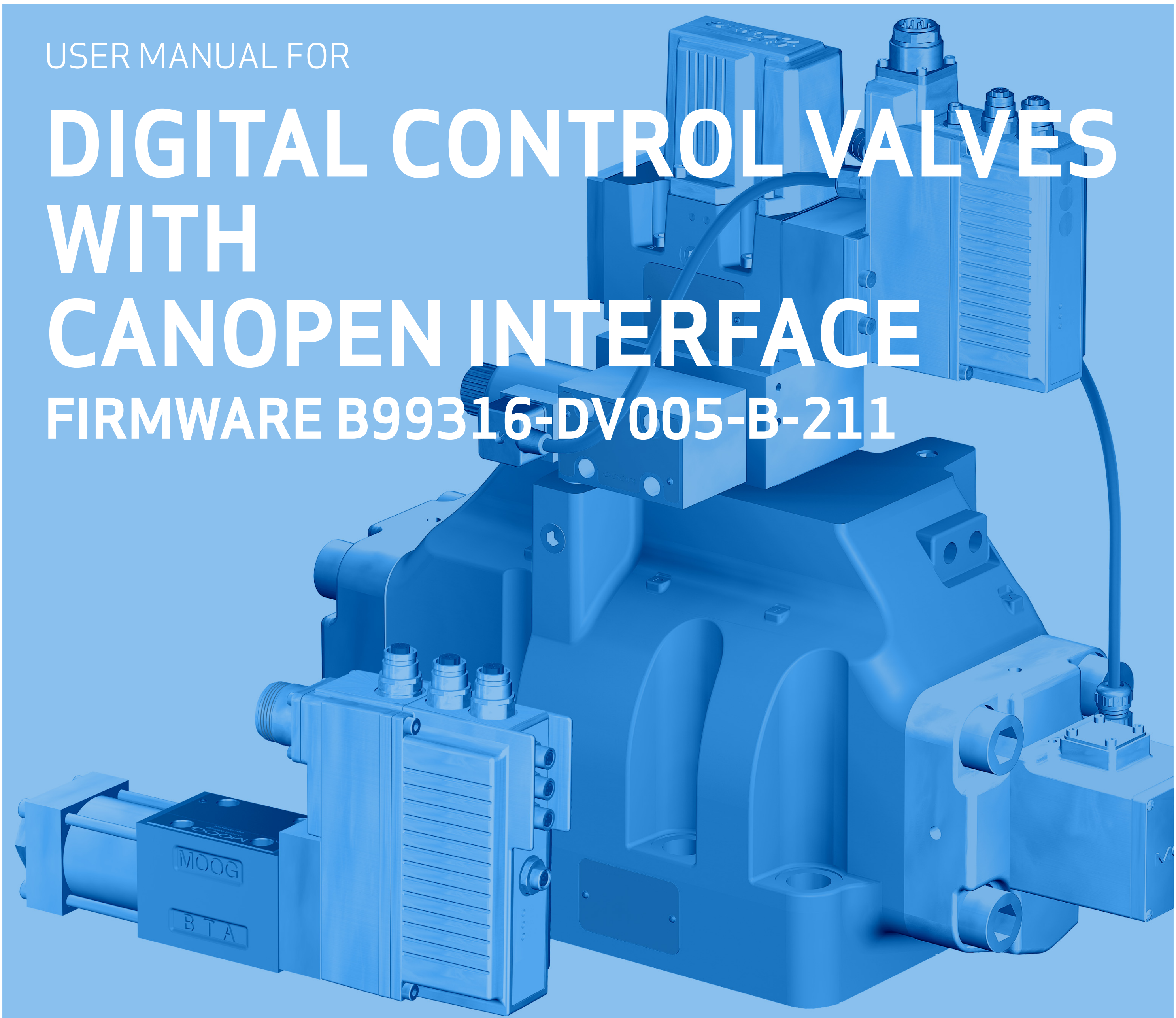
USER MANUAL FOR

DIGITAL CONTROL VALVES WITH CANOPEN INTERFACE

FIRMWARE B99316-DV005-B-211



Rev. B, 08.12.21

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

WHAT MOVES YOUR WORLD

MOOG

Copyright

Hanns-Klemm-Straße 28
71034 Boeblingen
Germany

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

Subject to change without notice.

Table of contents

List of tables

List of figures

1 General information

1.1 About this manual

This document describes the CANopen fieldbus interface of the Moog servo valves.

It describes the general structure of the CANopen fieldbus interface and the device specific CANopen profile of hydraulic valves.

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

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

> **!** This document is not a replacement for the CANopen standards as listed in the references.
> ⇨ Chapter "1.5 References", page 5

This manual was prepared with great care and the contents reflect the author's best knowledge. However, the possibility of errors remains and improvements are possible.

Please feel free to submit your comments regarding errors or improvements to Moog sales@moog.com.

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	
CDS45379-en	Application Instruction Getting Started - Digital Control Valves Series D63X, D94X, D767X
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 Servo Valves 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 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 fieldbus

The CANopen fieldbus 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 2.0

1.6 Definitions

1.6.1 Internal resolution (iR)

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

1.6.2 Volume flow direction

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

1.6.3 Servo valve position and stage names

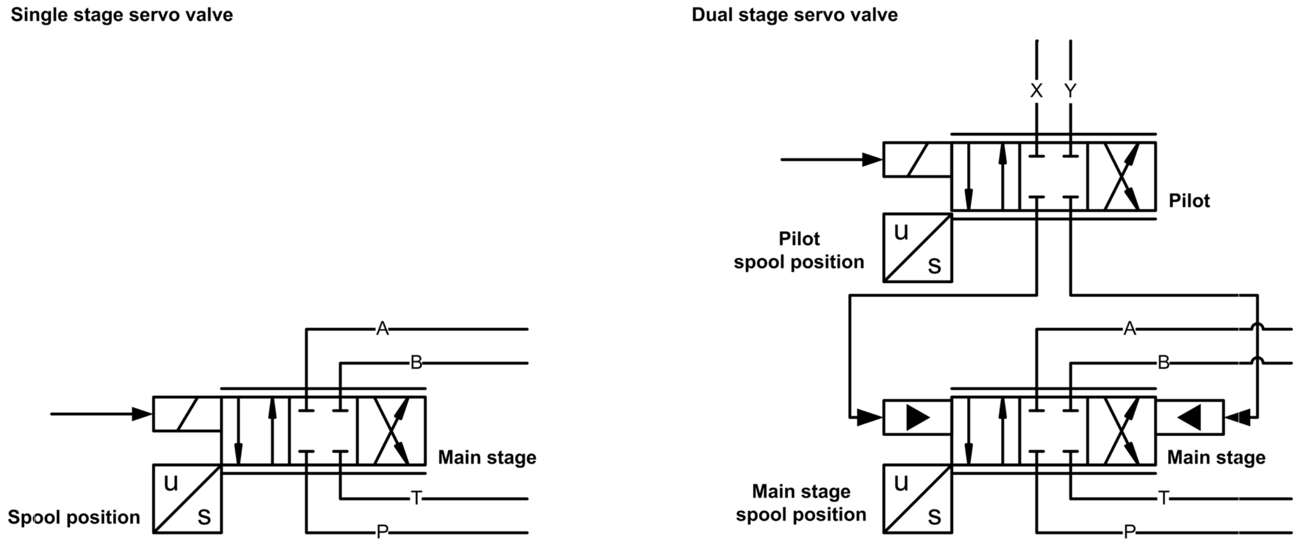


Figure 2: Servo valve position and stage names

1.7 Abbreviations

Table 1: Abbreviations (part 1 of 2)

Abbreviation	Explanation
A	Hydraulic port A of a valve
AC	Alternating Current
B	Hydraulic port B of a valve
CAN	Controller Area Network
CANopen	ISO/OSI Layer 7 protocol, specified by CAN in Automation (CiA)
CiA	CAN in Automation e.V.
COB-ID	Communication Object Identifier
DC	Direct Current
DCV	Moog Digital Control Valve
DSM	Device State Machine
EDS	Electronics Datasheet, containing a description of the CANopen object dictionary
EEPROM	Electrically erasable programmable read-only memory
inf	infinite number
iR	Internal resolution defined by CiA 408
ISO	International Organization for Standardization
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 Hydraulic port P of a valve
PD	Proportional derivative element
PDO	Process Data Object
PE	Protective earth / Electrical grounding

Table 1: Abbreviations (part 2 of 2)

Abbreviation	Explanation
PPT1	Proportional first order lag element
ro	Read only
rw	Read write
RPDO	Receive Process Data Object
SDO	Service Data Object
TR	State transition of the valve application state machine
TPDO	Transmit Process 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

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

Moog servo valves are equipped with a fieldbus for communication with the machine controller. In this case with the standardized open fieldbus CANopen.

The CAN interface is electrically isolated from the servo valve electronics. The CAN network cable must be terminated externally with appropriate bus terminating resistors of 120 Ω.

For operation in a CAN network, a unique node number and a baud rate must be set in the device during initial commissioning.

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 profile for the hydraulic valves CiA 408 is built on top of this basic profile.

2.2 Device profiles

The device profiles describe the application parameters and the functional behavior of the devices including the device class-specific state machines.

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 fieldbus in a standardized, fieldbus independent format across manufacturers. This profile is implemented in all Moog servo valves with fieldbus interface.

The CiA organization transformed this 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". Other device profiles are for example generic I/O-modules (CiA 401), drives and motion control (CiA 402).

2.3 CANopen 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 because such functionalities are not required in a real-time fieldbus system. The CANopen communication concept can be described similar to the ISO/OSI reference model.

