32	rw	N	UINT32	1
0x1011	1	N	rstpar[0]	CiA 301	Device, RestoreDefaultParameters / RestoreAllDefaultParameters	UINT32	rw	N	UINT32	1
0x1011	2	N	rstpar[1]	CiA 301	Device, RestoreDefaultParameters / RestoreCommunicationDefaultParameters	UINT32	rw	N	UINT32	1
0x1011	3	N	rstpar[2]	CiA 301	Device, RestoreDefaultParameters / RestoreApplicationDefaultParameters	UINT32	rw	N	UINT32	1
0x1011	4	N	rstpar[3]	CiA 301	Device, RestoreDefaultParameters / RestoreManufacturerDefinedDefaultParameters	UINT32	rw	N	UINT32	1
0x1012	0	N	tspmsgcob	CiA 301	CAN, CobIdTimeStampMessage / CobIdTimeStampMessage	UINT32	rw	Y	1...0x7FF	<NodeID> (0x100B) + 0x0081
0x1013	0	Y	hghrsltsp	CiA 301	CAN, HighResolutionTimeStamp / HighResolutionTimeStamp	UINT32	rw	Y	UINT32	-
0x1014	0	N	emgmsgcob	CiA 301	CAN, CobIdEmergencyMessage / CobIdEmergencyMessage	UINT32	rw	Y	1...0x7FF	<NodeID> (0x100B) + 0x0080

Table 122: Object dictionary (part 2 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x1015	0	N	emgmsginh	CiA 301	CAN, InhibitTimeEmergencyMessage / InhibitTimeEmergencyMessage	UINT16	rw	Y	UINT16	-
0x1017	0	N	proharbea	CiA 301	CAN, ProducerHeartbeatTime / ProducerHeartbeatTime	UINT16	rw	Y	UINT16	-
0x1018	1	N	ideobj[0]	CiA 301	Device, IdentityObject / VendorId	UINT32	ro	-	0x28...0x28	0x28
0x1018	2	N	ideobj[1]	CiA 301	Device, IdentityObject / ProductCode	UINT32	ro	-	0...0xFFFFFFFF	0
0x1018	3	N	ideobj[2]	CiA 301	Device, IdentityObject / RevisionNumber	UINT32	ro	-	0...0xFFFFFFFF	0
0x1018	4	N	ideobj[3]	CiA 301	Device, IdentityObject / SerialNumber	UINT32	ro	-	0...0xFFFFFFFF	0
0x1200	1	N	sdoclncob	CiA 301	CAN, CobIDSDO / CobIDSDO	UINT32	ro	-	UINT32	<NodeID> (0x100B) + 0x0600
0x1200	2	N	sdosrvcob	CiA 301	CAN, CobIDSDO / CobIDSDO	UINT32	ro	-	UINT32	<NodeID> (0x100B) + 0x0580
0x1400	1	N	pdrcob[0]	CiA 301	CAN, RPdo_CobIdUsedByPdo / RPdo1_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0200
0x1400	2	N	pdrtrn[0]	CiA 301	CAN, RPdo_TransmissionType / RPdo1_TransmissionType	UINT8	rw	Y	UINT8	255
0x1400	5	N	pdrtim[0]	CiA 301	CAN, RPdo_EventTimer / RPdo1_EventTimer	UINT16	rw	Y	UINT16	-
0x1401	1	N	pdrcob[1]	CiA 301	CAN, RPdo_CobIdUsedByPdo / RPdo2_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0300
0x1401	2	N	pdrtrn[1]	CiA 301	CAN, RPdo_TransmissionType / RPdo2_TransmissionType	UINT8	rw	Y	UINT8	255
0x1401	5	N	pdrtim[1]	CiA 301	CAN, RPdo_EventTimer / RPdo2_EventTimer	UINT16	rw	Y	UINT16	-
0x1402	1	N	pdrcob[2]	CiA 301	CAN, RPdo_CobIdUsedByPdo / RPdo3_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0400
0x1402	2	N	pdrtrn[2]	CiA 301	CAN, RPdo_TransmissionType / RPdo3_TransmissionType	UINT8	rw	Y	UINT8	255
0x1402	5	N	pdrtim[2]	CiA 301	CAN, RPdo_EventTimer / RPdo3_EventTimer	UINT16	rw	Y	UINT16	-

Table 122: Object dictionary (part 3 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x1403	1	N	pdrcob[3]	CiA 301	CAN, RPdo_CobIdUsedByPdo RPdo4_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	\<NodeID\> (0x100B) + 0x0500
0x1403	2	N	pdrtrn[3]	CiA 301	CAN, RPdo_TransmissionType RPdo4_TransmissionType	UINT8	rw	Y	UINT8	255
0x1403	5	N	pdrtim[3]	CiA 301	CAN, RPdo_EventTimer RPdo4_EventTimer	UINT16	rw	Y	UINT16	-
0x1600	0	N	pdrmapnum	CiA 301	CAN, RPdo_Mapping RPdo_Mapping	UINT8	rw	Y	0…8	1
0x1600	1…8	N	pdrmap[0…7]	CiA 301	CAN, RPdo_Mapping RPdo1_ApplicPara1	UINT32	rw	Y	UINT32	0x60400010
0x1601	0	N	pdrmapnum	CiA 301	CAN, RPdo_Mapping RPdo_Mapping	UINT8	rw	Y	0…8	2
0x1601	1…8	N	pdrmap[8…15]	CiA 301	CAN, RPdo_Mapping RPdo2_ApplicPara1	UINT32	rw	Y	UINT32	0x60400010
0x1602	0	N	pdrmapnum	CiA 301	CAN, RPdo_Mapping RPdo_Mapping	UINT8	rw	Y	0…8	2
0x1602	1…8	N	pdrmap[16…23]	CiA 301	CAN, RPdo_Mapping RPdo3_ApplicPara1	UINT32	rw	Y	UINT32	0x60400010
0x1603	0	N	pdrmapnum	CiA 301	CAN, RPdo_Mapping RPdo_Mapping	UINT8	rw	Y	0…8	3
0x1603	1…8	N	pdrmap[24…31]	CiA 301	CAN, RPdo_Mapping RPdo4_ApplicPara1	UINT32	rw	Y	UINT32	0x60400010
0x1800	1	N	pdtcob[0]	CiA 301	CAN, TPdo_CobIdUsedByPdo TPdo1_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	\<NodeID\> (0x100B) + 0x0180
0x1800	2	N	pdttrn[0]	CiA 301	CAN, TPdo_TransmissionType TPdo1_TransmissionType	UINT8	rw	Y	UINT8	255
0x1800	3	N	pdtinh[0]	CiA 301	CAN, TPdo_InhibitTime TPdo1_InhibitTime	UINT16	rw	Y	UINT16	-
0x1800	5	N	pdttim[0]	CiA 301	CAN, TPdo_EventTimer TPdo1_EventTimer	UINT16	rw	Y	UINT16	-
0x1801	1	N	pdtcob[1]	CiA 301	CAN, TPdo_CobIdUsedByPdo TPdo2_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	\<NodeID\> (0x100B) + 0x0280
0x1801	2	N	pdttrn[1]	CiA 301	CAN, TPdo_TransmissionType TPdo2_TransmissionType	UINT8	rw	Y	UINT8	255
0x1801	3	N	pdtinh[1]	CiA 301	CAN, TPdo_InhibitTime TPdo2_InhibitTime	UINT16	rw	Y	UINT16	-

Table 122: Object dictionary (part 4 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x1801	5	N	pdttim[1]	CiA 301	CAN, TPdo_EventTimer TPdo2_EventTimer	UINT16	rw	Y	UINT16	-
0x1802	1	N	pdtcob[2]	CiA 301	CAN, TPdo_CobIdUsedByPdo TPdo3_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0380
0x1802	2	N	pdttrn[2]	CiA 301	CAN, TPdo_TransmissionType TPdo3_TransmissionType	UINT8	rw	Y	UINT8	255
0x1802	3	N	pdtinh[2]	CiA 301	CAN, TPdo_InhibitTime TPdo3_InhibitTime	UINT16	rw	Y	UINT16	-
0x1802	5	N	pdttim[2]	CiA 301	CAN, TPdo_EventTimer TPdo3_EventTimer	UINT16	rw	Y	UINT16	-
0x1803	1	N	pdtcob[3]	CiA 301	CAN, TPdo_CobIdUsedByPdo TPdo4_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0480
0x1803	2	N	pdttrn[3]	CiA 301	CAN, TPdo_TransmissionType TPdo4_TransmissionType	UINT8	rw	Y	UINT8	255
0x1803	3	N	pdtinh[3]	CiA 301	CAN, TPdo_InhibitTime TPdo4_InhibitTime	UINT16	rw	Y	UINT16	-
0x1803	5	N	pdttim[3]	CiA 301	CAN, TPdo_EventTimer TPdo4_EventTimer	UINT16	rw	Y	UINT16	-
0x1A00	0	N	pdtmapnum	CiA 301	CAN, TPdo_Mapping TPdo_Mapping	UINT8	rw	Y	0...8	1
0x1A00	1...8	N	pdtmap[0...7]	CiA 301	CAN, TPdo_Mapping TPdo1_ApplicPara1	UINT32	rw	Y	UINT32	0x60410010
0x1A01	0	N	pdtmapnum	CiA 301	CAN, TPdo_Mapping TPdo_Mapping	UINT8	rw	Y	0...8	2
0x1A01	1...8	N	pdtmap[8...15]	CiA 301	CAN, TPdo_Mapping TPdo2_ApplicPara1	UINT32	rw	Y	UINT32	0x60410010
0x1A02	0	N	pdtmapnum	CiA 301	CAN, TPdo_Mapping TPdo_Mapping	UINT8	rw	Y	0...8	2
0x1A02	1...8	N	pdtmap[16...23]	CiA 301	CAN, TPdo_Mapping TPdo3_ApplicPara1	UINT32	rw	Y	UINT32	0x60410010
0x1A03	0	N	pdtmapnum	CiA 301	CAN, TPdo_Mapping TPdo_Mapping	UINT8	rw	Y	0...8	3
0x1A03	1...8	N	pdtmap[24...31]	CiA 301	CAN, TPdo_Mapping TPdo4_ApplicPara1	UINT32	rw	Y	UINT32	0x60410010
0x200A	0	N	iopbkpldver	Moog DCV	Software_PiggyBack, ManufacutrerIOPiggybackVersion ManufacutrerIOPiggybackVersion	UINT16	ro	-	UINT16	0

Table 122: Object dictionary (part 5 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name / Parameter name	Data type	Access	Persistence	Value range	Default
0x200F	0	N	pwrdly	Moog DCV	Device, PowerOnDelay / PowerOnDelay	UINT8	rw	Y	0…10	-
0x2149	0	N	stgitfnum	Moog DCV	MainStageControl, MainStageInterface / MainStageInterface	UINT8	rw	Y	0…8	-
0x2158	0	Y	stgposout	Moog DCV	ValveMainStageControl, ControllerOutput / ControllerOutput	INT16	ro	-	INT16	-
0x215B	1	Y	stgsplval	Moog DCV	MainStageControl, MainStagePositionActualValue / MainStagePositionActualValue	INT16	ro	-	INT16	-
0x215C	0	N	stgposcusogn	Moog DCV	MainStageControl, MainStageCustomerOverallGain / MainStageCustomerOverallGain	FLOAT32	rw	Y	0.00…2.00	1.00
0x2171	0	Y	stgposint	Moog DCV	MainStageControl, MainStageIntegralPart / MainStageIntegralPart	INT32	ro	-	INT32	-
0x2300	1…16	Y	cmpprstyp[0…15]	Moog DCV	ValvePressureControl, PressureControllerType / PressureControllerType	UINT8	rw	Y	UINT8	-
0x2301	1…16	N	cmpprsrs[0…15]	Moog DCV	ValvePressureControl, SystemPressure / SystemPressure	INT16	rw	Y	INT16	-
0x2302	1…16	N	cmpprspbr[0…15]	Moog DCV	ValvePressureControl, ReferencePressure / ReferencePressure	INT16	rw	Y	INT16	-
0x2303	1…16	Y	cmpprsrmp[0…15]	Moog DCV	ValvePressureControl, RampSlope / RampSlope	UINT16	rw	Y	UINT16	-
0x2304	1…16	Y	cmpprspgn[0…15]	Moog DCV	ValvePressureControl, ProportionalGain / ProportionalGain	FLOAT32	rw	Y	0.00…+inf	-
0x2305	1…16	Y	cmpprsign[0…15]	Moog DCV	ValvePressureControl, IntegratorGain / IntegratorGain	FLOAT32	rw	Y	0.00…+inf	-
0x2306	1…16	Y	cmpprsigf[0…15]	Moog DCV	ValvePressureControl, IntegratorFactor / IntegratorFactor	FLOAT32	rw	Y	0.00…+inf	0.10
0x2307	1…16	Y	cmpprsicr[0…15]	Moog DCV	ValvePressureControl, IntegratorControlRange / IntegratorControlRange	INT16	rw	Y	0…32767	163
0x2308	1…16	Y	cmpprsdgn[0…15]	Moog DCV	ValvePressureControl, DifferentiatorGain / DifferentiatorGain	FLOAT32	rw	Y	FLOAT32	-
0x2309	1…16	Y	cmpprsdtm[0…15]	Moog DCV	ValvePressureControl, DifferentiatorT1 / DifferentiatorT1	FLOAT32	rw	Y	0.00…+inf	-
0x230A	1…16	Y	cmpprsupp[0…15]	Moog DCV	ValvePressureControl, UpperOutputLimit / UpperOutputLimit	INT16	rw	Y	\<LowerOutputLimit\> (0x230B[n])…32767	16384
0x230B	1…16	Y	cmpprslow[0…15]	Moog DCV	ValvePressureControl, LowerOutputLimit / LowerOutputLimit	INT16	rw	Y	-32768…\<UpperOutputLimit\> (0x230A[n])	-16384
0x230C	1…16	Y	cmpprschy[0…15]	Moog DCV	ValvePressureControl, HydraulicCapacity / HydraulicCapacity	FLOAT32	rw	Y	0.00…+inf	-