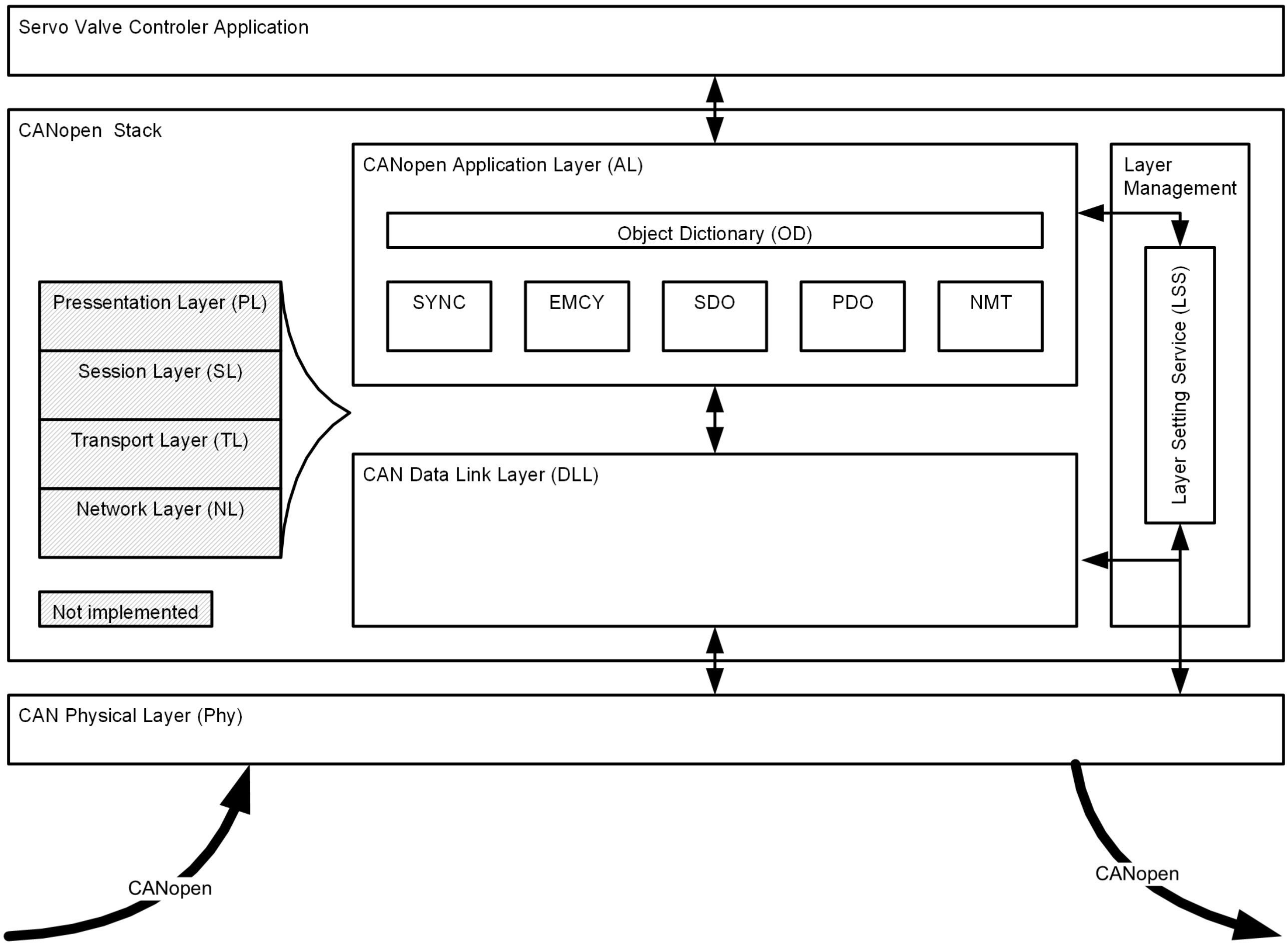


Figure 3: CANopen reference model

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.

Table 2: CANopen reference model

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)	

Table 2: CANopen reference model

Layer	Description	References CAN / CANopen
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)

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 "Object dictionary" you will find a form like the following one in which the attributes are explained in detail.
➪ Chapter "10 Object dictionary", page 290

Index	Sub-index	PDO-mapping	Short name	Specification	Block, object and parameter name	Data type	Access	Per-sistence	Value range	Default

Table 3: Fieldbus independent attributes

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. **CiA 301 data type** / **Short name** / **Description** INTEGERn / INTn / Signed integer with n bits UNSIGNEDn / UINTn / Unsigned integer with n bits REAL32 / REAL32 / Floating point with 32 bit according CiA 301 VISIBLE_STRING / STRING(n) / String of n ASCII characters DOMAIN / 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 parameter.
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 287
Specification	Related (fieldbus) standard defining the parameter. Possible entries: CiA 301: Parameters correspond to CiA 301. CiA 408: Parameters correspond to CiA 408. Moog DCV: Moog defined parameters for digital control servo 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.

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.

Table 4: Unit representation

Name of unit	International symbol	Notation index (hex)	Notation index (dec)
none	dimensionless or iR	0x00	0
meter	m	0x01	1
second	s	0x03	3
hertz	Hz	0x20	32
liter	l or L	0x44	68
minute (time)	min	0x47	71
hour	h	0x48	72
day	d	0x49	73
year	a	0x4A	74
bar	bar	0x4E	78
meter per square second	m/s^2	0x55	85

Table 5: Prefix representation

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

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.

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

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 used for the fieldbus CAN. Note: For configuring the local CAN interface, the objects 0x5000 and following are used.	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	

2.6 Electronic data sheet (EDS)

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. Model-specific EDS files are provided by Moog and can be downloaded from the Moog website http://www.moog.com/industrial.

2.7 CANopen communication protocols

CANopen communication protocols are classified as follows:

- Real-time data are transferred with the process data object (PDO) protocol.
- Configuration parameters are transferred with the service data object (SDO) protocol.
- Special tasks are realized using 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.

Table 7: CANopen communication objects (part 1 of 2)

Protocol	COB-ID	Description	SDO objects used for protocol configuration	Reference	Chapter
NMT	0x000	Network Management (NMT) protocol (Broadcast)		CiA 301	➪ Chapter "2.10 Network management (NMT) state machine", page 17
NMT	0x000+Node-ID	Network Management (NMT) protocol		CiA 301	
SYNC	0x080	Synchronization (Broadcast) protocol	0x1005, 0x1006, 0x1007, 0x1019	CiA 301	➪ Chapter "2.18 Synchronization (SYNC)", page 38
EMCY	0x080+Node-ID	Emergency protocol	0x1014	CiA 301	➪ Chapter "2.19 Emergency (EMCY)", page 40
TPDO	0x180+Node-ID	1st Transmit PDO protocol	0x1800, 0x1A00	CiA 301	➪ Chapter "2.13 Process data object (PDO)", page 23 ➪ Chapter "2.15 Transmit process data object (TPDO)", page 30
RPDO	0x200+Node-ID	1st Receive PDO protocol	0x1400, 0x1600	CiA 301	
TPDO	0x280+Node-ID	2nd Transmit PDO protocol	0x1801, 0x1A01	CiA 301	
RPDO	0x300+Node-ID	2nd Receive PDO protocol	0x1401, 0x1601	CiA 301	
TPDO	0x380+Node-ID	3rd Transmit PDO protocol	0x1802, 0x1A02	CiA 301	
RPDO	0x400+Node-ID	3rd Receive PDO protocol	0x1402, 0x1602	CiA 301	
TPDO	0x480+Node-ID	4th Transmit PDO protocol	0x1803, 0x1A03	CiA 301	
RPDO	0x500+Node-ID	4th Receive PDO protocol	0x1403, 0x1603	CiA 301	

Table 7: CANopen communication objects (part 2 of 2)

Protocol	COB-ID	Description	SDO objects used for protocol configuration	Reference	Chapter
SDO	0x580+Node-ID	Transmit SDO protocol	0x1200	CiA 301	➪ Chapter "2.16 Service data object (SDO)", page 37
SDO	0x600+Node-ID	Receive SDO protocol	0x1200	CiA 301	
NMT	0x700+Node-ID	Network Management (NMT) error control protocol (Bootup, Node Guarding, Heartbeat)	0x100C, 0x100D, 0x1016, 0x1017	CiA 301	➪ Chapter "2.10 Network management (NMT) state machine", page 17 ➪ Chapter "2.12 Network management (NMT) node guarding", page 22 ➪ Chapter "2.11 Network management (NMT) heartbeat", page 21
LSS	0x7E4	Transmit Layer Setting Services (LSS) protocol	0x1018	CiA 305	➪ Chapter "2.8 Bit rate and Node-ID configuration using Layer Setting Services (LSS)", page 15
LSS	0x7E5	Receive Layer Setting Services (LSS) protocol			

2.8 Bit rate and Node-ID configuration using Layer Setting Services (LSS)

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:

Table 8: CANopen bit rates

<Bitrate>	
Bit rate	**Maximum cable length**
1000 kBit/s	< 25 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

➪ For details, see document CA63420 001 "User Manual Electrical Interfaces"

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 an LSS master.

To configure the Node-ID of the CANopen fieldbus interface, the LSS protocol must be received via the field-bus connection.

The implemented LSS service provides four methods to change the Node-ID or bit rate of the servo valve:

1. LSS configuration service
   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. LSS inquiry service
   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 (0x1018) is worldwide unique and will be programmed during production. Information about the identification object (0x1018) is available from the name plate (LSS) of the device or can be inquired by using SDO.
3. LSS identification services
   The third method corresponds to the second with the possibility to find devices that match a certain VendorID and product code in the identification object (0x1018). This allows newly installed bus devices with changed revision number and serial number to be found.
4. LSS fastscan service
   As the LSS identify remote slave service, the LSS fastscan service may be used to scan for unconfigured LSS slaves. The value range of the active node-ID is from 0x01 to 0x7F. In addition to this range, the value range for pending Node-ID as well as the persistent Node-ID includes the value 0xFF which indicates an invalid setting.

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 <IdentityObject> (0x1018) or LSS address (VendorId / ProductCode / RevisionNumber / SerialNumber) can be found on the name plate of the servo valve. For more details about the LSS service see CiA 305.

➪ Chapter "4.1.5 Object 0x1018: Identity object", page 45

Figure 4: Nameplate of the device with identification object address

2.9 Bit rate and Node-ID configuration using service data object (SDO)

In case a network master does not support the LSS services, the bit rate and the Node-ID in the servo valve fieldbus CAN interface 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 an NMT-reset communication are initiated. That makes sure that the CAN communication will not get interrupted by changing these basic settings.

➪ Chapter "2.10 Network management (NMT) state machine", page 17

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.

2.9.1 Object 0x100B: Actual module identifier (Node-ID)

This parameter shows the currently used Node-ID of the CANopen fieldbus.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x100B	0	ActualModuleIdentifier	UINT32	ro	-	1...127	None

2.9.2 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, an NMT-reset communication, or an LSS-command has to be initiated.

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

2.9.3 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 an NMT-reset communication has to be initiated.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3003	0	Bitrate	UINT32	rw	Y	10000… 1000000	500000

Value description

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

Table 9: CANopen bit rates

<Bitrate>		
Bit rate	**Maximum cable length**	**Parameter value**
1000 kBit/s	< 25 m	1000000
500 kBit/s	< 100 m	500000
250 kBit/s	< 250 m	250000
125 kBit/s	< 500 m	125000
100 kBit/s - not supported	-	-
50 kBit/s	< 1000 m	50000
20 kBit/s	< 2500 m	20000

2.10 Network management (NMT) state machine

The CANopen NMT state machine is used to control the network communication of the fieldbus. 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 CANopen network management (NMT) state machine must not be mistaken with the device state machine.

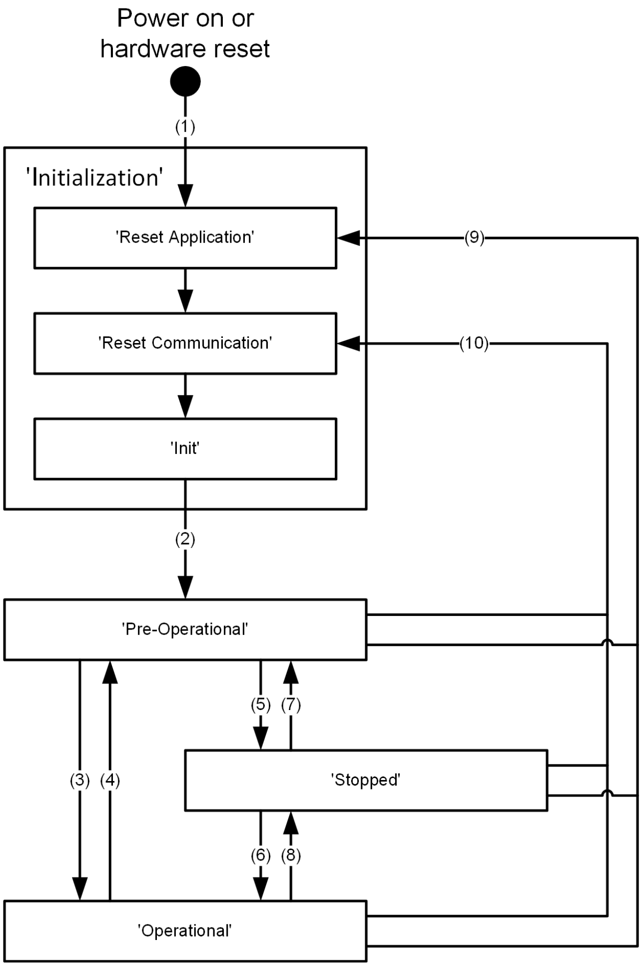


Figure 5: Network management (NMT) state machine

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. On transition to the 'Pre-Operational' state, a boot-up message is sent to signal successful booting to the master. 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 parameters. 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. For further information on the node control protocol see CiA 301.

NMT state transitions are caused by

- Reception of an NMT node control command from the CANopen bus
- Hardware reset, or
- Node control services locally initiated by application events.

Table 10: NMT state transitions

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".

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

Table 11: NMT states

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

2.10.1 Object 0x3004: Network management (NMT) state machine status

This parameter contains the network management (NMT) state machine status in bit coded form.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3004	0	NmtState	UINT16	ro	-	UINT16	None

Value description

Table 12: Object 0x3004: Network Management (NMT) state machine status

<NmtState>		
Status (hex)	Status (dec)	Status
0x0000	0	Invalid State
0x0001	1	State Power On or Hardware Reset
0x0002	2	Reset Application
0x0004	4	Reset Communication

Table 12: Object 0x3004: Network Management (NMT) state machine status

<NmtState>		
Status (hex)	**Status (dec)**	**Status**
0x0008	8	Init
0x0010	16	Pre-Operational
0x0020	32	Stopped
0x0040	64	Operational
0x8000	32768	Severe Error State

2.10.2 Object 0x3005: Network management (NMT) state machine control

This parameter is used to control the Network management (NMT) state machine. The possible status change depends on the current status.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3005	0	NmtSetState	UINT16	rw	N	1…130	None

Value description

⇨ Table 12, page 19

2.10.3 Object 0x1029: Network management (NMT) error behavior

If a serious CANopen device failure is detected in NMT state 'Operational', the CANopen device shall enter by default autonomously the NMT state 'Pre-operational'. The error behavior of the NMT can be changed via this parameter to NMT state 'Stopped' or remain in the current NMT state.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1029	0	ErrorBehavior	UINT8	rw	Y	UINIT8	0

Value description

<ErrorBehavior>	
0	Change to NMT state 'Pre-Operational' (only if currently in NMT state 'Operational')
1	No change of the NMT state
2	Change to NMT state 'Stopped'
3…256	No change of the NMT state

2.11 Network management (NMT) heartbeat

The Heartbeat Protocol consists of two separate services:

1. The monitoring of a periodically received heartbeat message, called consumer heartbeat. The Node-ID of the heartbeat transmitter and the time window until the next heartbeat message must arrive is set by the parameter <ConsumerHeartbeatTime> (0x1016). If the heartbeat transmitter does not respond within the specified time, error code 110 (0x6E) is triggered.
2. Periodically sending a heartbeat message, called producer heartbeat. For this purpose, the interval between two heartbeat messages can be set with the parameter <ProducerHeartbeatTime> (0x1017). The heartbeat protocol starts on the transition from the NMT state 'Initialization' to the NMT state 'Pre-Operational'. In this case the boot-up message is regarded as first heartbeat message. With the heartbeat message the servo valve NMT state is transmitted.

It is not allowed to use both error control mechanisms guarding protocol and heartbeat protocol on one NMT slave at the same time.

Even though both Heartbeat and Guarding are disabled by default, it is recommended to use error control mechanisms.

2.11.1 Object 0x1016: Consumer heartbeat time

The parameter <ConsumerHeartbeatTime> (0x1016) defines the expected heartbeat cycle time. It additionally defines the node-ID of the heartbeat producer to be monitored.

The consumer heartbeat time should be higher than the corresponding producer heartbeat time.

To enable the heartbeat message, a time greater zero must be set and a valid node-ID (between 1…127) must be configured. If the heartbeat time is 0 or the node-ID is 0 or greater than 127, the corresponding object entry shall be not used. The heartbeat time shall be given in multiples of 1 ms.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1016	0	ConsumerHeartbeatTime	UINT32	rw	Y	UINT32	0

Value description

<ConsumerHeartbeatTime>			
Bit	**31…24**	**23…16**	**15…0**
Description	Reserved	Node-ID	Heartbeat time in ms

2.11.2 Object 0x1017: Producer heartbeat time

The parameter <ProducerHeartbeatTime> (0x1017) specifies the time in ms at which the servo valve sends a heartbeat message. The value 0 disables the producer heartbeat.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1017	0	ProducerHeartbeatTime	UINT16	rw	Y	UINT16	0

2.12 Network management (NMT) node guarding

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 transmit 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 (0x6E) 'CAN life guard error or heartbeat error' is triggered is calculated as follows:
Monitoring time = <GuardTime> (0x100C) · <LifeTimeFactor> (0x100D)

!	It is not allowed to use both error control mechanisms guarding protocol and heartbeat protocol on one NMT slave at the same time.

If the network master supports the Heartbeat Protocol, this should be preferred to Node Guarding, as it allows better monitoring. The guarding protocol is still supported for compatibility reasons.

2.12.1 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

2.12.2 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	1

2.13 Process data object (PDO)

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

Four RPDOs and four TPDOs are implemented:

- Receive process data object (RPDO), ⇨ Chapter "2.14 Receive process data object (RPDO)", page 24
- Transmit process data object (TPDO), ⇨ Chapter "2.15 Transmit process data object (TPDO)", page 30

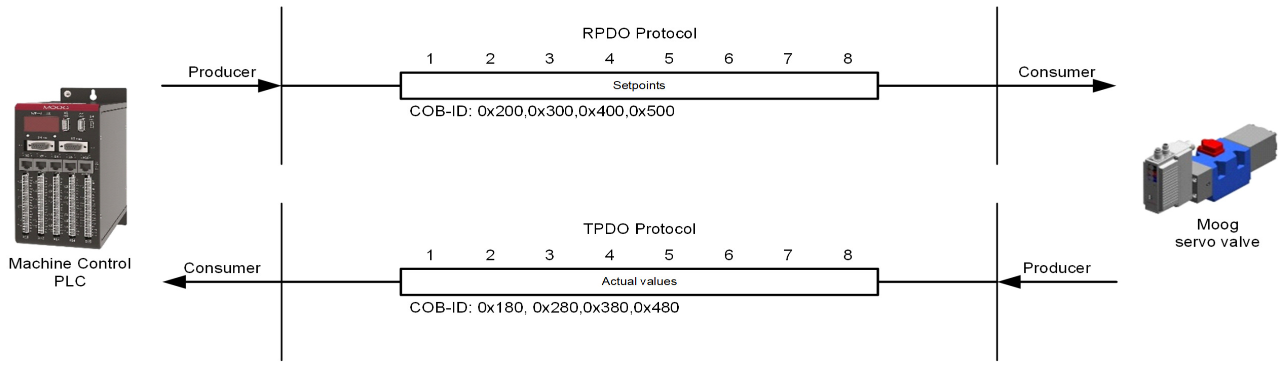


Figure 6: PDO Real Time Data Transmission

2.14 Receive process data object (RPDO)

With the receive process data object (RPDO) mapping most object dictionary entries can be mapped to RPDO. To enable receive process data object (RPDO) transmission, the local application parameters must be mapped to the RPDO and a transmission type must be selected for each RPDO channel. Four RPDO channels are available.