Table 122: Object dictionary (part 6 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name / Parameter name	Data type	Access	Persistence	Value range	Default
0x230D	1...16	N	cmpprsitf[0...15]	Moog DCV	ValvePressureControl, ActiveTransducerInterfaceAreaA / ActiveTransducerInterfaceAreaA	INT8	rw	Y	1...8	1
0x230E	1...16	Y	cmpprsptm[0...15]	Moog DCV	ValvePressureControl, ProportionalGainTimeConstant / ProportionalGainTimeConstant	FLOAT32	rw	Y	0.00...+inf	-
0x230F	1...16	N	cmpprsitfb[0...15]	Moog DCV	ValvePressureControl, ActiveTransducerInterfaceAreaB / ActiveTransducerInterfaceAreaB	INT8	rw	Y	0...8	-
0x2310	0	Y	cmpprsint	Moog DCV	ValvePressureControl, KiOutput / KiOutput	FLOAT32	ro	-	FLOAT32	-
0x2311	0	Y	cmpprspro	Moog DCV	ValvePressureControl, KpT1Output / KpT1Output	FLOAT32	ro	-	FLOAT32	-
0x2312	0	Y	cmpprsdt1	Moog DCV	ValvePressureControl, KdOutput / KdOutput	FLOAT32	ro	-	FLOAT32	-
0x2313	0	Y	cmpprsdir	Moog DCV	ValvePressureControl, DirectionalDependentGain / DirectionalDependentGain	FLOAT32	rw	Y	0.00...+inf	1.00
0x231A	1...16	Y	cmpprsiup[0...15]	Moog DCV	ValvePressureControl, IntegratorUpperOutputLimit / IntegratorUpperOutputLimit	INT16	rw	Y	<IntegratorLowerOutputLimit> (0x231B[n])...32767	16384
0x231B	1...16	Y	cmpprsilo[0...15]	Moog DCV	ValvePressureControl, IntegratorLowerOutputLimit / IntegratorLowerOutputLimit	INT16	rw	Y	-32768...<IntegratorUpperOutputLimit> (0x231A[n])	-16384
0x231C	1	Y	prsref	Moog DCV	ValvePressureControl_DemandValueGenerator, PrsReferenceValue / PrsReferenceValue	INT16	rw	N	0...32767	400
0x231C	2	N	prsrefuni	Moog DCV	ValvePressureControl_DemandValueGenerator, PrsReferenceValue / Unit	UINT8	ro	-	UINT8	0x4E
0x231C	3	N	prsrefprf	Moog DCV	ValvePressureControl_DemandValueGenerator, PrsReferenceValue / Prefix	INT8	ro	-	INT8	0
0x231D	0	N	cmpprsignsts	Moog DCV	ValvePressureControl, IntegratorGainStatus / IntegratorGainStatus	UINT8	ro	-	UINT8	-
0x2324	1...16	Y	cmpprsfb-spgn[0...15]	Moog DCV	PressureControl, SpoolPositionFeedBackGain / SpoolPositionFeedBackGain	FLOAT32	rw	Y	0.00...+inf	-
0x2330	0	N	cmpprsitfps	Moog DCV	ValveFlowControl, PressureControllerTransducerInterfacePS / PressureControllerTransducerInterfacePS	INT8	rw	Y	0...8	-
0x2350	0	Y	prssetnum	Moog DCV	ValvePressureControl, ActiveParameterSetNumber / ActiveParameterSetNumber	UINT8	rw	Y	1...16	1
0x2360	1...8	N	prssnsmaxcl-chlp[0...7]	Moog DCV	ValvePressureControl, PressureSensorMaximum / PressureSensorMaximum	UINT16	rw	Y	UINT16	-
0x2361	1...8	N	prssysmaxcl-chlp[0...7]	Moog DCV	ValvePressureControl, PressureSystemMaximum / PressureSystemMaximum	UINT16	rw	Y	UINT16	-
0x2362	1	N	drvsnsresclchlp	Moog DCV	Drive, DriveSensorResolution / DriveSensorResolution	FLOAT32	rw	Y	FLOAT32	-

Table 122: Object dictionary (part 7 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Object name, Block name / Parameter name	Data type	Access	Persistence	Value range	Default
0x23F0	1	N	prsfltbi[0]	Moog DCV	ValvePressureControl, PressureValueFilterBCoeff / PressureValueFilterBCoeff	FLOAT32	rw	Y	FLOAT32	-
0x23F0	2	N	prsfltbi[1]	Moog DCV	ValvePressureControl, PressureValueFilterBCoeff / PressureValueFilterBCoeff	FLOAT32	rw	Y	FLOAT32	-
0x23F0	3	N	prsfltbi[2]	Moog DCV	ValvePressureControl, PressureValueFilterBCoeff / PressureValueFilterBCoeff	FLOAT32	rw	Y	FLOAT32	-
0x23F0	4	N	prsfltbi[3]	Moog DCV	ValvePressureControl, PressureValueFilterBCoeff / PressureValueFilterBCoeff	FLOAT32	rw	Y	FLOAT32	-
0x23F1	1	N	prsfltai[0]	Moog DCV	ValvePressureControl, PressureValueFilterACoeff / PressureValueFilterACoeff	FLOAT32	rw	Y	FLOAT32	-
0x23F1	2	N	prsfltai[1]	Moog DCV	ValvePressureControl, PressureValueFilterACoeff / PressureValueFilterACoeff	FLOAT32	rw	Y	FLOAT32	-
0x23F1	3	N	prsfltai[2]	Moog DCV	ValvePressureControl, PressureValueFilterACoeff / PressureValueFilterACoeff	FLOAT32	rw	Y	FLOAT32	-
0x23F1	4	N	prsfltai[3]	Moog DCV	ValvePressureControl, PressureValueFilterACoeff / PressureValueFilterACoeff	FLOAT32	rw	Y	FLOAT32	-
0x23F2	0	N	cmpprsflt	Moog DCV	ValvePressureControl, ActualPressureFilterCutoffFrequency / ActualPressureFilterCutoffFrequency	FLOAT32	rw	Y	0.00...3333.33	-
0x23F3	0	N	prsfltord	Moog DCV	ValvePressureControl, ActualPressureFilterOrder / ActualPressureFilterOrder	UINT8	rw	Y	0...3	1
0x2416	0	N	cmpsplint	Moog DCV	ValvePositionControl, IntegratorTestValue / IntegratorTestValue	INT32	ro	-	INT32	-
0x2418	0	N	cmpprsout	Moog DCV	ValvePressureControl, ControllerOutput / ControllerOutput	INT16	ro	-	INT16	-
0x241F	0	N	cmpsplcusogn	Moog DCV	ValvePositionControl, CustomerOverallGain / CustomerOverallGain	FLOAT32	rw	Y	0.00...2.00	1.00
0x2420	0	N	faisaftyp	Moog DCV	ValveDigitalOutput, DigitalOutput1Type / DigitalOutput1Type	INT8	ro	-	0...4	0
0x2421	0	N	faisafupp	Moog DCV	ValveFailSafeWindowMonitoring, UpperLimit / UpperLimit	INT16	ro	-	<LowerLimit>(0x2422)...32767	16384
0x2422	0	N	faisaflow	Moog DCV	ValveFailSafeWindowMonitoring, LowerLimit / LowerLimit	INT16	ro	-	-32768...<UpperLimit>(0x2421)	-16384
0x2432	0	N	curobsena	Moog DCV	CurrentControl, CurrentObservation / CurrentObservation	UINT8	ro	-	0...1	-
0x2434	0	N	curobstim	Moog DCV	CurrentControl, CurrentObservationTime / CurrentObservationTime	INT16	ro	-	INT16	5
0x2435	0	N	curobscurlim	Moog DCV	CurrentControl, CurrentObservationCurrentLimit / CurrentObservationCurrentLimit	INT16	ro	-	INT16	8192

Table 122: Object dictionary (part 8 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x2437	0	Y	curobssts	Moog DCV	CurrentControl, CurrentObservationStatus / CurrentObservationStatus	UINT8	ro	-	UINT8	3
0x2438	0	N	curobsacttim	Moog DCV	CurrentControl, CurrentObservationActiveTime / CurrentObservationActiveTime	UINT16	ro	-	UINT16	-
0x2439	0	N	curobsctldvn	Moog DCV	CurrentControl, CurrentObservationControlDeviation / CurrentObservationControlDeviation	INT16	ro	-	INT16	327
0x2530	1	Y	dipswival[0]	Moog DCV	Hardware_I2C, Hardware_I2C / DIPSwitchValue_0	UINT8	ro	-	UINT8	0
0x2530	2	Y	dipswival[1]	Moog DCV	Hardware_I2C, Hardware_I2C / DIPSwitchValue_1	UINT8	ro	-	UINT8	
0x2531	0	N	dipswimod	Moog DCV	Hardware_I2C, DIPSwitchMode / DIPSwitchMode	UINT8	ro	-	UINT8	0
0x2803	0	Y	cpusup	Moog DCV	Hardware_DiagnosticData, CpuSupplyVoltage / CpuSupplyVoltage	UINT16	ro	-	UINT16	-
0x2804	0	Y	pwrsup	Moog DCV	Hardware_DiagnosticData, PowerSupplyVoltage / PowerSupplyVoltage	UINT16	ro	-	UINT16	-
0x2805	0	Y	pcbtmp	Moog DCV	Hardware_DiagnosticData, PcbTemperature / PcbTemperature	INT16	ro	-	INT16	-
0x2809	0	N	pcbtmpmaxcus	Moog DCV	Hardware_DiagnosticData, PcbMaxTemperatureCustomer / PcbMaxTemperatureCustomer	INT16	rw	Y	INT16	-
0x280D	1	N	oprtim[0]	Moog DCV	Hardware_DiagnosticData, Hardware_DiagnosticData / PowerOnTime	UINT32	ro	-	UINT32	-
0x280D	2	N	oprtim[1]	Moog DCV	Hardware_DiagnosticData, Hardware_DiagnosticData / OperatingTime	UINT32	ro	-	UINT32	-
0x2822	1...5	N	errval[0...4]	Moog DCV	ErrorHandler, InternalErrorCode / InternalErrorCode	UINT32	ro	-	UINT32	-
0x2823	1...5	N	errtim[0...4]	Moog DCV	ErrorHandler, InternalErrorTime / InternalErrorTime	UINT32	ro	-	UINT32	-
0x2830	1...120	N	faurea[0...119]	Moog DCV	FaultReaction, FaultReactionType / FaultReactionType	INT8	rw	Y	INT8	0
0x2831	1	Y	fausts[0]	Moog DCV	FaultReaction, FaultStatus / FaultStatus	UINT32	ro	-	UINT32	-
0x2831	2	Y	fausts[1]	Moog DCV	FaultReaction, FaultStatus / FaultStatus	UINT32	ro	-	UINT32	-
0x2831	3	Y	fausts[2]	Moog DCV	FaultReaction, FaultStatus / FaultStatus	UINT32	ro	-	UINT32	-
0x2831	4	Y	fausts[3]	Moog DCV	FaultReaction, FaultStatus / FaultStatus	UINT32	ro	-	UINT32	-

Table 122: Object dictionary (part 9 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x2832	0	N	faudsc	Moog DCV	FaultReaction, FaultReactionDescription FaultReactionDescription	CHAR[64]	ro	-	-	-
0x2833	0	N	fauhis	Moog DCV	FaultReaction, FaultHistoryNumber FaultHistoryNumber	UINT8	rw	N	0…7	-
0x2834	1	Y	faustsret[0]	Moog DCV	FaultReaction, FaultRetainStatus FaultRetainStatus	UINT32	rw	N	UINT32	-
0x2834	2	Y	faustsret[1]	Moog DCV	FaultReaction, FaultRetainStatus FaultRetainStatus	UINT32	rw	N	UINT32	-
0x2834	3	Y	faustsret[2]	Moog DCV	FaultReaction, FaultRetainStatus FaultRetainStatus	UINT32	rw	N	UINT32	-
0x2834	4	Y	faustsret[3]	Moog DCV	FaultReaction, FaultRetainStatus FaultRetainStatus	UINT32	rw	N	UINT32	-
0x2900	0	N	errflg	Moog DCV	FaultReaction, CustomerDefinedErrorFlag CustomerDefinedErrorFlag	INT8	rw	N	0…127	-
0x2901	0	N	ev0exp	Moog DCV	Eventhandler, EventExpression_1 EventExpression_1	CHAR[192]	rw	Y	-	-
0x2902	0	N	ev1exp	Moog DCV	Eventhandler, EventExpression_2 EventExpression_2	CHAR[192]	rw	Y	-	-
0x2903	0	N	ev2exp	Moog DCV	Eventhandler, EventExpression_3 EventExpression_3	CHAR[192]	rw	Y	-	-
0x2904	0	N	ev3exp	Moog DCV	Eventhandler, EventExpression_4 EventExpression_4	CHAR[192]	rw	Y	-	-
0x2905	0	N	ev4exp	Moog DCV	Eventhandler, EventExpression_5 EventExpression_5	CHAR[192]	rw	Y	-	-
0x2906	0	N	ev5exp	Moog DCV	Eventhandler, EventExpression_6 EventExpression_6	CHAR[192]	rw	Y	-	-
0x2907	0	N	ev6exp	Moog DCV	Eventhandler, EventExpression_7 EventExpression_7	CHAR[192]	rw	Y	-	-
0x2908	0	N	ev7exp	Moog DCV	Eventhandler, EventExpression_8 EventExpression_8	CHAR[192]	rw	Y	-	-
0x2909	1…8	Y	evtena[0…7]	Moog DCV	Eventhandler, EventEnable EventEnable_1	UINT8	rw	Y	UINT8	-
0x290B	1…8	Y	vars08[0…7]	Moog DCV	Eventhandler, Integer08 Integer08_1	INT8	rw	N	INT8	-
0x290C	1…8	Y	vars16[0…7]	Moog DCV	Eventhandler, Integer16 Integer16_1	INT16	rw	N	INT16	-
0x290D	1…8	Y	vars32[0…7]	Moog DCV	Eventhandler, Integer32 Integer32_1	INT32	rw	N	INT32	-