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 bits can be mapped within one RPDO. An arbitrary combination of different data types is possible if the sum of the mapped RPDO data is less or equal 8 bytes.

Two different kind of parameters are used to configure the transmission.

- Parameters 0x1400…0x1402 are used to configure the type of transmission.
- Parameters 0x1600…0x1603 are used to configure which application data is transmitted.

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

- Device state machine (DSM) Control Word <ControlWord> (0x6040) (sub-index 0x00)
  ➪ Chapter "5.1 Device state machine (DSM)", page 50
- Q setpoint <QSetpoint> (0x6300) (sub-index 0x01)
  ➪ Chapter "6.1.3 Q setpoint path", page 65
- Pressure setpoint <PrsSetpoint> (0x6380) (sub-index 0x01)
  ➪ Chapter "6.1.4 p setpoint path", page 67

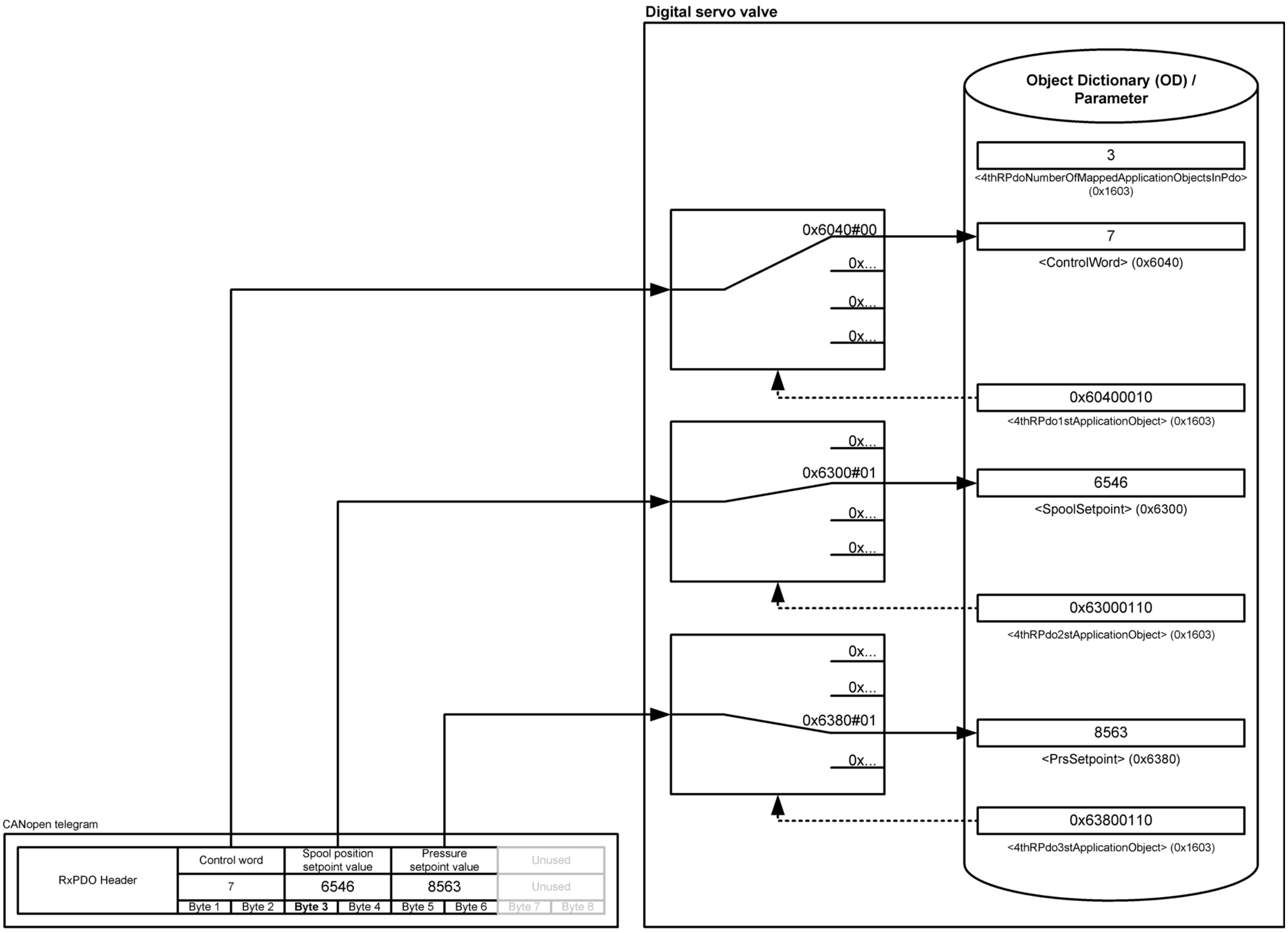


Figure 7: Receive process data object (RPDO) mapping

For example, the parameter <4thRPdoNumberOfMappedApplicationObjectsInPdo> (0x1603) defines the number of mapped values for the fourth RPDO. The second sub-index is a reference to the <ControlWord> (0x6040) (sub-index 0x00). The references to the parameters <QSetpoint> (0x6300) (sub index 0x01) and <PrsSetpoint> (0x6380) (sub-index 0x01) are defined in the same manner.

For detailed information how the PDO protocol is working, please take a look into the actual CiA 301 CANopen application layer and communication profile.

2.14.1 Object 0x1400: 1st RPDO configuration

First RPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1400	1	1stRPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0200+127
0x1400	2	1stRPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x1400	5	1stRPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description

Table 13: Object 0x1400: 1st RPDO configuration

Parameter	**Description**
<1stRPdoCobIdUsedByPdo>	COB-ID of the 1st RPDO.
<1stRPdoTransmissionType>	Different trigger can be configured: ⇨ Table 15, page 25
<1stRPdoEventTimer>	This parameter defines the timeout in milliseconds for the RPDO timeout monitoring. The event timer is used to recognize the expiration of the RPDO. 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 14: Possible values of parameter <1stRPdoCobIdUsedByPdo> (0x1400, sub-index 1)

<1stRPdoCobIdUsedByPdo>				
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 15: Possible values of parameter <1stRPdoTransmissionType> (0x1400, sub-index 2)

<1stRPdoTransmissionType>	
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 <1stRPdoTransmissionType> (0x1400) SYNC telegram.
241…253	Reserved
254…255	PDO will be processed immediately after reception.

2.14.2 Object 0x1401: 2nd RPDO configuration

Second RPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1401	1	2ndRPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0300+127
0x1401	2	2ndRPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x1401	5	2ndRPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
➪ Table 13, page 25

2.14.3 Object 0x1402: 3rd RPDO configuration

Third RPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1402	1	3rdRPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0400+127
0x1402	2	3rdRPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x1402	5	3rdRPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
➪ Table 13, page 25

2.14.4 Object 0x1403: 4th RPDO configuration

Fourth RPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1403	1	4thRPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0500+127
0x1403	2	4thRPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x1403	5	4thRPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
➪ Table 13, page 25

2.14.5 Object 0x1600: 1st RPDO mapping

With the parameter <1stRPdoNumberOfMappedApplicParaInPdo> (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	Persistence	Value range	Default
0x1600	0	1stRPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	1
0x1600	1	1stRPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60400010
0x1600	2	1stRPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0
0x1600	3	1stRPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x1600	4	1stRPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x1600	5	1stRPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x1600	6	1stRPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x1600	7	1stRPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x1600	8	1stRPdo8thApplicationObject	UINT32	rw	Y	UINT32	0

Value description

Table 16: Object 0x1600: 1st RPDO mapping

Parameter	Description
<1stRPdoNumberOfMappedApplicationObjectsInPdo>	Numbers of mapped application parameters
<1stRPdo1stApplicationObject>	Mapping of 1st application parameter
<1stRPdo2ndApplicationObject>	Mapping of 2nd application parameter
<1stRPdo3rdApplicationObject>	Mapping of 3rd application parameter
<1stRPdo4thApplicationObject>	Mapping of 4th application parameter
<1stRPdo5thApplicationObject>	Mapping of 5th application parameter
<1stRPdo6thApplicationObject>	Mapping of 6th application parameter
<1stRPdo7thApplicationObject>	Mapping of 7th application parameter
<1stRPdo8thApplicationObject>	Mapping of 8th application parameter

Table 17: Value description of mapping parameter
<1stRPdo1stApplicationObject>…<1stRPdo8thApplicationObject>

<1stRPdo1stApplicationObject>…<1stRPdo8thApplicationObject>				
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

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).

2.14.6 Object 0x1601: 2nd RPDO mapping

Second RPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1601	0	2ndRPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	2
0x1601	1	2ndRPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60400010
0x1601	2	2ndRPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63000110
0x1601	3	2ndRPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x1601	4	2ndRPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x1601	5	2ndRPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x1601	6	2ndRPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x1601	7	2ndRPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x1601	8	2ndRPdo8thApplicationObject	UINT32	rw	Y	UINT32	0

The mapping of this PDO is predefined as default mapping for a standard Q-control servo valve. It maps the ‹ControlWord› (0x6040) and the ‹QSetpoint› (0x6300).