Table 122: Object dictionary (part 10 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name / Parameter name	Data type	Access	Persistence	Value range	Default
0x290E	1…8	Y	varu08[0…7]	Moog DCV	Eventhandler, Unsigned08 / Unsigned08_1	UINT8	rw	N	UINT8	-
0x290F	1…8	Y	varu16[0…7]	Moog DCV	Eventhandler, Unsigned16 / Unsigned16_1	UINT16	rw	N	UINT16	-
0x2910	1…8	Y	varu32[0…7]	Moog DCV	Eventhandler, Unsigned32 / Unsigned32_1	UINT32	rw	N	UINT32	-
0x3000	0	N	pwd	Moog DCV	Can, Password / Password	UINT32	rw	N	UINT32	-
0x3002	0	N	modide	Moog DCV	CAN, ModuleIdentifier / ModuleIdentifier	UINT8	rw	Y	1…127	0x7F
0x3003	0	N	canbdr	Moog DCV	CAN, Bitrate / Bitrate	UINT32	rw	Y	10000…1000000	500000
0x3010	1	N	pdttrmman[0]	Moog DCV	CAN, TPdo_ManufacturerTransmissionType / TPdo1_ManufacturerTransmissionType	UINT8	rw	Y	UINT8	-
0x3010	2	N	pdttrmman[1]	Moog DCV	CAN, TPdo_ManufacturerTransmissionType / TPdo2_ManufacturerTransmissionType	UINT8	rw	Y	UINT8	-
0x3010	3	N	pdttrmman[2]	Moog DCV	CAN, TPdo_ManufacturerTransmissionType / TPdo3_ManufacturerTransmissionType	UINT8	rw	Y	UINT8	-
0x3010	4	N	pdttrmman[3]	Moog DCV	CAN, TPdo_ManufacturerTransmissionType / TPdo4_ManufacturerTransmissionType	UINT8	rw	Y	UINT8	-
0x3011	0	N	pdttrg	Moog DCV	CAN, TPdoTrigger / TPdoTrigger	UINT8	rw	N	0…4	-
0x3012	1	Y	pdrctr[0]	Moog DCV	CAN, RPdo_Counter / RPdo1_Counter	UINT32	rw	N	UINT32	-
0x3012	2	Y	pdrctr[1]	Moog DCV	CAN, RPdo_Counter / RPdo2_Counter	UINT32	rw	N	UINT32	-
0x3012	3	Y	pdrctr[2]	Moog DCV	CAN, RPdo_Counter / RPdo3_Counter	UINT32	rw	N	UINT32	-
0x3012	4	Y	pdrctr[3]	Moog DCV	CAN, RPdo_Counter / RPdo4_Counter	UINT32	rw	N	UINT32	-
0x3013	0	N	snctim	Moog DCV	CAN, SyncTimer / SyncTimer	UINT16	rw	Y	UINT16	-
0x3014	0	N	pdtsnc	Moog DCV	CAN, RealtimeSyncronisationTPdo / RealtimeSyncronisationTPdo	UINT8	rw	Y	0…4	-
0x3015	0	N	trgpdt	Moog DCV	CAN, CopyTrgpdtToPdtsnc / CopyTrgpdtToPdtsnc	UINT8	rw	Y	0…4	-
0x3100	0	Y	fcntyp	Moog DCV	FunctionGenerator, FunctionGenType / FunctionGenType	INT8	rw	N	0…5	-

Table 122: Object dictionary (part 11 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x3101	0	Y	fcndem	Moog DCV	FunctionGenerator, FunctionGenOutput FunctionGenOutput	INT16	ro	-	INT16	-
0x3102	0	Y	fcnsqr	Moog DCV	FunctionGenerator, FunctionGenSquareOutput FunctionGenSquareOutput	INT16	ro	-	INT16	-
0x3103	0	Y	fcntim	Moog DCV	FunctionGenerator, FunctionGenFrequency FunctionGenFrequency	UINT16	rw	N	1...<LvdtFrequency> (0x3030)	10
0x3104	0	Y	fcnmag	Moog DCV	FunctionGenerator, FunctionGenMagnitude FunctionGenMagnitude	INT16	rw	N	0...32767	-
0x3105	0	Y	fcnofs	Moog DCV	FunctionGenerator, FunctionGenOffset FunctionGenOffset	INT16	rw	N	INT16	-
0x3107	0	Y	fcnsgn	Moog DCV	FunctionGenerator, FunctionGenSign FunctionGenSign	INT8	rw	N	-1...1	1
0x3108	0	Y	fcnprf	Moog DCV	FunctionGenerator, FunctionGenFrequencyPrefix FunctionGenFrequencyPrefix	INT8	rw	N	-4...0	-
0x3180	0	N	dlgctl	Moog DCV	DataLogger, Control Control	UINT8	rw	N	0...1	-
0x3181	0	Y	dlgsts	Moog DCV	DataLogger, Status Status	UINT8	ro	-	0...3	-
0x3182	0	N	dlgdiv	Moog DCV	DataLogger, Divider Divider	UINT16	rw	Y	1...65535	1
0x3183	0	N	dlgsmp	Moog DCV	DataLogger, NumberOfSamples NumberOfSamples	INT32	ro	-	-2147483648...2048	-
0x3184	1	N	dlgena[0]	Moog DCV	DataLogger, EnableChannel EnableChannel1	UINT8	rw	Y	UINT8	-
0x3184	2	N	dlgena[1]	Moog DCV	DataLogger, EnableChannel EnableChannel2	UINT8	rw	Y	UINT8	-
0x3184	3	N	dlgena[2]	Moog DCV	DataLogger, EnableChannel EnableChannel3	UINT8	rw	Y	UINT8	-
0x3184	4	N	dlgena[3]	Moog DCV	DataLogger, EnableChannel EnableChannel4	UINT8	rw	Y	UINT8	-
0x3185	1	N	dlgpar[0]	Moog DCV	DataLogger, ChannelParameter ChannelParameter1	UINT32	rw	Y	UINT32	0x63100110
0x3185	2	N	dlgpar[1]	Moog DCV	DataLogger, ChannelParameter ChannelParameter2	UINT32	rw	Y	UINT32	0x63010110
0x3185	3	N	dlgpar[2]	Moog DCV	DataLogger, ChannelParameter ChannelParameter3	UINT32	rw	Y	UINT32	0x63900110
0x3185	4	N	dlgpar[3]	Moog DCV	DataLogger, ChannelParameter ChannelParameter4	UINT32	rw	Y	UINT32	0x63810110

Table 122: Object dictionary (part 12 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name / Parameter name	Data type	Access	Persistence	Value range	Default
0x3186	0	N	dlgmem	Moog DCV	DataLogger, Memory / Memory	DOMAIN	ro	-	-	-
0x3187	0	N	dlgofs	Moog DCV	DataLogger, SampleStartOffset / SampleStartOffset	UINT32	ro	-	UINT32	-
0x3188	0	N	trgtyp	Moog DCV	DataLogger, TriggerType / TriggerType	UINT8	rw	Y	0...2	1
0x3189	0	N	trgpar	Moog DCV	DataLogger, TriggerParameter / TriggerParameter	UINT32	rw	Y	UINT32	0x63100110
0x318A	0	N	trgcpl	Moog DCV	DataLogger, TriggerCoupling / TriggerCoupling	UINT8	rw	Y	0...2	1
0x318B	0	N	trgslp	Moog DCV	DataLogger, TriggerSlope / TriggerSlope	UINT8	rw	Y	1...3	1
0x318C	0	N	trglvl	Moog DCV	DataLogger, TriggerLevelOrBitmask / TriggerLevelOrBitmask	INT32	rw	Y	INT32	-
0x318D	0	N	trgpos	Moog DCV	DataLogger, TriggerPosition / TriggerPosition	INT32	rw	Y	INT32	-
0x318E	0	N	trgtim	Moog DCV	DataLogger, TriggerTimeStamp / TriggerTimeStamp	UINT32	ro	-	UINT32	-
0x3200	0	N	an0typ	Moog DCV	AnalogInput0, AnaInpType0 / AnaInpType0	INT8	rw	Y	0...12	1
0x3204	0	Y	an0val	Moog DCV	AnalogInput0, AnaInpActualValue0 / AnaInpActualValue0	INT16	ro	-	INT16	-
0x3208	0	N	an1typ	Moog DCV	AnalogInput1, AnaInpType1 / AnaInpType1	INT8	rw	Y	0...12	2
0x320C	0	Y	an1val	Moog DCV	AnalogInput1, AnaInpActualValue1 / AnaInpActualValue1	INT16	ro	-	INT16	-
0x3210	0	N	an2typ	Moog DCV	AnalogInput2, AnaInpType2 / AnaInpType2	INT8	rw	Y	0...12	2
0x3214	0	Y	an2val	Moog DCV	AnalogInput2, AnaInpActualValue2 / AnaInpActualValue2	INT16	ro	-	INT16	-
0x3217	0	N	an2mon	Moog DCV	AnalogInput2, AnaInpMonitorCurrent2 / AnaInpMonitorCurrent2	UINT8	rw	Y	0...1	-
0x3218	0	N	an3typ	Moog DCV	AnalogInput3, AnaInpType3 / AnaInpType3	INT8	rw	Y	0...12	2
0x321C	0	Y	an3val	Moog DCV	AnalogInput3, AnaInpActualValue3 / AnaInpActualValue3	INT16	ro	-	INT16	-
0x3220	0	N	an4typ	Moog DCV	AnalogInput4, AnaInpType4 / AnaInpType4	INT8	rw	Y	0...12	2

Table 122: Object dictionary (part 13 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x3224	0	Y	an4val	Moog DCV	AnalogInput4, AnalnpActualValue4 AnalnpActualValue4	INT16	ro	-	INT16	-
0x3227	0	N	an4mon	Moog DCV	AnalogInput4, AnalnMonitorCurrent4 AnalnMonitorCurrent4	UINT8	rw	Y	0...1	-
0x3228	0	N	an3mon	Moog DCV	AnalogInput3, AnalnMonitorCurrent3 AnalnMonitorCurrent3	UINT8	rw	Y	0...1	-
0x3235	0	Y	extlvdval	Moog DCV	ExternalLVDT, ExternalVDTActualValue ExternalVDTActualValue	INT16	ro	-	INT16	-
0x3237	1	N	extlvdref[0]	Moog DCV	ExternalLVDT, ExternalvdtScaCustomer ExternalvdtScaNumerator	INT16	rw	Y	INT16	16384
0x3237	2	N	extlvdref[1]	Moog DCV	ExternalLVDT, ExternalvdtScaCustomer ExternalvdtScaDenominator	INT16	rw	Y	INT16	16384
0x3237	3	N	extlvdref[2]	Moog DCV	ExternalLVDT, ExternalvdtScaCustomer ExternalvdtScaOffset	INT16	rw	Y	INT16	0
0x3240	0	N	da0par	Moog DCV	AnalogOutput0, AnaOutMappingParameter0 AnaOutMappingParameter0	UINT32	rw	Y	UINT32	0x63010110
0x3243	0	N	da0typ	Moog DCV	AnalogOutput0, AnaOutType0 AnaOutType0	UINT8	rw	Y	0...1	0
0x3244	1	N	da0ref[0]	Moog DCV	AnalogOutput0, AnaOutScaCustomer0 AnaOutScaNumerator0	INT16	rw	Y	INT16	16384
0x3244	2	N	da0ref[1]	Moog DCV	AnalogOutput0, AnaOutScaCustomer0 AnaOutScaDenominator0	INT16	rw	Y	INT16	16384
0x3244	3	N	da0ref[2]	Moog DCV	AnalogOutput0, AnaOutScaCustomer0 AnaOutScaOffset0	INT16	rw	Y	INT16	0
0x3245	0	Y	da0val	Moog DCV	AnalogOutput0, AnaOutValue0 AnaOutValue0	INT16	ro	-	INT16	-
0x3250	0	N	anamonlow	Moog DCV	AnalogInput, LowerCurrentBorder LowerCurrentBorder	FLOAT32	rw	Y	2.20...20.00	3.00
0x3251	0	N	anamontim	Moog DCV	AnalogInput, anInputMonitoringTime_in_ms anInputMonitoringTime_in_ms	UINT16	rw	Y	0...60000	10
0x3252	0	N	ssierrcnt	Moog DCV	ErrorHandler, SSIErrorCount SSIErrorCount	UINT8	rw	Y	0...254	-
0x3260	0	N	da1par	Moog DCV	AnalogOutput1, AnaOutMappingParameter1 AnaOutMappingParameter1	UINT32	rw	Y	UINT32	0x63810110
0x3263	0	N	da1typ	Moog DCV	AnalogOutput1, AnaOutType0 AnaOutType0	UINT8	rw	Y	0...1	0
0x3265	1	N	da1ref[0]	Moog DCV	AnalogOutput1, AnaOutScaCustomer1 AnaOutScaNumerator1	INT16	rw	Y	INT16	16384

Table 122: Object dictionary (part 14 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x3265	2	N	da1ref[1]	Moog DCV	AnalogOutput1, AnaOutScaCustomer1 / AnaOutScaDenominator1	INT16	rw	Y	INT16	16384
0x3265	3	N	da1ref[2]	Moog DCV	AnalogOutput1, AnaOutScaCustomer1 / AnaOutScaOffset1	INT16	rw	Y	INT16	0
0x3266	0	Y	da1val	Moog DCV	AnalogOutput1, AnaOutValue1 / AnaOutValue1	INT16	ro	-	INT16	-
0x3271	0	N	drvtrdstc	Moog DCV	Drive, DriveTransducerStructure / DriveTransducerStructure	DOMAIN	rw	Y	-	-
0x3300	0	Y	spldemplt	Moog DCV	ValvePositionControl, DemandValvePilot / DemandValvePilot	INT16	ro	-	INT16	-
0x3301	0	Y	splvalplt	Moog DCV	ValvePositionControl, ActualValvePilot / ActualValvePilot	INT16	ro	-	INT16	-
0x3302	0	Y	pltctldvn	Moog DCV	ValvePositionControl, PilotControlDeviation / PilotControlDeviation	INT16	ro	-	INT16	-
0x3307	0	N	spgposmin	Moog DCV	Hardware_DiagnosticData, SpringPositionMinimum / SpringPositionMinimum	INT16	ro	-	INT16	-
0x3308	0	N	spgposmax	Moog DCV	Hardware_DiagnosticData, SpringPositionMaximum / SpringPositionMaximum	INT16	ro	-	INT16	-
0x3310	0	N	prspar	Moog DCV	ValvePressureControl, PrsSetpointParameter / PrsSetpointParameter	UINT32	rw	Y	UINT32	0x63800110
0x3320	0	N	splpar	Moog DCV	ValvePositionControl, SplSetpointParameter / SplSetpointParameter	UINT32	rw	Y	UINT32	0x63000110
0x3330	0	N	pospar	Moog DCV	DrivePositionControl, DemandParameter / DemandParameter	UINT32	rw	Y	UINT32	0x66000120
0x3404	0	Y	prstrd	Moog DCV	PressureTransducer, ActualValue / ActualValue	INT16	ro	-	INT16	-
0x3506	0	N	lvdcusofs	Moog DCV	Lvdt, CustomerScalingOffset / CustomerScalingOffset	INT16	rw	Y	-819...819	-
0x4000	0	N	xmldat	Moog DCV	Device, XmlDescription / XmlDescription	DOMAIN	ro	-	-	-
0x4015	0	N	faurea[6]	Moog DCV	FaultReaction, FaultReactionType / FaultReactionType	INT8	rw	Y	INT8	2
0x4019	1...5	N	faurea[26...30]	Moog DCV	FaultReaction, FaultReactionType / FaultReactionType	INT8	rw	Y	INT8	0
0x401A	0	N	faurea[20]	Moog DCV	FaultReaction, FaultReactionType / FaultReactionType	INT8	rw	Y	INT8	0
0x401B	0	N	faurea[90]	Moog DCV	FaultReaction, FaultReactionType / FaultReactionType	INT8	rw	Y	INT8	0