Value description

⇨ Table 16, page 27

2.14.7 Object 0x1602: 3rd RPDO mapping

Third RPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1602	0	3rdRPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	2
0x1602	1	3rdRPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60400010
0x1602	2	3rdRPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63800110
0x1602	3	3rdRPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x1602	4	3rdRPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x1602	5	3rdRPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x1602	6	3rdRPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x1602	7	3rdRPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x1602	8	3rdRPdo8thApplicationObject	UINT32	rw	Y	UINT32	0

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

Value description

⇨ Table 16, page 27

2.14.8 Object 0x1603: 4th RPDO mapping

Fourth RPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1603	0	4thRPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	3
0x1603	1	4thRPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60400010
0x1603	2	4thRPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63000110
0x1603	3	4thRPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0x63800110
0x1603	4	4thRPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x1603	5	4thRPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x1603	6	4thRPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x1603	7	4thRPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x1603	8	4thRPdo8thApplicationObject	UINT32	rw	Y	UINT32	0

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

Value description
➩ Table 16, page 27

2.14.9 Object 0x3012: RPDO counter

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

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x3012	1	1stRPdoCounter	UINT32	rw	N	UINT32	0
0x3012	2	2ndRPdoCounter	UINT32	rw	N	UINT32	0
0x3012	3	3rdRPdoCounter	UINT32	rw	N	UINT32	0
0x3012	4	4thRPdoCounter	UINT32	rw	N	UINT32	0

2.15 Transmit process data object (TPDO)

The transmit process data object (TPDO) protocol must be configured and the transmit values must be mapped to the local parameters. Four transmit TPDO 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.

With the transmit process data object (TPDO) mapping the most object dictionary entries can be mapped to a TPDO. 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 TPDO. An arbitrary combination of different data types is possible, if the sum of the mapped TPDO 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 TPDO (default mapping):

- Device state machine (DSM) Status Word <StatusWord> (0x6041) (sub-index 0x00)
  ⇨ Chapter "5.1 Device state machine (DSM)", page 50
- Spool position actual value <SpoolActualValue> (0x6301) (sub-index 0x01)
  ⇨ Chapter "7.3 Spool position controller", page 163
- Pressure actual value <PrsActualValue> (0x6381) (sub-index 0x01)
  ⇨ Chapter "7.5.11 Actual pressure value filter", page 196

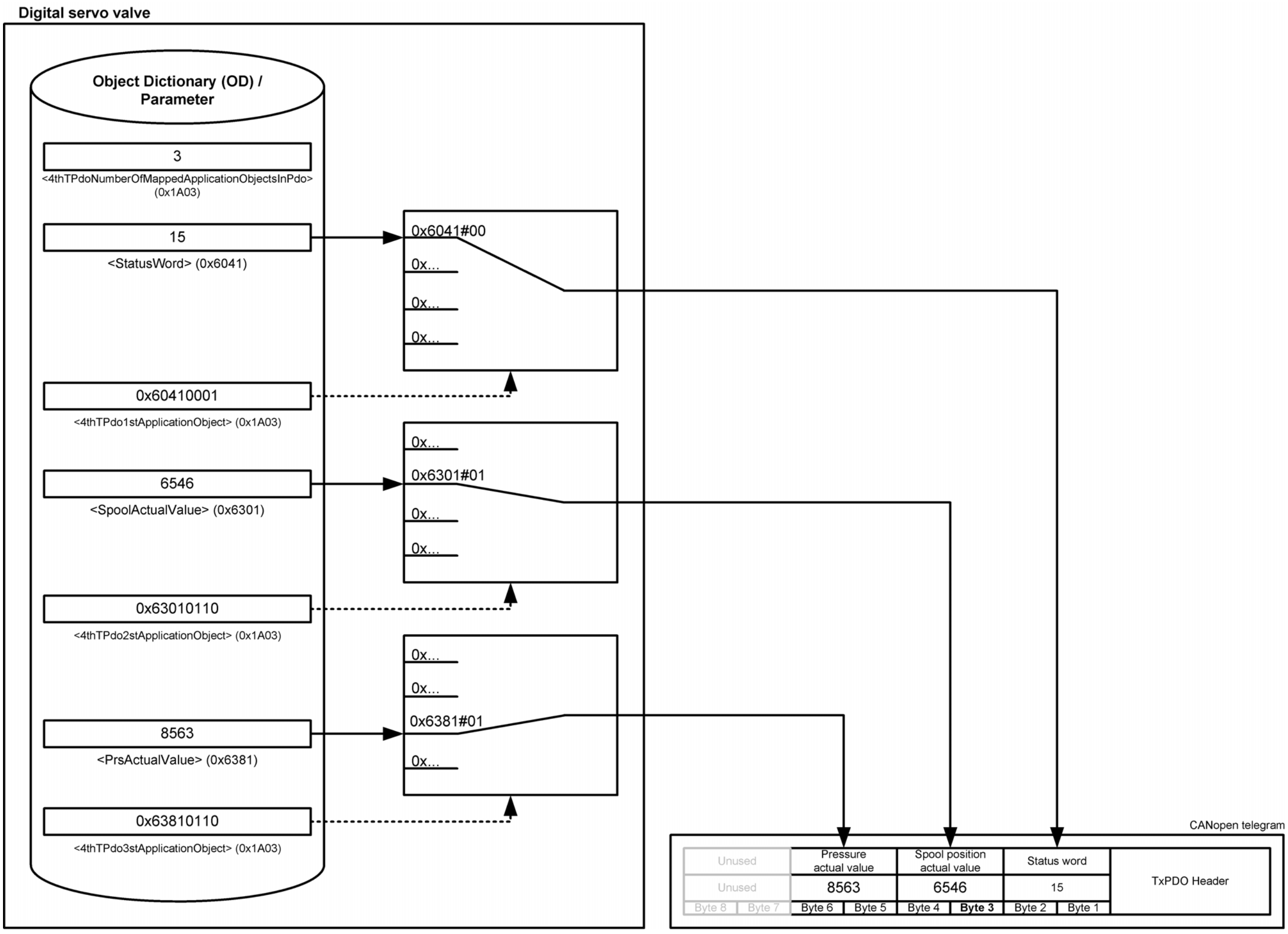


Figure 8: Transmit process data object (TPDO) mapping

With the parameter <4thTPdoNumberOfMappedApplicationObjectsInPdo> 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 <SpoolActualValue> (0x6301) (sub-index 0x01) and <PrsActualValue> (0x6381) (sub-index 0x01) are done in the same manner.

For detailed information how the PDO protocol is working, please take a look into the actual CiA 301 CANopen application layer and communication profile.

2.15.1 Object 0x1800: 1st TPDO configuration

First TPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1800	1	1stTPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0180+127
0x1800	2	1stTPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x1800	3	1stTPdoInhibitTime	UINT16	rw	Y	UINT16	0
0x1800	5	1stTPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description

Table 18: Object 0x1800: 1st TPDO configuration

Parameter	Description
‹1stTPdoCobIdUsedByPdo›	Defines the COB-ID for the 1st TPDO.
‹1stTPdoTransmissionType›	Defines the transmission behavior for the 1st TPDO.
‹1stTPdoInhibitTime›	Not implemented.
‹1stTPdoEventTimer›	This parameter defines the event time in milliseconds. This time defines the cycle time of the TPDO timer event. If the event time elapsed, a trigger to transmit a TPDO is initiated and the event timer will be restarted.

Table 19: Possible values of parameter ‹1stTPdoCobIdUsedByPdo› (0x1800, sub-index 1)

‹1stTPdoCobIdUsedByPdo›				
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 20: Possible values of parameter ‹1stTPdoTransmissionType› (0x1800, sub-index 2)

‹1stTPdoTransmissionType›	
0	The process data for the TPDO will be updated and sent immediately after the next incoming SYNC telegram.
1…240	The process data for the TPDO will be updated and sent immediately every ‹1stTPdoTransmissionType› SYNC telegrams.
241…253	Reserved
254…255	The TPDO will be sent after the event time is elapses (when the event time is nonzero).

To initiate a single request only, the ‹1stTPdoEventTimer› (0x1800)…‹4thTPdoEventTimer› (0x1803) must be set to 0. The transmission then can be triggered by the parameter ‹TPdoTrigger› (0x3011).

➪ Chapter "2.15.9 Object 0x3011: TPDO trigger", page 36

2.15.2 Object 0x1801: 2nd TPDO configuration

Second TPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1801	1	2ndTPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0280+127
0x1801	2	2ndTPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x1801	3	2ndTPdoInhibitTime	UINT16	rw	Y	UINT16	0
0x1801	5	2ndTPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
⇨ Table 18, page 32

2.15.3 Object 0x1802: 3rd TPDO configuration

Third TPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1802	1	3rdTPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0380+127
0x1802	2	3rdTPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x1802	3	3rdTPdoInhibitTime	UINT16	rw	Y	UINT16	0
0x1802	5	3rdTPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
⇨ Table 18, page 32

2.15.4 Object 0x1803: 4th TPDO configuration

Fourth TPDO channel configuration.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1803	1	4thTPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0480+127
0x1803	2	4thTPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x1803	3	4thTPdoInhibitTime	UINT16	rw	Y	UINT16	0
0x1803	5	4thTPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
⇨ Table 18, page 32

2.15.5 Object 0x1A00: 1st TPDO mapping

With the parameter <1stTPdoNumberOfMappedApplicationObjectsInPdo> (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	Per-sis-tence	Value range	Default
0x1A00	0	1stTPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	1
0x1A00	1	1stTPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60410010
0x1A00	2	1stTPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0
0x1A00	3	1stTPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x1A00	4	1stTPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A00	5	1stTPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A00	6	1stTPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A00	7	1stTPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A00	8	1stTPdo8thApplicationObject	UINT32	rw	Y	UINT32	0

Value description

Table 21: Object 0x1A00: 1st TPDO mapping

Parameter	Description
<1stTPdoNumberOfMappedApplicationObjectsInPdo>	Numbers of mapped application parameters
<1stTPdo1stApplicationObject>	1st application parameter
<1stTPdo2ndApplicationObject>	2nd application parameter
<1stTPdo3rdApplicationObject>	3rd application parameter
<1stTPdo4thApplicationObject>	4th application parameter
<1stTPdo5thApplicationObject>	5th application parameter
<1stTPdo6thApplicationObject>	6th application parameter
<1stTPdo7thApplicationObject>	7th application parameter
<1stTPdo8thApplicationObject>	8th application parameter

Table 22: Possible values of parameter <1stTPdo1stApplicationObject>…<1stTPdo8thApplicationObject>

<1stTPdo1stApplicationObject>…<1stTPdo8thApplicationObject>				
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

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).