Table 122: Object dictionary (part 15 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x4031	1…6	N	drvrdfltcof[0…5]	CiA 408	Drive_ActualValueConditioning, FilterCoefficient FilterCoefficient	FLOAT32	rw	N	FLOAT32	-
0x4032	0	N	drvtrdpar	CiA 408	Drive_ActualValueConditioning, TransducerPort TransducerPort	UINT32	rw	N	UINT32	-
0x403F	0	N	ctllocdef	Moog DCV	Device, LocalControlWordDefault LocalControlWordDefault	UINT16	rw	Y	UINT16	0x107
0x4040	0	Y	ctlloc	Moog DCV	Device, LocalControlWord LocalControlWord	UINT16	rw	N	UINT16	<LocalControlWordDefault> (0x403F)
0x4042	0	N	devmoddef	Moog DCV	Device, DeviceModeDefault DeviceModeDefault	INT8	rw	Y	1…2	1
0x4043	0	N	ctlmoddef	Moog DCV	Device, ControlModeDefault ControlModeDefault	INT8	rw	Y	-1…14	2
0x4230	1	N	posminref	Moog DCV	DrivePositionControl, MinimumReference MinimumReference	INT32	rw	Y	INT32	-16384
0x4231	1	N	posmaxref	Moog DCV	DrivePositionControl, MaximumReference MaximumReference	INT32	rw	Y	INT32	16384
0x4347	0	N	splchrtbl	Moog DCV	ValvePositionControl_DemandValueGenerator, CharacteristicCompensation_LookUpTable CharacteristicCompensation_LookUpTable	DOMAIN	rw	Y	-	-
0x4348	1…128	N	splchrtbl[0…127]	Moog DCV	ValvePositionControl_DemandValueGenerator, CharacteristicCompensation_LookUpTable CharacteristicCompensation_LookUpTable	INT16	rw	Y	INT16	-
0x4349	0	N	splchrtbl[128]	Moog DCV	ValvePositionControl_DemandValueGenerator, CharacteristicCompensation_LookUpTable CharacteristicCompensation_LookUpTable	INT16	rw	Y	INT16	-
0x434A	1…128	N	splchrtbl[129…256]	Moog DCV	ValvePositionControl_DemandValueGenerator, CharacteristicCompensation_LookUpTable CharacteristicCompensation_LookUpTable	INT16	rw	Y	INT16	-
0x5200	1	Y	flwbrncor	Moog DCV	ValveFlowControl, BernoulliCorrection BernoulliCorrection	FLOAT32	ro	-	FLOAT32	-
0x5202	0	N	flwpar	Moog DCV	ValveFlowControl, SetpointParameter SetpointParameter	UINT32	ro	-	UINT32	0x53000110
0x5203	0	N	prspsnom	Moog DCV	ValveFlowControl, NominalSupplyPressure NominalSupplyPressure	INT16	rw	Y	1…32767	400
0x5204	1	N	prsflwtim[0]	Moog DCV	ValveFlowControl, ValveFlowControl AntiOscillationWaitTime_0	FLOAT32	rw	Y	0.00…+inf	30.00

Table 122: Object dictionary (part 16 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Object name, Parameter name Block name	Data type	Access	Persistence	Value range	Default
0x5204	2	N	prsflwtim[1]	Moog DCV	ValveFlowControl, ValveFlowControl AntiOscillationWaitTime_1	FLOAT32	rw	Y	0.00...+inf	3.00
0x5205	0	N	flwmod	Moog DCV	ValveFlowControl, FlowControlMode FlowControlMode	UINT8	rw	Y	UINT8	-
0x520D	0	N	flwprspeatrs	Moog DCV	ValveFlowControl, PressurePeakThreshold PressurePeakThreshold	INT16	rw	Y	0...32767	20000
0x520E	0	Y	flwvlvdir	Moog DCV	ValveFlowControl, ValveDirection ValveDirection	INT8	rw	Y	-1...1	1
0x520F	0	N	flwctlout	Moog DCV	ValveFlowControl, ControllerOutput ControllerOutput	INT16	ro	-	INT16	-
0x5210	0	Y	flwprsfdbflt	Moog DCV	ValveFlowControl, PressureFeedbackFilterPT1TimeConstant PressureFeedbackFilterPT1TimeConstant	FLOAT32	rw	Y	0.00...+inf	-
0x5212	0	Y	flwctlsts	Moog DCV	ValveFlowControl, FlowActualStatus FlowActualStatus	UINT8	ro	-	UINT8	-
0x5213	0	Y	flwdem	Moog DCV	ValveFlowControl, FlowDemand FlowDemand	INT16	ro	-	INT16	-
0x5215	0	N	vlvqan	Moog DCV	ValveFlowControl, RatedValveFlowASide RatedValveFlowASide	FLOAT32	rw	Y	0.01...+inf	1.00
0x5216	0	N	vlvqbn	Moog DCV	ValveFlowControl, RatedValveFlowBSide RatedValveFlowBSide	FLOAT32	rw	Y	0.01...+inf	1.00
0x5217	0	N	cmpprsitfpt	Moog DCV	ValveFlowControl, PressureControllerTransducerInterfacePT PressureControllerTransducerInterfacePT	INT8	rw	Y	0...8	-
0x5219	0	N	flwalpha	Moog DCV	ValveFlowControl, Alpha Alpha	FLOAT32	ro	-	FLOAT32	-
0x521A	0	N	flwbeta	Moog DCV	ValveFlowControl, Beta Beta	FLOAT32	ro	-	FLOAT32	-
0x521B	0	N	flwsetsca	Moog DCV	ValveFlowControl, SetpointScaling SetpointScaling	FLOAT32	rw	N	-1.00...1.00	1.00
0x521C	0	N	prsflwctlout	Moog DCV	ValveFlowControl, P_Flow_ControllerOutput P_Flow_ControllerOutput	INT16	ro	-	INT16	-
0x521D	0	N	prssensmax	Moog DCV	ValveFlowControl, PressureSensorMaximum PressureSensorMaximum	INT16	ro	-	INT16	-
0x521E	0	N	vlvdpn	Moog DCV	ValveFlowControl, RatedValvePressureDrop RatedValvePressureDrop	INT16	rw	Y	1...32767	400
0x521F	0	N	flwqamax	Moog DCV	ValveFlowControl, MaximalFlowQAMax MaximalFlowQAMax	FLOAT32	ro	-	FLOAT32	-
0x5220	0	N	flwfrcfhi	Moog DCV	ValveFlowControl, ForceHightPassCutOffFreq ForceHightPassCutOffFreq	FLOAT32	rw	Y	FLOAT32	20.00

Table 122: Object dictionary (part 17 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x5221	0	Y	flwfrcgn	Moog DCV	ValveFlowControl, ForceGain ForceGain	FLOAT32	rw	Y	FLOAT32	1.00
0x5222	0	N	flwfdbpar	Moog DCV	ValveFlowControl, flowFeedbackSource flowFeedbackSource	UINT32	rw	Y	UINT32	0x52230010
0x5223	0	N	flwfrc	Moog DCV	ValveFlowControl, ForceFeedback ForceFeedback	INT16	ro	-	INT16	-
0x5224	0	N	flwfdbgn	Moog DCV	ValveFlowControl, FlowFeedbackGain FlowFeedbackGain	FLOAT32	rw	Y	FLOAT32	-
0x5300	1	Y	flwset	Moog DCV	ValveFlowControl, FlowSetpoint FlowSetpoint	INT16	rw	N	INT16	-
0x5300	2	N	flwuni	Moog DCV	ValveFlowControl, FlowSetpoint Unit	UINT8	ro	-	UINT8	0
0x5300	3	N	flwprf	Moog DCV	ValveFlowControl, FlowSetpoint Prefix	INT8	ro	-	INT8	0
0x5314	1	Y	flwsethld	Moog DCV	ValveFlowControl, FlowHoldSetPoint FlowHoldSetPoint	INT16	rw	Y	INT16	-
0x5314	2	N	flwuni	Moog DCV	ValveFlowControl, FlowHoldSetPoint Unit	UINT8	ro	-	UINT8	0
0x5314	3	N	flwprf	Moog DCV	ValveFlowControl, FlowHoldSetPoint Prefix	INT8	ro	-	INT8	0
0x5400	1	N	locpdrcob[0]	Moog DCV	LocalCAN, RPdo LocalCANRPdo1_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0200
0x5400	2	N	locpdrtrn[0]	Moog DCV	LocalCAN, RPdo LocalCANRPdo1_TransmissionType	UINT8	rw	Y	UINT8	255
0x5400	5	N	locpdrtim[0]	Moog DCV	LocalCAN, RPdo LocalCANRPdo1_EventTimer	UINT16	rw	Y	UINT16	-
0x5401	1	N	locpdrcob[1]	Moog DCV	LocalCAN, RPdo LocalCANRPdo2_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0300
0x5401	2	N	locpdrtrn[1]	Moog DCV	LocalCAN, RPdo LocalCANRPdo2_TransmissionType	UINT8	rw	Y	UINT8	255
0x5401	5	N	locpdrtim[1]	Moog DCV	LocalCAN, RPdo LocalCANRPdo2_EventTimer	UINT16	rw	Y	UINT16	-
0x5402	1	N	locpdrcob[2]	Moog DCV	LocalCAN, RPdo LocalCANRPdo3_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0400
0x5402	2	N	locpdrtrn[2]	Moog DCV	LocalCAN, RPdo LocalCANRPdo3_TransmissionType	UINT8	rw	Y	UINT8	255

Table 122: Object dictionary (part 18 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name / Parameter name	Data type	Access	Persistence	Value range	Default
0x5402	5	N	locpdrtim[2]	Moog DCV	LocalCAN, RPdo / LocalCANRPdo3_EventTimer	UINT16	rw	Y	UINT16	-
0x5403	1	N	locpdrcob[3]	Moog DCV	LocalCAN, RPdo / LocalCANRPdo4_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0500
0x5403	2	N	locpdrtrn[3]	Moog DCV	LocalCAN, RPdo / LocalCANRPdo4_TransmissionType	UINT8	rw	Y	UINT8	255
0x5403	5	N	locpdrtim[3]	Moog DCV	LocalCAN, RPdo / LocalCANRPdo4_EventTimer	UINT16	rw	Y	UINT16	-
0x5501	1	Y	pospgn[0]	Moog DCV	DrivePositionControl, DrivePositionControl / Kp	FLOAT32	rw	Y	FLOAT32	-
0x5501	2	Y	pospgn[1]	Moog DCV	DrivePositionControl, DrivePositionControl / KpT1	FLOAT32	rw	Y	FLOAT32	-
0x5502	1	Y	posdgn	Moog DCV	DrivePositionControl, Td / Td	FLOAT32	rw	Y	FLOAT32	-
0x5503	1	Y	posdt1	Moog DCV	DrivePositionControl, T1 / T1	FLOAT32	rw	Y	0.00...+inf	-
0x5504	1	Y	posign[0]	Moog DCV	DrivePositionControl, DrivePositionControl / Outer_I_Gain	FLOAT32	rw	Y	FLOAT32	-
0x5504	2	Y	posign[1]	Moog DCV	DrivePositionControl, DrivePositionControl / Inner_I_Gain	FLOAT32	rw	Y	FLOAT32	-
0x5505	1	Y	posicrout	Moog DCV	DrivePositionControl, Integrator_OuterControlRange / Integrator_OuterControlRange	FLOAT32	rw	Y	0.00...2147483648.00	-
0x5506	1	Y	posffv	Moog DCV	DrivePositionControl, VelocityFeedforwardProportionalGain / VelocityFeedforwardProportionalGain	FLOAT32	rw	Y	FLOAT32	-
0x5507	1	Y	posffa	Moog DCV	DrivePositionControl, AccelerationFeedforwardProportionalGain / AccelerationFeedforwardProportionalGain	FLOAT32	rw	Y	FLOAT32	-
0x5508	1	Y	pospt1	Moog DCV	DrivePositionControl, KpT1TimeConstant / KpT1TimeConstant	FLOAT32	rw	Y	0.00...+inf	-
0x5509	1	N	posminitf	Moog DCV	DrivePositionControl, MinimumInterface / MinimumInterface	INT32	rw	Y	INT32	-16384
0x550A	1	Y	posfba	Moog DCV	DrivePositionControl, StateFeedbackAccelerationProportionalGain / StateFeedbackAccelerationProportionalGain	FLOAT32	rw	Y	FLOAT32	-
0x550B	1	Y	posfbv	Moog DCV	DrivePositionControl, StateFeedbackVelocityProportionalGain / StateFeedbackVelocityProportionalGain	FLOAT32	rw	Y	FLOAT32	-
0x550C	1	Y	posfbp	Moog DCV	DrivePositionControl, FilterLowPassFilter / FilterLowPassFilter	FLOAT32	rw	Y	0.00...+inf	-

Table 122: Object dictionary (part 19 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name / Parameter name	Data type	Access	Persistence	Value range	Default
0x550D	1	Y	posposgn	Moog DCV	DrivePositionControl, PositiveOverallGain / PositiveOverallGain	FLOAT32	rw	Y	FLOAT32	1.00
0x550E	1	Y	posneggn	Moog DCV	DrivePositionControl, NegativeOverallGain / NegativeOverallGain	FLOAT32	rw	Y	FLOAT32	1.00
0x550F	1	Y	posctlout	Moog DCV	DrivePositionControl, PositionControllerOutput / PositionControllerOutput	FLOAT32	ro	-	FLOAT32	-
0x5510	1	N	posmaxitf	Moog DCV	DrivePositionControl, MaximumInterface / MaximumInterface	INT32	rw	Y	INT32	16384
0x5511	1	Y	posicrin	Moog DCV	DrivePositionControl, Integrator_InnerControlRange / Integrator_InnerControlRange	FLOAT32	rw	Y	0.00...2147483648.00	-
0x5513	1	Y	posignlim	Moog DCV	DrivePositionControl, IntegratorLimit / IntegratorLimit	FLOAT32	rw	Y	0.00...+inf	16384.00
0x5514	1	N	velpgn	Moog DCV	DriveVelocityControl, Kp / Kp	FLOAT32	rw	Y	FLOAT32	-
0x5516	1	Y	velicrout	Moog DCV	DriveVelocityControl, Integrator_OuterControlRange / OuterControlRange	FLOAT32	rw	Y	0.00...2147483648.00	-
0x5517	1	Y	velicrin	Moog DCV	DriveVelocityControl, Integrator_InnerControlRange / InnerControlRange	FLOAT32	rw	Y	0.00...2147483648.00	-
0x5518	1	Y	velignlim	Moog DCV	DriveVelocityControl, Integrator_IntegratorLimit / IntegratorLimit	FLOAT32	rw	Y	0.00...+inf	16384.00
0x5519	1	Y	velctlout	Moog DCV	DriveVelocityControl, VelocityControllerOutput / VelocityControllerOutput	FLOAT32	ro	-	FLOAT32	-
0x5520	0	N	velpar	Moog DCV	DriveVelocityControl, DemandParameter / DemandParameter	UINT32	rw	Y	UINT32	0x65000120
0x5522	1	Y	velign[0]	Moog DCV	DriveVelocityControl, IntegratorGain_Low / IntegratorGain_Low	FLOAT32	rw	Y	FLOAT32	-
0x5522	2	Y	velign[1]	Moog DCV	DriveVelocityControl, IntegratorGain_High / IntegratorGain_High	FLOAT32	rw	Y	FLOAT32	-
0x5523	1	N	velminref	Moog DCV	DriveVelocityControl, MinimumReference / MinimumReference	INT32	rw	Y	INT32	-16384
0x5524	1	N	velmaxref	Moog DCV	DriveVelocityControl, MaximumReference / MaximumReference	INT32	rw	Y	INT32	16384
0x5525	1	N	velminitf	Moog DCV	DriveVelocityControl, MinimumInterface / MinimumInterface	INT32	rw	Y	INT32	-16384
0x5526	1	N	velmaxitf	Moog DCV	DriveVelocityControl, MaximumInterface / MaximumInterface	INT32	rw	Y	INT32	16384
0x5527	1	Y	velsethld	Moog DCV	DrivePositionControl_DemandValueGenerator, HoldSetpoint / HoldSetpoint	INT32	rw	Y	INT32	-