2.15.6 Object 0x1A01: 2nd TPDO mapping

Second TPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1A01	0	2ndTPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	2
0x1A01	1	2ndTPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60410010
0x1A01	2	2ndTPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63010110
0x1A01	3	2ndTPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x1A01	4	2ndTPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A01	5	2ndTPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A01	6	2ndTPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A01	7	2ndTPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A01	8	2ndTPdo8thApplicationObject	UINT32	rw	Y	UINT32	0

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

Value description
⇨ Table 21, page 34

2.15.7 Object 0x1A02: 3rd TPDO mapping

Third TPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1A02	0	3rdTPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	2
0x1A02	1	3rdTPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60410010
0x1A02	2	3rdTPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63810110
0x1A02	3	3rdTPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x1A02	4	3rdTPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A02	5	3rdTPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A02	6	3rdTPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A02	7	3rdTPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A02	8	3rdTPdo8thApplicationObject	UINT32	rw	Y	UINT32	0

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

Value description
⇨ Table 21, page 34

2.15.8 Object 0x1A03: 4th TPDO mapping

Fourth TPDO mapping.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x1A03	0	4thTPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	2
0x1A03	1	4thTPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60410010
0x1A03	2	4thTPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63010110
0x1A03	3	4thTPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0x63810110
0x1A03	4	4thTPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A03	5	4thTPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A03	6	4thTPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A03	7	4thTPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A03	8	4thTPdo8thApplicationObject	UINT32	rw	Y	UINT32	0

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

Value description

⇨ Table 21, page 34

2.15.9 Object 0x3011: TPDO trigger

Writing this parameter triggers a single TPDO. This can be used to transmit a PDO on request only. To trigger one of the four TPDOs, 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	Per-sis-tence	Value range	Default
0x3011	0	TPdoTrigger	UINT8	rw	N	1…4	None

2.16 Service data object (SDO)

The service data objects are used to configure the settings for the communication and to set or read (upload) application parameters. The SDO data object is connected to the application data via the CANopen object directory. The SDOs are transmitted non real-time with low priority.

If an SDO request cannot be processed, e.g., the value is too big or the client has not the access rights to write (download) a parameter, an SDO abort message with an abort code is sent back from the server to the client.
➪ Chapter "8.3 Abort SDO Transfer Protocol", page 285

For detailed information how the SDO protocol is working, please look into the actual CiA 301 CANopen application layer and communication profile.

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

Figure 9: Write service data object request

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

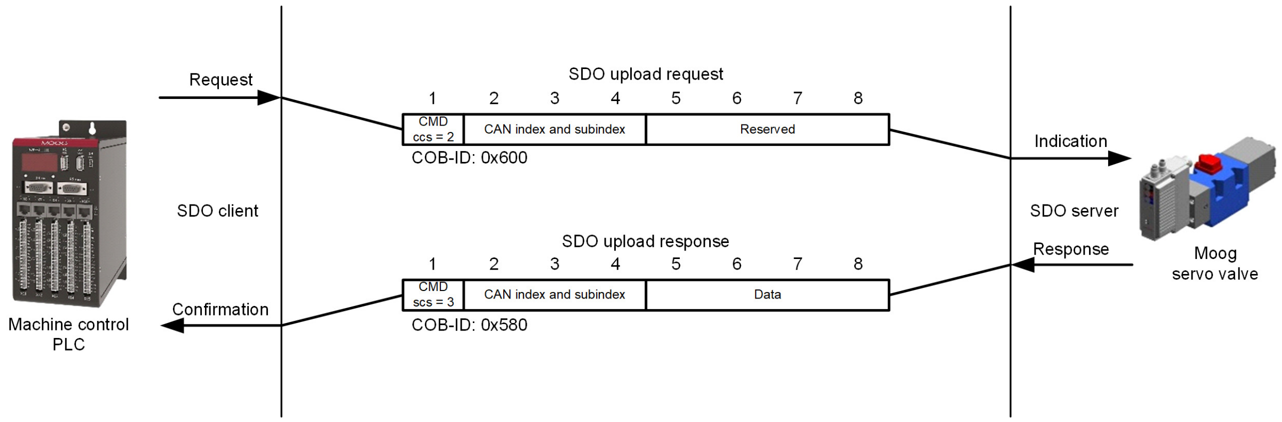


Figure 10: Read service data object request

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

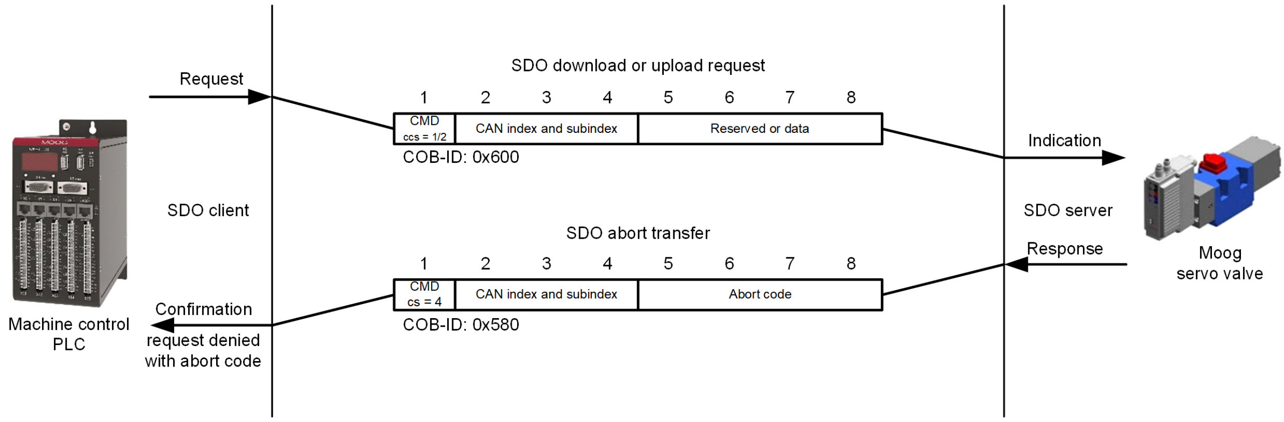


Figure 11: Typical SDO abort message

2.16.1 Object 0x1200: SDO client/server parameter

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

Value description

Table 23: Object 0x1200: SDO client/server parameter

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

2.17 Service data object (SDO) gateway

If an external CAN device is connected to the local CAN interface of the servo valve, the SDO parameters of the external CAN device can be read (upload) and written (download) by the fieldbus master via the SDO gateway.
Information on the local CAN gateway: ➪ Chapter "6.7.16 Service data object (SDO) gateway", page 136

2.18 Synchronization (SYNC)

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 "2.14.1 Object 0x1400: 1st RPDO configuration", page 25
➪ Chapter "2.15.1 Object 0x1800: 1st TPDO configuration", page 32
➪ Chapter "2.15.9 Object 0x3011: TPDO trigger", page 36

2.18.1 Object 0x1005: SYNC protocol COB-ID configuration

This parameter 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+127

Value description

Table 24: Possible values of parameter <CobIdSyncMessage> (0x1005)

<CobIdSyncMessage>				
Bit	**31**	**30**	**29…11**	**10…0**
Description	Reserved	0: SYNC consumer 1: SYNC producer	Reserved	11 bit COB-ID

2.18.2 Object 0x1006: Communication cycle period

The value shall be given in multiple of µs. If the value is set to 0, the transmission of SYNC messages is disabled. To activate the sync signal the COB-ID SYNC message, object 0x1005 must be configured accordingly.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1006	0	CommunicationCyclePeriod	UINT32	rw	Y	UINT32	0

2.18.3 Object 0x1007: Synchronous window length

This object represents the configured length of the time window for synchronous PDOs. If the synchronous window length expires, all synchronous TPDOs may be discarded and an EMCY message will be transmitted.

The value is given in multiple of µs. If the value is set to 0, the synchronous window shall be disabled.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1007	0	SynchronousWindowLength	UINT32	rw	Y	UINT32	0

2.18.4 Object 0x1015: Inhibit time emergency message

This object indicates the configured inhibit time for the EMCY message. The value is given in multiples of 100 µs. The value 0 shall disable the inhibit time.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1015	0	InhibitTimeEmergencyMessage	UINT16	rw	Y	UINT16	0

2.18.5 Object 0x1019: SYNC protocol counter overflow value

This object shall indicate the configured highest value the synchronous counter supports. If the value is greater than 1, the SYNC message shall have a data length of 1 byte. An EMCY message (error code 0x8240) may be transmitted by a SYNC consumer in the case the configured data length of the SYNC message does not meet the data length of a received SYNC message. The value used shall be the least common multiple of all the TPDO transmission types (1 < n <= 240) used. This ensures that periodic SYNC events always happen in the SYNC cycles with the same counter value.

CANopen							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1019	0	SynchronousCounterOverflowValue	UINT8	rw	Y	UINT8	0

Value description

<SynchronousCounterOverflowValue>	**Description**
0	The SYNC message shall be transmitted as a CAN message of data length 0.
1	Reserved
2…240	The SYNC message shall be transmitted as a CAN message of data length 1. The first data byte contains the counter.
241…255	Reserved

2.19 Emergency (EMCY)

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 specified in the CANopen communication profiles CiA 301 and CiA 408. They are described in the chapter Diagnostics.

➪ Chapter "8 Diagnostics", page 261

2.19.1 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

Table 25: Possible values of parameter <CobIdEmergencyMessage> (0x1014)

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

3 Device structure

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

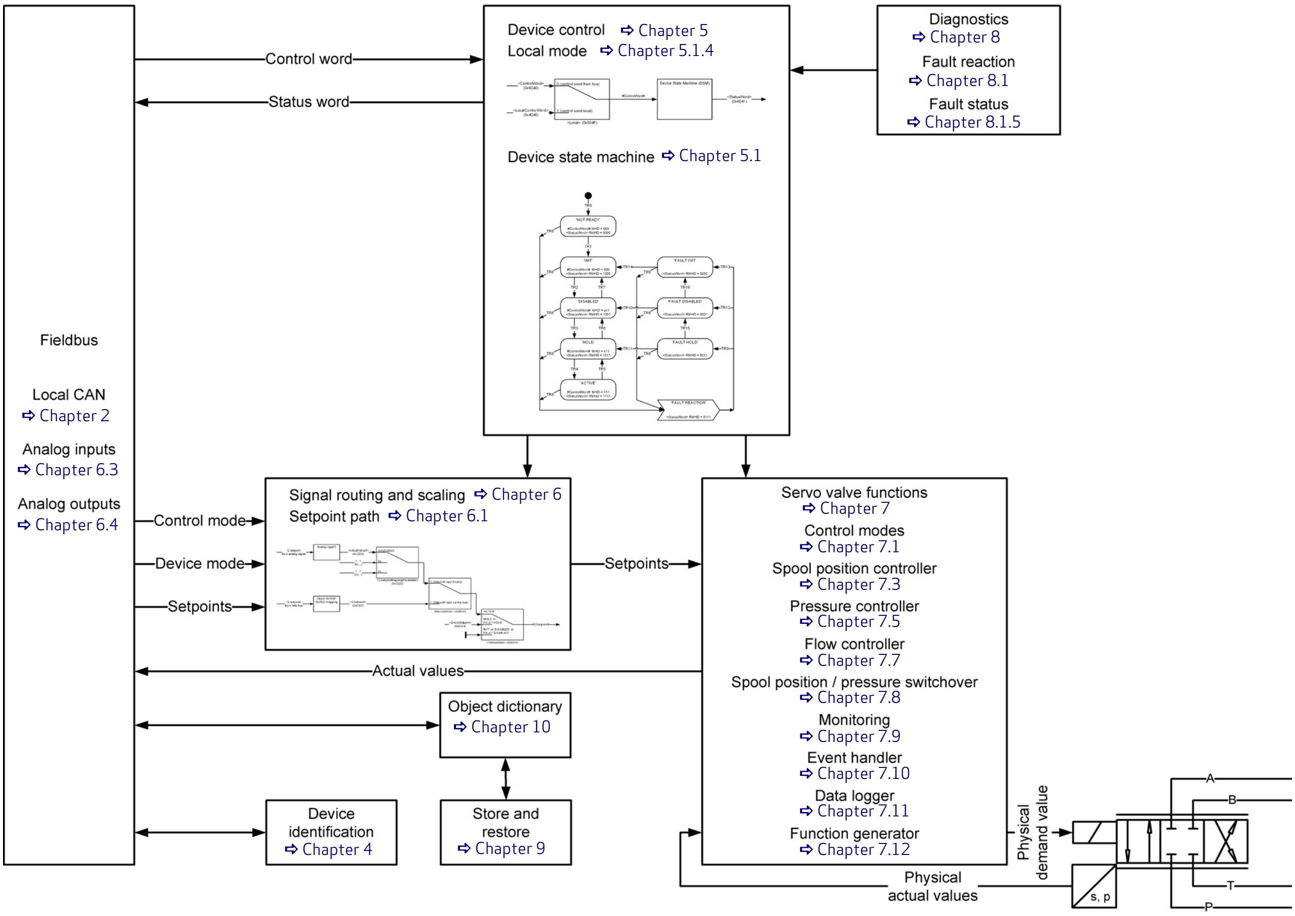


Figure 12: Device structure

The most important functional units are shortly described below:

- Device identification
  For the device identification (model number, serial number, device capability ...) a special set of parameters is defined.
- 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.
- Signal routing and scaling
  The signal routing and scaling function block prepares the analog setpoint and actual values signals and transmits them to the valve controller. This signal routing depends among other things on the <DeviceMode> (0x6042) and the device state machine state.
- Servo valve functions
  All signals in the servo valve are processed by the internal servo valve function block. Depending on the valve, the controller can control the spool position (Q or flow) or pressure (p) 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.

- 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.
- Storing / restoring parameters
  Save and load several servo valve parameters to the internal non-volatile memory of the device.
- Object dictionary
  The object dictionary holds all servo valve parameters. These parameters can be read or written using the fieldbus or the local CAN interface and the Moog Valve and Pump Configuration Software.

3.1 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.

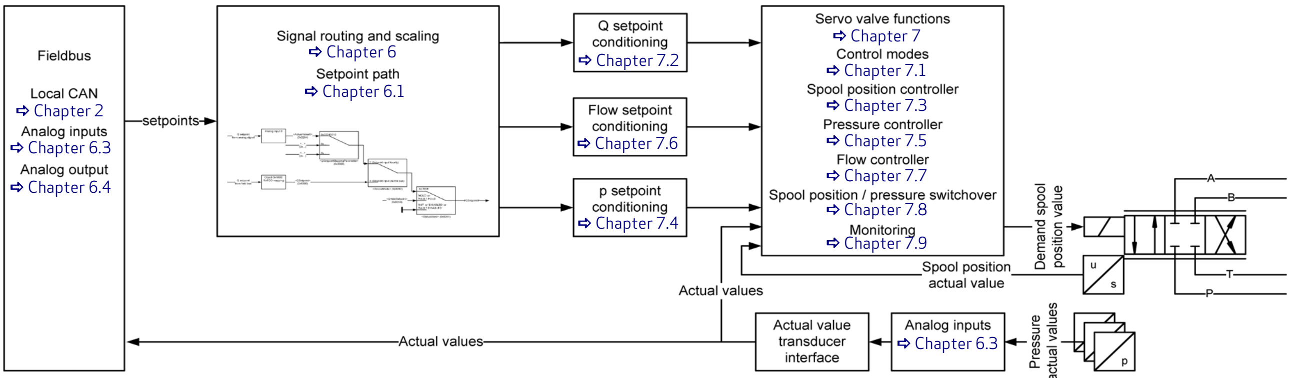


Figure 13: Device controller structure

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

The lower 16 bits of this parameter indicate the code of the underlying device profile. The default value 408 of device profile specifies the device profile CiA 408.
The upper 16 bits of this parameter specify the available control modes according CiA 408.

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

Value description

Table 26: Possible values of parameter <DeviceType> (0x1000) (part 1 of 2)

Bit	Value	Description	Control mode	Drive
0…15	0…15	CiA device profile number 408 This number stands for the Device Profile CiA 408 "Device profile for fluid power technology proportional valve and hydrostatic transmissions".		
16	0	Open loop movement control not supported	6	x
	1	Open loop movement control supported		
17	0	Velocity control axis not supported	7	x
	1	Velocity control axis supported		
18	0	Pressure (force) control not supported	8	x
	1	Pressure (force) control supported		
19	0	Position control axis not supported	9	x
	1	Position control axis supported		
20	0	Positional dependent deceleration not supported	10	x
	1	Positional dependent deceleration supported		
21	0	Position control axis with pressure (force) override not supported	11	x
	1	Position control axis with pressure (force) override supported		
22	0	Velocity control axis with pressure (force) override not supported	12	
	1	Velocity control axis with pressure (force) override supported		
23	0	Spool position control not supported	2	
	1	Spool position control supported		
24	0	Pressure (force) control axis not supported	4	
	1	Pressure (force) control axis supported		
25	0	pQ control not supported	5	
	1	pQ control supported		
26	0	Flow control not supported	13	
	1	Flow control supported		
27	0	pFlow control not supported	14	
	1	pFlow control supported		

Table 26: Possible values of parameter <DeviceType> (0x1000) (part 2 of 2)

Bit	Value	Description	Control mode	Drive
28	0	pFlow control with power limitation not supported	15	
	1	pFlow control with power limitation supported		
29…30	00	Reserved		
	01	Valve (pump) with 16-bit parameter support; Drive not supported		
	10	Valve (pump) with 32-bit parameter support; Drive with 32-bit parameter support		
	11	Valve (pump) with 16-bit parameter support; Drive with 32-bit parameter support		
31	0	Reserved		

4.1.2 Object 0x1008: Manufacturer device name

This parameter indicates the name of the servo valve. It corresponds to the model number as printed on the nameplate.

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x1008	0	ManufacturerDeviceName	STRING(64)	ro	-	STRING(64)	""

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(64)	ro	-	STRING(64)	""

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(64)	ro	-	STRING(64)	Device specific value

4.1.5 Object 0x1018: Identity object

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	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 nameplate of the servo valve as shown in the following figure.