Table 122: Object dictionary (part 20 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x5527	2	N	veluni	Moog DCV	DrivePositionControl_DemandValueGenerator, HoldSetpoint Velocity_Unit	UINT8	ro	-	UINT8	0
0x5527	3	N	velprf	Moog DCV	DrivePositionControl_DemandValueGenerator, HoldSetpoint Velocity_Prefix	INT8	ro	-	INT8	0
0x5529	1	Y	velfbapgn	Moog DCV	DriveVelocityControl, StateFeedbackAccelerationProportionalGain StateFeedbackAccelerationProportionalGain	FLOAT32	rw	Y	FLOAT32	-
0x552A	1	Y	pospgnout[0]	Moog DCV	DrivePositionControl, DrivePositionControl KpOutput	FLOAT32	ro	-	FLOAT32	-
0x552A	2	Y	pospgnout[1]	Moog DCV	DrivePositionControl, DrivePositionControl KpT1Output	FLOAT32	ro	-	FLOAT32	-
0x552B	1	Y	posdgnout	Moog DCV	DrivePositionControl, KdOutput KdOutput	FLOAT32	ro	-	FLOAT32	-
0x552C	1	Y	posignout	Moog DCV	DrivePositionControl, KiOutput KiOutput	FLOAT32	ro	-	FLOAT32	-
0x552E	1	Y	posffvout	Moog DCV	DrivePositionControl, FeedForwardVelocityOutput FeedForwardVelocityOutput	FLOAT32	ro	-	FLOAT32	-
0x552F	1	Y	posffaout	Moog DCV	DrivePositionControl, FeedForwardAccelerationOutput FeedForwardAccelerationOutput	FLOAT32	ro	-	FLOAT32	-
0x5530	1	Y	posfbvout	Moog DCV	DrivePositionControl, VelocityStateFeedbackOutput VelocityStateFeedbackOutput	FLOAT32	ro	-	FLOAT32	-
0x5531	1	Y	posfbaout	Moog DCV	DrivePositionControl, AccelerationStateFeedbackOutput AccelerationStateFeedbackOutput	FLOAT32	ro	-	FLOAT32	-
0x5533	1	Y	veldemflt	Moog DCV	DriveVelocityControl, VelocityDemandFilter VelocityDemandFilter	FLOAT32	rw	Y	0.00...+inf	-
0x5534	1	Y	velposgn	Moog DCV	DriveVelocityControl, VelocityPositiveOverallGain VelocityPositiveOverallGain	FLOAT32	rw	Y	FLOAT32	-
0x5535	1	Y	velneggn	Moog DCV	DriveVelocityControl, VelocityNegativeOverallGain VelocityNegativeOverallGain	FLOAT32	rw	Y	FLOAT32	-
0x5537	1	Y	velpgnout	Moog DCV	DriveVelocityControl, VelocityProportionalGainOut VelocityProportionalGainOut	FLOAT32	ro	-	FLOAT32	-
0x5539	1	Y	velfbapgnout	Moog DCV	DriveVelocityControl, VelocityFeedbackAccelerationGain VelocityFeedbackAccelerationGain	FLOAT32	ro	-	FLOAT32	-
0x553B	1	Y	veldemfltout	Moog DCV	DriveVelocityControl, VelocityDemandFilterOutput VelocityDemandFilterOutput	FLOAT32	ro	-	FLOAT32	-
0x553C	1	Y	velignout	Moog DCV	DriveVelocityControl, VelocityIntegratorOutput VelocityIntegratorOutput	FLOAT32	ro	-	FLOAT32	-
0x553F	1	Y	posctldiv	Moog DCV	DrivePositionControl, ControlTimeDivider ControlTimeDivider	UINT8	rw	Y	2...255	10

Table 122: Object dictionary (part 21 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Object name, Block name / Parameter name	Data type	Access	Persistence	Value range	Default
0x5540	1	Y	velctldiv	Moog DCV	DriveVelocityControl, ControlTimeDivider / ControlTimeDivider	UINT8	rw	Y	1...255	2
0x5542	1	N	velitfref	Moog DCV	DriveVelocityControl, InterfaceReference / InterfaceReference	UINT8	rw	Y	1...<MaxInterfaceNo> (0x6200)	1
0x5544	1	N	velres	Moog DCV	DriveVelocityControl, VelocityResolution / VelocityResolution	FLOAT32	rw	Y	FLOAT32	1.00
0x5545	1	N	velcalmod	Moog DCV	DriveVelocityControl, VelocityCalculationMode / VelocityCalculationMode	UINT8	rw	Y	UINT8	-
0x5546	1	N	velffvout	Moog DCV	DriveVelocityControl, VelocityFeedForwardOut / VelocityFeedForwardOut	FLOAT32	ro	-	FLOAT32	-
0x5547	1	N	velffv	Moog DCV	DriveVelocityControl, VelocityFeedForwardGain / VelocityFeedForwardGain	FLOAT32	rw	Y	FLOAT32	-
0x5548	1	N	velvelflt	Moog DCV	DriveVelocityControl, VelocitySignalFilterTimeConstant / VelocitySignalFilterTimeConstant	FLOAT32	rw	Y	0.00...+inf	-
0x5549	1	N	velvelfltout	Moog DCV	DriveVelocityControl, VelocitySignalFilterOutput / VelocitySignalFilterOutput	FLOAT32	ro	-	FLOAT32	-
0x5550	1	N	velquelen	Moog DCV	DriveVelocityControl, VelocitySignalQueueLength / VelocitySignalQueueLength	UINT8	rw	Y	1...20	1
0x5551	1	N	accval	Moog DCV	DriveVelocityControl, AccelerationSignal / AccelerationSignal	FLOAT32	ro	-	FLOAT32	-
0x5552	1	N	velposfltout	Moog DCV	DriveVelocityControl, PositionSignalFilterOutput / PositionSignalFilterOutput	FLOAT32	ro	-	FLOAT32	-
0x5553	1	N	veldemrmp	Moog DCV	DriveVelocityControl, VelocityDemandRamp / VelocityDemandRamp	INT32	rw	Y	INT32	-
0x5600	0	N	locpdrmapnum	Moog DCV	LocalCAN, LocalCANRPdo_NumberOfMappedApplicParaInPdo / LocalCANRPdo_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0...8	0
0x5600	1...8	N	locpdrmap[0...7]	Moog DCV	LocalCAN, RPdo_Mapping / LocalRPdo1_ApplicPara1	UINT32	rw	Y	UINT32	-
0x5601	0	N	locpdrmapnum	Moog DCV	LocalCAN, LocalCANRPdo_NumberOfMappedApplicParaInPdo / LocalCANRPdo_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0...8	0
0x5601	1...8	N	locpdrmap[8...15]	Moog DCV	LocalCAN, RPdo_Mapping / LocalRPdo2_ApplicPara1	UINT32	rw	Y	UINT32	-
0x5602	0	N	locpdrmapnum	Moog DCV	LocalCAN, LocalCANRPdo_NumberOfMappedApplicParaInPdo / LocalCANRPdo_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0...8	0
0x5602	1...8	N	locpdrmap[16...23]	Moog DCV	LocalCAN, RPdo_Mapping / LocalRPdo3_ApplicPara1	UINT32	rw	Y	UINT32	-
0x5603	0	N	locpdrmapnum	Moog DCV	LocalCAN, LocalCANRPdo_NumberOfMappedApplicParaInPdo / LocalCANRPdo_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0...8	0

Table 122: Object dictionary (part 22 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x5603	1…8	N	loocpdrmap[24…31]	Moog DCV	LocalCAN, RPdo_Mapping LocalRPdo4_ApplicPara1	UINT32	rw	Y	UINT32	-
0x5610	0	Y	zplsena	Moog DCV	DrivePositionControl, ZPulseEnable ZPulseEnable	UINT8	rw	N	0…1	-
0x5611	0	Y	zplsdet	Moog DCV	DrivePositionControl, ZPulseDetected ZPulseDetected	UINT8	rw	N	0…1	-
0x5612	0	N	zplsset	Moog DCV	DrivePositionControl, ZPulseSet ZPulseSet	INT32	rw	Y	INT32	-
0x5613	0	Y	encraw	Moog DCV	DrivePositionControl, EncoderValue EncoderValue	INT32	rw	N	INT32	-
0x5614	0	N	refvel	Moog DCV	DrivePositionControl, ReferencingVelocity ReferencingVelocity	INT32	rw	Y	1000…2147483647	1000
0x5615	1	Y	veldem	Moog DCV	DriveVelocityControl, VelocityDemandValue VelocityDemandValue	INT32	ro	-	INT32	-
0x5615	2	N	veluni	Moog DCV	DriveVelocityControl, VelocityDemandValue Velocity_Unit	UINT8	ro	-	UINT8	0
0x5615	3	N	velprf	Moog DCV	DriveVelocityControl, VelocityDemandValue Velocity_Prefix	INT8	ro	-	INT8	0
0x5616	1	Y	posaccdem	Moog DCV	DrivePositionControl, AccelerationDemandvalue AccelerationDemandValue	INT32	ro	-	INT32	-
0x5617	0	Y	zplstrg	Moog DCV	DrivePositionControl, ZPulseTrigger ZPulseTrigger	UINT8	rw	Y	0…1	-
0x5618	0	Y	zplsclr	Moog DCV	DrivePositionControl, ZPulseClear ZPulseClear	UINT8	rw	N	UINT8	-
0x5619	0	Y	snssup	Moog DCV	DrivePositionControl, SensorSupplyEnable SensorSupplyEnable	UINT8	rw	Y	UINT8	-
0x561A	0	N	reffrc	Moog DCV	DrivePositionControl, ReferencingForce ReferencingForce	INT16	rw	Y	INT16	-
0x561B	0	N	refstp	Moog DCV	DrivePositionControl, ReferencingStop ReferencingStop	FLOAT32	rw	Y	FLOAT32	-
0x561C	0	Y	drvsts	Moog DCV	Drive, DriveStatus DriveStatus	UINT32	rw	N	UINT32	-
0x561D	0	Y	drvsts16	Moog DCV	Drive, 16BitDriveStatus 16BitDriveStatus	UINT16	rw	N	UINT16	-
0x561E	0	Y	posveldem	Moog DCV	DrivePositionControl, VelocityDemandValue VelocityDemandValue	INT32	ro	-	INT32	-
0x561F	0	N	drvtrdbdr	Moog DCV	Drive_ActualValueConditioning, BitRate BitRate	UINT8	rw	Y	0…7	2

Table 122: Object dictionary (part 23 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Object name, Block name / Parameter name	Data type	Access	Persistence	Value range	Default
0x5620	0	N	drvtrdmstslv	Moog DCV	Drive_ActualValueConditioning, MasterSlave / MasterSlave	UINT8	rw	N	1...1	1
0x5621	0	N	encrawset	Moog DCV	DrivePositionControl, EncoderSetValue / EncoderSetValue	INT32	rw	N	INT32	-
0x5700	0	Y	trjtyp	Moog DCV	DrivePositionControl, TrajectoryType / TrajectoryType	INT8	rw	Y	0...2	-
0x5701	0	Y	trjjmx	Moog DCV	DrivePositionControl, MaximumJerk / MaximumJerk	INT32	rw	Y	1...2147483647	10000000
0x5702	0	Y	trjamx	Moog DCV	DrivePositionControl, MaximumAcceleration / MaximumAcceleration	INT32	rw	Y	1...2147483647	1000000
0x5703	0	Y	trjvmx	Moog DCV	DrivePositionControl, MaximumVelocity / MaximumVelocity	INT32	rw	Y	1...2147483647	100000
0x5705	1	Y	trjcfgwng[0]	Moog DCV	DrivePositionControl, DrivePositionControl_Trajectory / ConfigWarning_0	UINT8	ro	-	UINT8	-
0x5705	2	Y	trjcfgwng[1]	Moog DCV	DrivePositionControl, DrivePositionControl_Trajectory / ConfigWarning_1	UINT8	ro	-	UINT8	-
0x5705	3	Y	trjcfgwng[2]	Moog DCV	DrivePositionControl, DrivePositionControl_Trajectory / ConfigWarning_2	UINT8	ro	-	UINT8	-
0x570C	1	Y	poslimwin[0]	Moog DCV	DrivePositionControl, DrivePositionControl / PositionWindow_0	INT32	rw	Y	0...2147483647	-
0x570C	2	Y	poslimwin[1]	Moog DCV	DrivePositionControl, DrivePositionControl / PositionWindow_1	INT32	rw	Y	0...2147483647	-
0x570D	0	Y	vellimwin	Moog DCV	DrivePositionControl, VelocityLimitWindow / VelocityLimitWindow	INT32	rw	Y	0...2147483647	-
0x570E	0	Y	vellim	Moog DCV	DrivePositionControl, VelocityLimit / VelocityLimit	INT32	rw	Y	INT32	-
0x5710	0	Y	prswin	Moog DCV	DrivePositionControl, PressureWindow / PressureWindow	INT16	rw	Y	0...32767	-
0x5711	0	Y	prsmaxwin	Moog DCV	DrivePositionControl, MaximumPressureWindow / MaximumPressureWindow	INT16	rw	Y	0...32767	-
0x5800	1	N	locpdtcob[0]	Moog DCV	LocalCAN, LocalCANTPdo_CobIdUsedByPdo / LocalCANTPdo1_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	\<NodeID\> (0x100B) + 0x0180
0x5800	2	N	locpdttrn[0]	Moog DCV	LocalCAN, LocalCANTPdo_TransmissionType / LocalCANTPdo1_TransmissionType	UINT8	rw	Y	UINT8	255
0x5800	3	N	locpdtinh[0]	Moog DCV	LocalCAN, LocalCANTPdo_InhibitTime / LocalCANTPdo1_InhibitTime	UINT16	rw	Y	UINT16	-

Table 122: Object dictionary (part 24 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x5800	5	N	loopdttim[0]	Moog DCV	LocalCAN, LocalCANTPdo1_EventTimer LocalCANTPdo1_EventTimer	UINT16	rw	Y	UINT16	-
0x5801	1	N	loopdtcob[1]	Moog DCV	LocalCAN, LocalCANTPdo2_CobIdUsedByPdo LocalCANTPdo2_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0280
0x5801	2	N	loopdttrn[1]	Moog DCV	LocalCAN, LocalCANTPdo2_TransmissionType LocalCANTPdo2_TransmissionType	UINT8	rw	Y	UINT8	255
0x5801	3	N	loopdtinh[1]	Moog DCV	LocalCAN, LocalCANTPdo2_InhibitTime LocalCANTPdo2_InhibitTime	UINT16	rw	Y	UINT16	-
0x5801	5	N	loopdttim[1]	Moog DCV	LocalCAN, LocalCANTPdo2_EventTimer LocalCANTPdo2_EventTimer	UINT16	rw	Y	UINT16	-
0x5802	1	N	loopdtcob[2]	Moog DCV	LocalCAN, LocalCANTPdo3_CobIdUsedByPdo LocalCANTPdo3_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0380
0x5802	2	N	loopdttrn[2]	Moog DCV	LocalCAN, LocalCANTPdo3_TransmissionType LocalCANTPdo3_TransmissionType	UINT8	rw	Y	UINT8	255
0x5802	3	N	loopdtinh[2]	Moog DCV	LocalCAN, LocalCANTPdo3_InhibitTime LocalCANTPdo3_InhibitTime	UINT16	rw	Y	UINT16	-
0x5802	5	N	loopdttim[2]	Moog DCV	LocalCAN, LocalCANTPdo3_EventTimer LocalCANTPdo3_EventTimer	UINT16	rw	Y	UINT16	-
0x5803	1	N	loopdtcob[3]	Moog DCV	LocalCAN, LocalCANTPdo4_CobIdUsedByPdo LocalCANTPdo4_CobIdUsedByPdo	UINT32	rw	Y	1...0x800007FF	<NodeID> (0x100B) + 0x0480
0x5803	2	N	loopdttrn[3]	Moog DCV	LocalCAN, LocalCANTPdo4_TransmissionType LocalCANTPdo4_TransmissionType	UINT8	rw	Y	UINT8	255
0x5803	3	N	loopdtinh[3]	Moog DCV	LocalCAN, LocalCANTPdo4_InhibitTime LocalCANTPdo4_InhibitTime	UINT16	rw	Y	UINT16	-
0x5803	5	N	loopdttim[3]	Moog DCV	LocalCAN, LocalCANTPdo4_EventTimer LocalCANTPdo4_EventTimer	UINT16	rw	Y	UINT16	-
0x5829	0	Y	sncpgn	Moog DCV	DrivePositionControl, SynchronisationGain SynchronisationGain	FLOAT32	rw	Y	FLOAT32	-
0x582A	0	Y	posdemflt	Moog DCV	DrivePositionControl, PositionDemandFilter PositionDemandFilter	FLOAT32	rw	Y	0.00...+inf	-
0x582B	0	Y	posdemfltout	Moog DCV	DrivePositionControl, PositionDemandFilterOutput PositionDemandFilterOutput	FLOAT32	ro	-	FLOAT32	-
0x5830	0	N	sncnumaxs	Moog DCV	DrivePositionControl, NumberOfAxisToSyncronize NumberOfAxisToSyncronize	INT8	rw	Y	1...4	3
0x5857	1...16	N	cmppr-signswitrs[0...15]	Moog DCV	ValvePressureControl, IntegratorGainSwitchThreshold IntegratorGainSwitchThreshold	UINT32	rw	Y	UINT32	5000

Table 122: Object dictionary (part 25 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x5858	1…16	Y	cmpprsfbspgn_2[0…15]	Moog DCV	PressureControl, SpoolPositionFeedBackGain_2 SpoolPositionFeedBackGain_2	FLOAT32	rw	Y	0.00…+inf	-
0x585C	0	N	poshldtgt	Moog DCV	DrivePositionControl, HoldSetpointTarget HoldSetpointTarget	INT8	rw	Y	0…2	-
0x585D	0	N	cmpfrcroda	Moog DCV	ValvePressureControl, CylinderRodDiameterA CylinderRodDiameterA	FLOAT32	rw	Y	0.00…<CylinderPistonDiameter> (0x585F)	-
0x585E	0	N	cmpfrcrodb	Moog DCV	ValvePressureControl, CylinderRodDiameterB CylinderRodDiameterB	FLOAT32	rw	Y	0.00…<CylinderPistonDiameter> (0x585F)	-
0x585F	0	N	cmpfrcpst	Moog DCV	ValvePressureControl, CylinderPistonDiameter CylinderPistonDiameter	FLOAT32	rw	Y	0.00…+inf	1000000.00
0x5860	1	Y	prsintprlvar[0]	Moog DCV	ValvePressureControl, IntegratorPreloadValue_0	INT16	ro	-	INT16	-
0x5860	2	Y	prsintprlvar[1]	Moog DCV	ValvePressureControl, IntegratorPreloadValue_1	INT16	ro	-	INT16	-
0x5861	1…16	Y	prsintfbapgn[0…15]	Moog DCV	ValvePressureControl, IntegralProportionalPartPGain IntegralProportionalPartPGain	FLOAT32	rw	Y	FLOAT32	-
0x5862	0	Y	cmpprsdt1_2	Moog DCV	ValvePressureControl, kdFeedbackOutput kdFeedbackOutput	FLOAT32	ro	-	FLOAT32	-
0x5863	1…16	Y	cmpprsdgn_2[0…15]	Moog DCV	ValvePressureControl, DifferentiatorGain_2 DifferentiatorGain_2	FLOAT32	rw	Y	FLOAT32	-
0x5864	1…16	Y	cmpprsdtm_2[0…15]	Moog DCV	ValvePressureControl, DifferentiatorT1_2 DifferentiatorT1_2	FLOAT32	rw	Y	0.00…+inf	-
0x5865	1…16	Y	prsoutuppl[0…15]	Moog DCV	ValvePressureControl, UpperControllerOutputLimit UpperControllerOutputLimit	INT16	rw	Y	<LowerOutputLimit> (0x230B[n])…32767	16384
0x5866	1…16	Y	prsoutlowl[0…15]	Moog DCV	ValvePressureControl, LowerControllerOutputLimit LowerControllerOutputLimit	INT16	rw	Y	-32768…<UpperOutputLimit> (0x230A[n])	-16384
0x5867	1…16	Y	prsfwp[0…15]	Moog DCV	ValvePressureControl, FeedForwardGain FeedForwardGain	FLOAT32	rw	Y	FLOAT32	-
0x5868	1…16	Y	prsffwpar[0…15]	Moog DCV	ValvePressureControl, FeedForwardParameter FeedForwardParameter	UINT32	rw	Y	UINT32	0x63800110
0x5869	0	Y	prsintprlpgn	Moog DCV	ValvePressureControl, IntegratorPreloadGain IntegratorPreloadGain	FLOAT32	rw	Y	FLOAT32	-
0x586A	0	Y	prsintprlpar	Moog DCV	ValvePressureControl, IntegratrPreloadParameter IntegratrPreloadParameter	UINT32	rw	Y	UINT32	0x63800110
0x586B	0	Y	prsintprlmod	Moog DCV	ValvePressureControl, IntegratorPreloadMode IntegratorPreloadMode	UINT8	rw	Y	0…2	-
0x586C	1…16	Y	splprsmod[0…15]	Moog DCV	ValvePressureControl, pQSwitchingMode pQSwitchingMode	UINT8	rw	Y	0…2	-

Table 122: Object dictionary (part 26 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x586D	0	Y	prsdemsgnmod	Moog DCV	ValvePressureControl, PressureDemandSignMode PressureDemandSignMode	UINT8	rw	Y	0…1	-
0x586E	1	Y	dirparset[0]	Moog DCV	ValvePressureControl, ValvePressureControl DirectionalParameterSet_0	UINT8	rw	Y	1…16	1
0x586E	2	Y	dirparset[1]	Moog DCV	ValvePressureControl, ValvePressureControl DirectionalParameterSet_1	UINT8	rw	Y	1…16	1
0x586F	0	Y	dirparsetmod	Moog DCV	ValvePressureControl, DirectionalDependantParameterSetMode DirectionalDependantParameterSetMode	UINT8	rw	Y	0…1	-
0x5870	1…16	Y	prsffwofs[0…15]	Moog DCV	ValvePressureControl, FeedForwardOffset FeedForwardOffset	INT16	rw	Y	INT16	-
0x5A00	0	N	locpdtmapnum	Moog DCV	LocalCAN, TPdoMapping_NumberOfMappedApplicParaInPdo TPdoMapping_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5A00	1…8	N	locpdtmap[0…7]	Moog DCV	LocalCAN, TPdo_Mapping LocalTPdo1_ApplicPara1	UINT32	rw	Y	UINT32	-
0x5A01	0	N	locpdtmapnum	Moog DCV	LocalCAN, TPdoMapping_NumberOfMappedApplicParaInPdo TPdoMapping_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5A01	1…8	N	locpdtmap[8…15]	Moog DCV	LocalCAN, TPdo_Mapping LocalTPdo2_ApplicPara1	UINT32	rw	Y	UINT32	-
0x5A02	0	N	locpdtmapnum	Moog DCV	LocalCAN, TPdoMapping_NumberOfMappedApplicParaInPdo TPdoMapping_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5A02	1…8	N	locpdtmap[16…23]	Moog DCV	LocalCAN, TPdo_Mapping LocalTPdo3_ApplicPara1	UINT32	rw	Y	UINT32	-
0x5A03	0	N	locpdtmapnum	Moog DCV	LocalCAN, TPdoMapping_NumberOfMappedApplicParaInPdo TPdoMapping_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	0
0x5A03	1…8	N	locpdtmap[24…31]	Moog DCV	LocalCAN, TPdo_Mapping LocalTPdo4_ApplicPara1	UINT32	rw	Y	UINT32	-
0x5A08	1	N	locpdttrnman[0]	Moog DCV	LocalCAN, LocalCANTPdoManufacturerTransmissionType LocalCANTPdo1_ManufacturerTransmissionType	UINT8	rw	Y	UINT8	-
0x5A08	2	N	locpdttrnman[1]	Moog DCV	LocalCAN, LocalCANTPdoManufacturerTransmissionType LocalCANTPdo2_ManufacturerTransmissionType	UINT8	rw	Y	UINT8	-
0x5A08	3	N	locpdttrnman[2]	Moog DCV	LocalCAN, LocalCANTPdoManufacturerTransmissionType LocalCANTPdo3_ManufacturerTransmissionType	UINT8	rw	Y	UINT8	-
0x5A08	4	N	locpdttrnman[3]	Moog DCV	LocalCAN, LocalCANTPdoManufacturerTransmissionType LocalCANTPdo4_ManufacturerTransmissionType	UINT8	rw	Y	UINT8	-
0x5B00	0	N	locmodide	Moog DCV	LocalCAN, LocalCAN_ModuleIdentifier LocalCAN_ModuleIdentifier	UINT8	rw	Y	1…127	0x7F
0x5B01	0	N	locbdr	Moog DCV	LocalCAN, LocalCAN_Bitrate LocalCAN_Bitrate	UINT32	rw	Y	0…1000000	500000

Table 122: Object dictionary (part 27 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x5B02	0	Y	locsrn	Moog DCV	LocalCAN, LocalCAN_StartRemoteNode / LocalCAN_StartRemoteNode	UINT8	rw	N	UINT8	-
0x5B03	0	N	locpdttrg	Moog DCV	LocalCAN, LocalCAN_TPdoTrigger / LocalCAN_TPdoTrigger	UINT8	rw	N	0…4	-
0x5B10	0	N	locrempar	Moog DCV	LocalCAN, LocalCAN_RemoteParameter / LocalCAN_RemoteParameter	UINT32	rw	N	UINT32	-
0x5B11	0	N	locremadr	Moog DCV	LocalCAN, LocalCAN_RemoteParameterAdress / LocalCAN_RemoteParameterAdress	UINT32	rw	N	UINT32	-
0x5B12	0	N	locremnod	Moog DCV	LocalCAN, LocalCAN_RemoteNodeId / LocalCAN_RemoteNodeId	UINT8	rw	N	0…127	-
0x5B13	0	N	locremtrn	Moog DCV	LocalCAN, LocalCAN_RemoteTransmission / LocalCAN_RemoteTransmission	INT8	rw	N	-1…2	-
0x5B14	0	N	locbuster	Moog DCV	LocalCAN, LocalCAN_TerminationResistor / LocalCAN_TerminationResistor	UINT8	rw	Y	0…1	-
0x5E41	1	N	digouttyp[0]	Moog DCV	ValveDigitalOutput, DigitalOutputConfiguration0	UINT8	ro	-	0…4	3
0x5E41	2	N	digouttyp[1]	Moog DCV	ValveDigitalOutput, DigitalOutputConfiguration1	UINT8	ro	-	0…4	0
0x5E42	1	N	digout[0]	Moog DCV	ValveDigitalOutput, DigitalOutputSetpoint0	UINT8	rw	N	0…1	-
0x5E42	2	Y	digout[1]	Moog DCV	ValveDigitalOutput, DigitalOutputSetpoint1	UINT8	rw	N	0…1	-
0x5E43	0	N	sfwenasig	Moog DCV	EnableSignal, SoftwareEnableSignal / SoftwareEnableSignal	UINT8	ro	-	0…1	0
0x5E44	1	Y	digoutmon[0]	Moog DCV	ValveDigitalOutput, ValveDigitalOutput DigitalOutputValue_0	UINT8	ro	-	UINT8	-
0x5E44	2	Y	digoutmon[1]	Moog DCV	ValveDigitalOutput, ValveDigitalOutput DigitalOutputValue_1	UINT8	ro	-	UINT8	-
0x6040	0	Y	ctlwrd	CiA 408	Device, ControlWord / ControlWord	UINT16	rw	N	UINT16	-
0x6041	0	Y	stswrd	CiA 408	Device, StatusWord / StatusWord	UINT16	ro	-	UINT16	-
0x6042	0	Y	devmod	CiA 408	Device, DeviceMode / DeviceMode	INT8	rw	N	1…4	<DeviceModeDefault> (0x4042)
0x6043	0	Y	ctlmod	CiA 408	Device, ControlMode / ControlMode	INT8	rw	N	-1…14	<ControlModeDefault> (0x4043)

Table 122: Object dictionary (part 28 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name / Parameter name	Data type	Access	Persistence	Value range	Default
0x604F	0	Y	locmod	CiA 408	Device, Local / Local	INT8	rw	Y	-128…1	-
0x6050	0	N	devver	CiA 408	Device, DeviceVersion / DeviceVersion	CHAR[64]	ro	-	-	-
0x6051	0	Y	devcodnum	CiA 408	Device, CodeNumber / CodeNumber	UINT16	rw	Y	UINT16	-
0x6052	0	N	sernum	CiA 408	Device, SerialNumber / SerialNumber	CHAR[64]	ro	-	-	-
0x6053	0	N	devdsc	CiA 408	Device, Description / Description	CHAR[64]	rw	Y	-	-
0x6054	0	N	devmdldsc	CiA 408	Device, ModelDescription / ModelDescription	CHAR[64]	ro	-	-	-
0x6055	0	N	devmdlurl	CiA 408	Device, ModelURL / ModelURL	CHAR[64]	ro	-	-	www.moog.com
0x6056	0	Y	devprmcod	CiA 408	Device, ParameterSetCode / ParameterSetCode	UINT8	rw	Y	0…254	-
0x6057	0	N	devvennam	CiA 408	Device, VendorName / VendorName	CHAR[64]	ro	-	-	MOOG
0x605F	0	Y	devcap	CiA 408	Device, Capability / Capability	UINT32	ro	-	16777216…1061134336	1061093376
0x6200	0	N	drvtrdmax	CiA 408	Drive_ActualValueConditioning, MaxInterfaceNo / MaxInterfaceNo	UINT8	ro	-	UINT8	8
0x6201	0	N	drvtrdnum	CiA 408	Drive_ActualValueConditioning, InterfaceNo / InterfaceNo	UINT8	rw	N	1…<MaxInterfaceNo> (0x6200)	1
0x6202	0	N	drvtrdtyp	CiA 408	Drive_ActualValueConditioning, Type / Type	INT8	rw	N	-61…69	-
0x6203	0	N	drvtrdsgn	CiA 408	Drive_ActualValueConditioning, Sign / Sign	INT8	rw	N	-1…1	1
0x6204	1	Y	drvtrdval	CiA 408	Drive_ActualValueConditioning, ActualValue / ActualValue	INT32	ro	-	INT32	-
0x6210	1	Y	drvtrditfval[0]	CiA 408	Drive_ActValCond, ActualValue / ActualValue	INT32	ro	-	INT32	-
0x6211	1	Y	drvtrditfval[1]	CiA 408	Drive_ActValCond, ActualValue / ActualValue	INT32	ro	-	INT32	-
0x6212	1	Y	drvtrditfval[2]	CiA 408	Drive_ActValCond, ActualValue / ActualValue	INT32	ro	-	INT32	-
0x6213	1	Y	drvtrditfval[3]	CiA 408	Drive_ActValCond, ActualValue / ActualValue	INT32	ro	-	INT32	-

Table 122: Object dictionary (part 29 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name / Parameter name	Data type	Access	Persistence	Value range	Default
0x6214	1	Y	drvtrditfval[4]	CiA 408	Drive_ActValCond, ActualValue / ActualValue	INT32	ro	-	INT32	-
0x6215	1	Y	drvtrditfval[5]	CiA 408	Drive_ActValCond, ActualValue / ActualValue	INT32	ro	-	INT32	-
0x6216	1	Y	drvtrditfval[6]	CiA 408	Drive_ActValCond, ActualValue / ActualValue	INT32	ro	-	INT32	-
0x6217	1	Y	drvtrditfval[7]	CiA 408	Drive_ActValCond, ActualValue / ActualValue	INT32	ro	-	INT32	-
0x6220	1	N	drvtrdprsmin	CiA 408	Drive_ActualValueConditioning, MinimumPressure / MinimumPressure	INT32	rw	N	-32768…32767	-16384
0x6221	1	N	drvtrdprsmax	CiA 408	Drive_ActualValueConditioning, MaximumPressure / MaximumPressure	INT32	rw	N	-32768…32767	16384
0x6222	1	N	drvtrdprsare	CiA 408	Drive_ActualValueConditioning, PressureArea / PressureArea	INT32	rw	N	-32768…32767	-
0x6223	1	N	drvtrdprsofs	CiA 408	Drive_ActualValueConditioning, PressureOffset / PressureOffset	INT32	rw	N	-32768…32767	-
0x6224	1	N	drvtrdprssigmin	CiA 408	Drive_ActualValueConditioning, MinimumSignal / MinimumSignal	INT32	rw	N	-32768…32767	-16384
0x6225	1	N	drvtrdprssigmax	CiA 408	Drive_ActualValueConditioning, MaximumSignal / MaximumSignal	INT32	rw	N	-32768…32767	16384
0x6228	1	N	drvtrdgenmin	CiA 408	Drive_ActualValueConditioning, GeneralInputMinimum / GeneralInputMinimum	INT32	rw	N	INT32	-
0x6229	1	N	drvtrdgenmax	CiA 408	Drive_ActualValueConditioning, GeneralInputMaximum / GeneralInputMaximum	INT32	rw	N	INT32	16384
0x622B	1	N	drvtrdgenofs	CiA 408	Drive_ActualValueConditioning, GeneralInputOffset / GeneralInputOffset	INT32	rw	N	INT32	-
0x622C	1	N	drvtrdgensigmin	CiA 408	Drive_ActualValueConditioning, GeneralInputMinimumSignal / GeneralInputMinimumSignal	INT32	rw	N	INT32	-
0x622D	1	N	drvtrdgensigmax	CiA 408	Drive_ActualValueConditioning, GeneralInputMaximumSignal / GeneralInputMaximumSignal	INT32	rw	N	INT32	16384
0x6230	1	N	drvtrdminref	CiA 408	Drive_ActualValueConditioning, MinimumReference / MinimumReference	INT32	rw	N	INT32	-1
0x6231	1	N	drvtrdmaxref	CiA 408	Drive_ActualValueConditioning, MaximumReference / MaximumReference	INT32	rw	N	INT32	1
0x6232	1	N	drvtrdltc	CiA 408	Drive_ActualValueConditioning, T1 / T1	UINT32	rw	N	UINT32	-
0x6233	1	N	drvtrdmintf	CiA 408	Drive_ActualValueConditioning, MinimumInterface / MinimumInterface	INT32	rw	N	INT32	-1

Table 122: Object dictionary (part 30 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Object name, Block name, Parameter name	Data type	Access	Persistence	Value range	Default
0x6234	1	N	drvtrdmaxitf	CiA 408	Drive_ActualValueConditioning, MaximumInterface MaximumInterface	INT32	rw	N	INT32	1
0x6240	1	N	drvtrdres	CiA 408	Drive_ActualValueConditioning, Resolution Resolution	INT32	rw	N	INT32	1
0x6241	1	N	drvtrdofs	CiA 408	Drive_ActualValueConditioning, PositionOffset PositionOffset	INT32	rw	N	INT32	-
0x6242	1	N	drvtrdzroshf	CiA 408	Drive_ActualValueConditioning, ZeroShift ZeroShift	UINT8	rw	N	UINT8	-
0x6243	1	N	drvtrdbitsiz	CiA 408	Drive_ActualValueConditioning, BitSize BitSize	UINT8	rw	N	0...32	24
0x6244	1	N	drvtrdc	CiA 408	Drive_ActualValueConditioning, StartStopEncoderTime StartStopEncoderTime	INT32	rw	N	0...2147483647	-
0x6245	1	N	drvtrdstastotyp	CiA 408	Drive_ActualValueConditioning, StartStopEncoderType StartStopEncoderType	INT8	rw	N	0...127	1
0x6300	1	Y	splset	CiA 408	ValvePositionControl, SplSetpoint SplSetpoint	INT16	rw	N	INT16	-
0x6300	2	N	spluni	CiA 408	ValvePositionControl, SplSetpoint Unit	UINT8	ro	-	UINT8	0
0x6300	3	N	splprf	CiA 408	ValvePositionControl, SplSetpoint Prefix	INT8	ro	-	INT8	0
0x6301	1	Y	splval	CiA 408	ValvePositionControl, SplActualValue SplActualValue	INT16	ro	-	INT16	-
0x6301	2	N	spluni	CiA 408	ValvePositionControl, SplActualValue Unit	UINT8	ro	-	UINT8	0
0x6301	3	N	splprf	CiA 408	ValvePositionControl, SplActualValue Prefix	INT8	ro	-	INT8	0
0x6310	1	Y	spldem	CiA 408	ValvePositionControl_DemandValueGeneratorSplDemand, SplDemandValue SplDemandValue	INT16	ro	-	INT16	-
0x6310	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGeneratorSplDemand, SplDemandValue Unit	UINT8	ro	-	UINT8	0
0x6310	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGeneratorSplDemand, SplDemandValue Prefix	INT8	ro	-	INT8	0
0x6311	1	Y	spldemref	CiA 408	ValvePositionControl_DemandValueGenerator, SplReferenceValue SplReferenceValue	INT16	ro	-	INT16	16384
0x6311	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator, SplReferenceValue Unit	UINT8	ro	-	UINT8	0
0x6311	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator, SplReferenceValue Prefix	INT8	ro	-	INT8	0

Table 122: Object dictionary (part 31 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name / Parameter name	Data type	Access	Persistence	Value range	Default
0x6314	1	Y	splsethld	CiA 408	ValvePositionControl_DemandValueGeneratorSplHoldSetPoint, SplHoldSetPoint	INT16	rw	Y	INT16	-
0x6314	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGeneratorSplHoldSetPoint, SplHoldSetPoint Unit	UINT8	ro	-	UINT8	0
0x6314	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGeneratorSplHoldSetPoint, SplHoldSetPoint Prefix	INT8	ro	-	INT8	0
0x6320	1	Y	spllimupp	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, UpperLimit	INT16	rw	Y	<LowerLimit>(0x6321)…32767	16384
0x6320	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, UpperLimit Unit	UINT8	ro	-	UINT8	0
0x6320	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, UpperLimit Prefix	INT8	ro	-	INT8	0
0x6321	1	Y	spllimlow	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, LowerLimit	INT16	rw	Y	-32768…<UpperLimit>(0x6320)	-16384
0x6321	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, LowerLimit Unit	UINT8	ro	-	UINT8	0
0x6321	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, LowerLimit Prefix	INT8	ro	-	INT8	0
0x6322	0	Y	spldemfct	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Factor	UINT32	rw	Y	UINT32	0x10001
0x6323	1	Y	spldemofs	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Offset	INT16	rw	Y	INT16	-
0x6323	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Offset Unit	UINT8	ro	-	UINT8	0
0x6323	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Offset Prefix	INT8	ro	-	INT8	0
0x6324	1	Y	splzrocor	CiA 408	ValvePositionControl_DemandValueGenerator_ZeroCorrection, Offset	INT16	rw	Y	INT16	-
0x6324	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_ZeroCorrection, Offset Unit	UINT8	ro	-	UINT8	0
0x6324	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_ZeroCorrection, Offset Prefix	INT8	ro	-	INT8	0
0x6330	0	Y	splrmptyp	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Type	INT8	rw	Y	0…3	-
0x6331	1	Y	splrmpacl	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, AccelerationTime	UINT16	rw	Y	UINT16	-
0x6331	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, AccelerationTime Unit	UINT8	ro	-	UINT8	3

Table 122: Object dictionary (part 32 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x6331	3	Y	splrmpaclpf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, AccelerationTime AccelerationTime_Prefix	INT8	rw	Y	-4…0	-3
0x6332	1	Y	splrmpaclpos	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, AccelerationTimePositive AccelerationTimePositive	UINT16	rw	Y	UINT16	-
0x6332	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, AccelerationTimePositive Unit	UINT8	ro	-	UINT8	3
0x6332	3	Y	splrmpaclposprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, AccelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3
0x6333	1	Y	splrmpacineg	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, AccelerationTimeNegative AccelerationTimeNegative	UINT16	rw	Y	UINT16	-
0x6333	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, AccelerationTimeNegative Unit	UINT8	ro	-	UINT8	3
0x6333	3	Y	splrmpacinegprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, AccelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3
0x6334	1	Y	splrmpdcl	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, DecelerationTime DecelerationTime	UINT16	rw	Y	UINT16	-
0x6334	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, DecelerationTime Unit	UINT8	ro	-	UINT8	3
0x6334	3	Y	splrmpdclpf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, DecelerationTime_Prefix	INT8	rw	Y	-4…0	-3
0x6335	1	Y	splrmpdclpos	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, DecelerationTimePositive DecelerationTimePositive	UINT16	rw	Y	UINT16	-
0x6335	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, DecelerationTimePositive Unit	UINT8	ro	-	UINT8	3
0x6335	3	Y	splrmpdclposprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, DecelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3
0x6336	1	Y	splrmpdclneg	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, DecelerationTimeNegative DecelerationTimeNegative	UINT16	rw	Y	UINT16	-
0x6336	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, DecelerationTimeNegative Unit	UINT8	ro	-	UINT8	3
0x6336	3	Y	splrmpdclnegprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, DecelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3
0x6340	0	Y	spldirtyp	CiA 408	ValvePositionControl_DemandValueGenerator, DirectionalDependentGain_Type DirectionalDependentGain_Type	INT8	rw	Y	0…1	-
0x6341	0	Y	spldirfct	CiA 408	ValvePositionControl_DemandValueGenerator, DirectionalDependentGain_Factor DirectionalDependentGain_Factor	UINT32	rw	Y	UINT32	0x10001
0x6342	0	Y	spldbdtyp	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensationType DeadbandCompensationType	INT8	rw	Y	0…2	-

Table 122: Object dictionary (part 33 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x6343	1	Y	spldbdsida	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_ASide / DeadbandCompensation_ASide	INT16	rw	Y	0...16384	-
0x6343	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_ASide / Unit	UINT8	ro	-	UINT8	0
0x6343	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_ASide / Prefix	INT8	ro	-	INT8	0
0x6344	1	Y	spldbdsidb	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_BSide / DeadbandCompensation_BSide	INT16	rw	Y	0...16384	-
0x6344	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_BSide / Unit	UINT8	ro	-	UINT8	0
0x6344	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_BSide / Prefix	INT8	ro	-	INT8	0
0x6345	1	Y	spldbdtrs	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_Threshold old / DeadbandCompensation_Threshold	INT16	rw	Y	0...16383	-
0x6345	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_Threshold old / Unit	UINT8	ro	-	UINT8	0
0x6345	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator, DeadbandCompensation_Threshold old / Prefix	INT8	ro	-	INT8	0
0x6346	0	Y	splchrtyp	CiA 408	ValvePositionControl_DemandValueGenerator, CharacteristicCompensation_Type / CharacteristicCompensation_Type	INT8	rw	Y	-1...0	-
0x6350	1	Y	splctldvn	CiA 408	ValvePositionControl, SplControlDeviation / SplControlDeviation	INT16	ro	-	INT16	-
0x6350	2	N	spluni	CiA 408	ValvePositionControl, SplControlDeviation / Unit	UINT8	ro	-	UINT8	0
0x6350	3	N	splprf	CiA 408	ValvePositionControl, SplControlDeviation / Prefix	INT8	ro	-	INT8	0
0x6351	0	Y	splmontyp	CiA 408	ValvePositionControl_ControlMonitoring, Type / Type	INT8	rw	Y	0...1	-
0x6352	1	Y	splmontim	CiA 408	ValvePositionControl_ControlMonitoring, DelayTime / DelayTime	UINT16	rw	Y	UINT16	30
0x6352	2	N	timuni	CiA 408	ValvePositionControl_ControlMonitoring, DelayTime / Unit	UINT8	ro	-	UINT8	3
0x6352	3	N	timprf	CiA 408	ValvePositionControl_ControlMonitoring, DelayTime / Prefix	INT8	ro	-	INT8	-3
0x6354	1	Y	splmonupp	CiA 408	ValvePositionControl_ControlMonitoring, UpperThreshold / UpperThreshold	INT16	rw	Y	INT16	512

Table 122: Object dictionary (part 34 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6354	2	N	spluni	CiA 408	ValvePositionControl_ControlMonitoring, UpperThreshold Unit	UINT8	ro	-	UINT8	0
0x6354	3	N	splprf	CiA 408	ValvePositionControl_ControlMonitoring, UpperThreshold Prefix	INT8	ro	-	INT8	0
0x6355	1	Y	splmonlow	CiA 408	ValvePositionControl_ControlMonitoring, LowerThreshold LowerThreshold	INT16	rw	Y	INT16	-512
0x6355	2	N	spluni	CiA 408	ValvePositionControl_ControlMonitoring, LowerThreshold Unit	UINT8	ro	-	UINT8	0
0x6355	3	N	splprf	CiA 408	ValvePositionControl_ControlMonitoring, LowerThreshold Prefix	INT8	ro	-	INT8	0
0x6380	1	Y	prsset	CiA 408	ValvePressureControl, PrsSetpoint PrsSetpoint	INT16	rw	N	INT16	-
0x6380	2	N	prsuni	CiA 408	ValvePressureControl, PrsSetpoint Unit	UINT8	ro	-	UINT8	0
0x6380	3	N	prsprf	CiA 408	ValvePressureControl, PrsSetpoint Prefix	INT8	ro	-	INT8	0
0x6381	1	Y	prsval	CiA 408	ValvePressureControl, PrsActualValue PrsActualValue	INT16	ro	-	INT16	-
0x6381	2	N	prsuni	CiA 408	ValvePressureControl, PrsActualValue Unit	UINT8	ro	-	UINT8	0
0x6381	3	N	prsprf	CiA 408	ValvePressureControl, PrsActualValue Prefix	INT8	ro	-	INT8	0
0x6390	1	Y	prsdem	CiA 408	ValvePressureControl_DemandValueGenerator, PrsDemandValue PrsDemandValue	INT16	ro	-	INT16	-
0x6390	2	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator, PrsDemandValue Unit	UINT8	ro	-	UINT8	0
0x6390	3	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator, PrsDemandValue Prefix	INT8	ro	-	INT8	0
0x6391	1	Y	prsdemref	CiA 408	ValvePressureControl_DemandValueGenerator, PrsReferenceValue PrsReferenceValue	INT16	ro	-	INT16	16384
0x6391	2	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator, PrsReferenceValue Unit	UINT8	ro	-	UINT8	0
0x6391	3	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator, PrsReferenceValue Prefix	INT8	ro	-	INT8	0
0x6394	1	Y	prssethld	CiA 408	ValvePressureControl_DemandValueGenerator, PrsHoldSetpoint PrsHoldSetpoint	INT16	rw	Y	INT16	-
0x6394	2	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator, PrsHoldSetpoint Unit	UINT8	ro	-	UINT8	0

Table 122: Object dictionary (part 35 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x6394	3	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator, PrsHoldSetpoint Prefix	INT8	ro	-	INT8	0
0x63A0	1	Y	prslimupp	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, UpperLimit	INT16	rw	Y	<LowerLimit>(0x63A1)...32767	16384
0x63A0	2	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, UpperLimit Unit	UINT8	ro	-	UINT8	0
0x63A0	3	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, UpperLimit Prefix	INT8	ro	-	INT8	0
0x63A1	1	Y	prslimlow	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, LowerLimit	INT16	rw	Y	-32768...<UpperLimit>(0x63A0)	-16384
0x63A1	2	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, LowerLimit Unit	UINT8	ro	-	UINT8	0
0x63A1	3	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, LowerLimit Prefix	INT8	ro	-	INT8	0
0x63A2	0	Y	prsdemfct	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Factor Factor	UINT32	rw	Y	UINT32	0x10001
0x63A3	1	Y	prsdemofs	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Offset Offset	INT16	rw	Y	INT16	-
0x63A3	2	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Offset Unit	UINT8	ro	-	UINT8	0
0x63A3	3	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Offset Prefix	INT8	ro	-	INT8	0
0x63B0	0	Y	prsrmptyp	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Type Type	INT8	rw	Y	0...3	-
0x63B1	1	Y	prsrmpacl	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, AccelerationTime AccelerationTime	UINT16	rw	Y	UINT16	-
0x63B1	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, AccelerationTime Unit	UINT8	ro	-	UINT8	3
0x63B1	3	Y	prsrmpaclprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, AccelerationTime_Prefix	INT8	rw	Y	-4...0	-3
0x63B2	1	Y	prsrmpaclpos	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, AccelerationTimePositive AccelerationTimePositive	UINT16	rw	Y	UINT16	-
0x63B2	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, AccelerationTimePositive Unit	UINT8	ro	-	UINT8	3
0x63B2	3	Y	prsrmpaclposprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, AccelerationTimePositive_Prefix	INT8	rw	Y	-4...0	-3
0x63B3	1	Y	prsrmpaclneg	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, AccelerationTimeNegative AccelerationTimeNegative	UINT16	rw	Y	UINT16	-

Table 122: Object dictionary (part 36 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name, Parameter name	Data type	Access	Persistence	Value range	Default
0x63B3	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, AccelerationTimeNegative Unit	UINT8	ro	-	UINT8	3
0x63B3	3	Y	prsrmpaclnegprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, AccelerationTimeNegative_Prefix	INT8	rw	Y	-4...0	-3
0x63B4	1	Y	prsrmpdcl	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTime	UINT16	rw	Y	UINT16	-
0x63B4	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTime Unit	UINT8	ro	-	UINT8	3
0x63B4	3	Y	prsrmpdclprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTime_Prefix	INT8	rw	Y	-4...0	-3
0x63B5	1	Y	prsrmpdclpos	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimePositive	UINT16	rw	Y	UINT16	-
0x63B5	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimePositive Unit	UINT8	ro	-	UINT8	3
0x63B5	3	Y	prsrmpdclposprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimePositive_Prefix	INT8	rw	Y	-4...0	-3
0x63B6	1	Y	prsrmpdclneg	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimeNegative	UINT16	rw	Y	UINT16	-
0x63B6	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimeNegative Unit	UINT8	ro	-	UINT8	3
0x63B6	3	Y	prsrmpdclnegprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, DecelerationTimeNegative_Prefix	INT8	rw	Y	-4...0	-3
0x63D0	1	Y	prsctldvn	CiA 408	ValvePressureControl, PrsControlDeviation PrsControlDeviation	INT16	ro	-	INT16	-
0x63D0	2	N	prsuni	CiA 408	ValvePressureControl, PrsControlDeviation Unit	UINT8	ro	-	UINT8	0
0x63D0	3	N	prsprf	CiA 408	ValvePressureControl, PrsControlDeviation Prefix	INT8	ro	-	INT8	0
0x63D1	0	Y	prsmontyp	CiA 408	ValvePressureControl_ControlMonitoring, Type Type	INT8	rw	Y	0...1	-
0x63D2	1	Y	prsmontim	CiA 408	ValvePressureControl_ControlMonitoring, DelayTime DelayTime	UINT16	rw	Y	UINT16	30
0x63D2	2	N	timuni	CiA 408	ValvePressureControl_ControlMonitoring, DelayTime Unit	UINT8	ro	-	UINT8	3
0x63D2	3	N	timprf	CiA 408	ValvePressureControl_ControlMonitoring, DelayTime Prefix	INT8	ro	-	INT8	-3
0x63D4	1	Y	prsmonupp	CiA 408	ValvePressureControl_ControlMonitoring, UpperThreshold UpperThreshold	INT16	rw	Y	INT16	512

Table 122: Object dictionary (part 37 of 39)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name / Parameter name	Data type	Access	Persistence	Value range	Default
0x63D4	2	N	prsuni	CiA 408	ValvePressureControl_ControlMonitoring, UpperThreshold / Unit	UINT8	ro	-	UINT8	0
0x63D4	3	N	prsprf	CiA 408	ValvePressureControl_ControlMonitoring, UpperThreshold / Prefix	INT8	ro	-	INT8	0
0x63D5	1	Y	prsmonlow	CiA 408	ValvePressureControl_ControlMonitoring, LowerThreshold / LowerThreshold	INT16	rw	Y	INT16	-512
0x63D5	2	N	prsuni	CiA 408	ValvePressureControl_ControlMonitoring, LowerThreshold / Unit	UINT8	ro	-	UINT8	0
0x63D5	3	N	prsprf	CiA 408	ValvePressureControl_ControlMonitoring, LowerThreshold / Prefix	INT8	ro	-	INT8	0
0x6500	1	Y	velset	CiA 408	DriveVelocityControl, Setpoint / Setpoint	INT32	rw	N	-10000000…10000000	-
0x6500	2	N	veluni	CiA 408	DriveVelocityControl, Setpoint / Velocity_Unit	UINT8	ro	-	UINT8	0
0x6500	3	N	velprf	CiA 408	DriveVelocityControl, Setpoint / Velocity_Prefix	INT8	ro	-	INT8	0
0x6501	0	Y	velval	CiA 408	DriveVelocityControl, ActualValue / ActualValue	INT32	ro	-	INT32	-
0x6520	1	Y	vellimupp	CiA 408	DriveVelocityControl, VelocityUpperDemandLimit / VelocityUpperDemandLimit	INT32	rw	Y	INT32	16384
0x6521	1	Y	vellimlow	CiA 408	DriveVelocityControl, VelocityLowerDemandLimit / VelocityLowerDemandLimit	INT32	rw	Y	INT32	-16384
0x6550	1	Y	velctldvn	CiA 408	DriveVelocityControl, dV / dV	INT32	ro	-	INT32	-
0x6551	0	Y	velmontyp	CiA 408	DriveVelocityControl, ControlMonitoring_Type / ControlMonitoring_Type	INT8	rw	Y	0…1	-
0x6552	0	Y	velmontim	CiA 408	DriveVelocityControl, ControlMonitoring_Time / ControlMonitoring_Time	UINT16	rw	Y	UINT16	-
0x6554	0	Y	velmonupp	CiA 408	DriveVelocityControl, ControlMonitoring_UpperThreshold / ControlMonitoring_UpperThreshold	INT32	rw	Y	INT32	512
0x6555	0	Y	velmonlow	CiA 408	DriveVelocityControl, ControlMonitoring_LowerThreshold / ControlMonitoring_LowerThreshold	INT32	rw	Y	INT32	-512
0x6600	1	Y	posset	CiA 408	DrivePositionControl, Setpoint / Setpoint	INT32	rw	N	INT32	-
0x6600	2	N	posuni	CiA 408	DrivePositionControl, Setpoint / Position_Unit	UINT8	ro	-	UINT8	0
0x6600	3	N	posprf	CiA 408	DrivePositionControl, Setpoint / Position_Prefix	INT8	ro	-	INT8	0

Table 122: Object dictionary (part 38 of 39)

Index	Sub-index	PDO mapping	Short name	Object name, Block name, Parameter name	Specification	Data type	Access	Persistence	Value range	Default
0x6601	1	Y	posval[0]	DrivePositionControl, ActualValues ActualValue#1	CiA 408	INT32	ro	-	INT32	-
0x6601	2	Y	posval[1]	DrivePositionControl, ActualValues ActualValue#2	CiA 408	INT32	ro	-	INT32	-
0x6601	3	Y	posval[2]	DrivePositionControl, ActualValues ActualValue#3	CiA 408	INT32	ro	-	INT32	-
0x6601	4	Y	posval[3]	DrivePositionControl, ActualValues ActualValue#4	CiA 408	INT32	ro	-	INT32	-
0x6602	1	N	positfref[0]	DrivePositionControl, InterfaceReference InterfaceReference	CiA 408	UINT8	rw	Y	0...8	1
0x6602	2	N	positfref[1]	DrivePositionControl, InterfaceReference InterfaceReference	CiA 408	UINT8	rw	Y	0...8	0
0x6602	3	N	positfref[2]	DrivePositionControl, InterfaceReference InterfaceReference	CiA 408	UINT8	rw	Y	0...8	0
0x6602	4	N	positfref[3]	DrivePositionControl, InterfaceReference InterfaceReference	CiA 408	UINT8	rw	Y	0...8	0
0x6610	1	Y	posdem	DrivePositionControl_DemandValueGenerator, DemandValue DemandValue	CiA 408	INT32	ro	-	INT32	-
0x6610	2	N	posuni	DrivePositionControl_DemandValueGenerator, DemandValue Position_Unit	CiA 408	UINT8	ro	-	UINT8	0
0x6610	3	N	posprf	DrivePositionControl_DemandValueGenerator, DemandValue Position_Prefix	CiA 408	INT8	ro	-	INT8	0
0x6614	1	Y	possethld	DrivePositionControl_DemandValueGenerator_HoldSetpoint DemandValueGenerator_HoldSetpoint	CiA 408	INT32	rw	Y	INT32	-
0x6614	2	N	posuni	DrivePositionControl_DemandValueGenerator_HoldSetpoint Position_Unit	CiA 408	UINT8	ro	-	UINT8	0
0x6614	3	N	posprf	DrivePositionControl_DemandValueGenerator_HoldSetpoint Position_Prefix	CiA 408	INT8	ro	-	INT8	0
0x6650	1	Y	posctldvn	DrivePositionControl, dX dX	CiA 408	INT32	ro	-	INT32	-
0x6651	0	Y	posmontyp	DrivePositionControl, ControlMonitoring_Type ControlMonitoring_Type	CiA 408	INT8	rw	Y	0...1	-
0x6652	0	Y	posmontim	DrivePositionControl, ControlMonitoring_Time ControlMonitoring_Time	CiA 408	UINT16	rw	Y	UINT16	-
0x6654	0	Y	posmonupp	DrivePositionControl, ControlMonitoring_UpperThreshold ControlMonitoring_UpperThreshold	CiA 408	INT32	rw	Y	INT32	512
0x6655	0	Y	posmonlow	DrivePositionControl, ControlMonitoring_LowerThreshold ControlMonitoring_LowerThreshold	CiA 408	INT32	rw	Y	INT32	-512

Table 122: Object dictionary (part 39 of 39)

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

WHAT MOVES YOUR WORLD

MOOG