Figure 14: Nameplate of the device with identification object address

Value description

Table 27: Structure of the Identity object (0x1018)

Sub-index	Parameter	Meaning
1	<VendorId>	Unique vendor ID, 0x28 (40) 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 as on the nameplate. ⇨ Chapter "4.2.3 Object 0x6052: Device serial number", page 46

<VendorId>				
Description	Department (0x00)		Company (0x000028)	
Bit	31	24	23	0
	MSB			LSB

4.2 Objects defined by Device Profile Fluid Power

4.2.1 Object 0x6050: Device 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(128)	ro	-	STRING(128)	""

4.2.2 Object 0x6051: Device code number

The user can use this parameter to set a user defined code number for the actual device configuration.

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

4.2.3 Object 0x6052: Device 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(64)	ro	-	STRING(64)	""

4.2.4 Object 0x6053: Device 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(64)	rw	Y	STRING(64)	""

4.2.5 Object 0x6054: Device 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(64)	ro	-	STRING(64)	""

4.2.6 Object 0x6055: Device 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(64)	ro	-	STRING(64)	"www.moog.com"

4.2.7 Object 0x6056: Device 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: Device 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(64)	ro	-	STRING(64)	"Moog"

4.2.9 Object 0x605F: Device capability

This parameter is now only supported for compatibility reasons. According to device profile CiA 408 the information is now represented in the DeviceType 0x1000.

4.3 Vendor-specific objects

4.3.1 Object 0x2019: Software Version 1

The parameters 0x2019 up to 0x201B indicate all software versions actually used in this valve. The software is listed in arbitrary order.

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2019	0	SoftwareVersion1	STRING(64)	ro	-	None	Device-specific value

4.3.2 Object 0x201A: Software Version 2

The Parameters 0x2019 up to 0x201B indicate all software versions actually used in this valve. The software is listed in arbitrary order.

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x201A	0	SoftwareVersion2	STRING(64)	ro	-	None	Device-specific value

4.3.3 Object 0x201B: Software Version 3

The Parameters 0x2019 up to 0x201B indicate all software versions actually used in this valve. The software is listed in arbitrary order.

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x201B	0	SoftwareVersion3	STRING(64)	ro	-	None	Device-specific value

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 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).

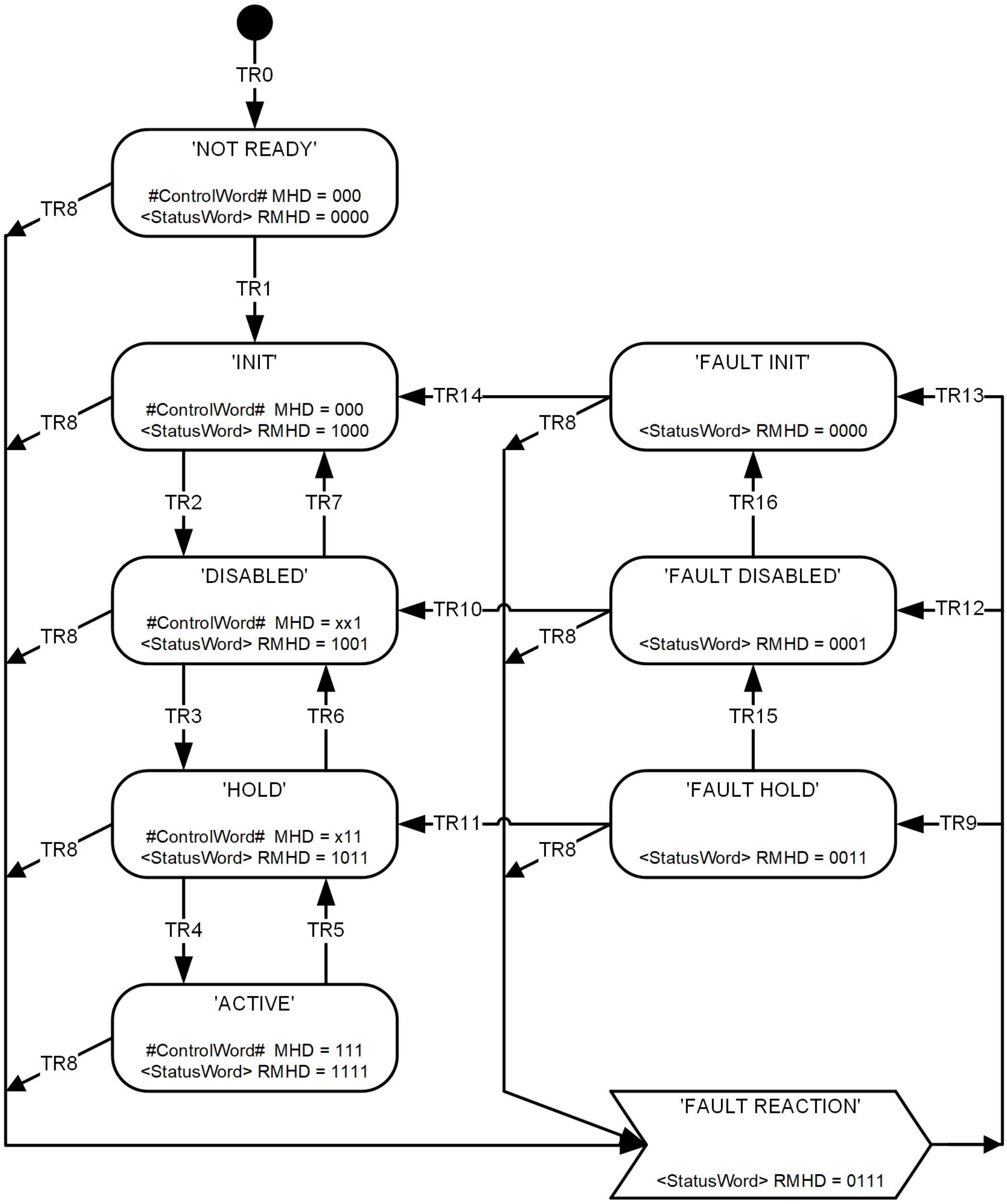


Figure 15: Device state machine

<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

5.1.1 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 146
- The setpoints from the bus or from the analog input according to the chosen <DeviceMode> (0x6042) are not effective.
  ➪ Chapter "6.1.1 Object 0x6042: Device mode", page 64
- The control loop depending on the selected <ControlMode> (0x6043) is active and the corresponding pre-defined hold setpoint is used, for example, the <QHoldSetpoint> (0x6314) for the spool control or the <PrsHoldSetpoint> (0x6394) for the pressure control or the <FlowHoldSetpoint> (0x6314) for the flow control.
  ➪ Chapter "6.1.3.3 Object 0x6314: Q setpoint conditioning hold setpoint", page 66
  ➪ Chapter "6.1.4.3 Object 0x6394: Pressure setpoint conditioning hold setpoint", page 69
  ➪ Chapter "6.1.5.3 Object 0x6714: Flow setpoint conditioning hold setpoint", page 71

'ACTIVE':

- Device parameters can be set.
- Actual values are available.
- The setpoints 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 146
- The setpoints from the bus or from the analog input according to the chosen ‹DeviceMode› (0x6042) are not effective.
  ➪ Chapter "6.1.1 Object 0x6042: Device mode", page 64
- The control loop depending on the selected ‹ControlMode› (0x6043) is active and the corresponding predefined hold setpoint is used, for example, the ‹QHoldSetpoint› (0x6314) for the spool control or the ‹PrsHoldSetpoint› (0x6394) for the pressure control or the ‹FlowHoldSetpoint› (0x6714).
  ➪ Chapter "6.1.3.3 Object 0x6314: Q setpoint conditioning hold setpoint", page 66
  ➪ Chapter "6.1.4.3 Object 0x6394: Pressure setpoint conditioning hold setpoint", page 69
  ➪ Chapter "6.1.5.3 Object 0x6714: Flow setpoint conditioning 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", page 275
- 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	Can change to fault state
'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 #EnableSignal# on is defined by the parameter ‹LocalControlModeDefault› (0x403F).
➪ Chapter "5.1.7 Object 0x403F: Local control word default", page 59

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!

5.1.2 State transitions

State transitions are caused by

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

5.1.2.1 DSM state transitions caused by the control word

The following table lists the transitions depending on the #ControlWord#.
⇨ Chapter "5.1 Device state machine (DSM)", page 50

The device control commands, which cause a state transition, are formed by the four low-order bits of the #ControlWord#.
⇨ Chapter "5.1.4 DSM control word source selection / local mode", page 58

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

Transition (TR)	Control Word	Control word bits								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 #EnableSignal# ⇨ Chapter "5.1.2.4 Enable behavior", page 54
TR4	Activate 'ACTIVE'	x	x	x	x	x	1	1	1	Depending on #EnableSignal# ⇨ Chapter "5.1.2.4 Enable behavior", page 54
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 #EnableSignal# toggles from 0 to 1. ⇨ Chapter "5.1.2.4 Enable behavior", page 54 Behavior of error output pin: ⇨ Chapter "5.1.2.5 Error output pin", page 55
		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 #EnableSignal# toggles from 0 to 1. ⇨ Chapter "5.1.2.4 Enable behavior", page 54 Behavior of error output pin: ⇨ Chapter "5.1.2.5 Error output pin", page 55
		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 #EnableSignal# toggles from 0 to 1. ⇨ Chapter "5.1.2.4 Enable behavior", page 54 Behavior of error output pin: ⇨ Chapter "5.1.2.5 Error output pin", page 55
		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	

5.1.2.2 DSM state transitions caused by the #EnableSignal#

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 "5.1.2.5 Error output pin", page 55
TR11	Reset 'FAULT_HOLD'	Behavior of error output pin: ➪ Chapter "5.1.2.5 Error output pin", page 55
TR14	Reset 'FAULT_INIT'	Behavior of error output pin: ➪ Chapter "5.1.2.5 Error output pin", page 55
TR15	'FAULT_HOLD' to 'FAULT_DISABLED'	Depending on the enable behavior

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

5.1.2.3 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 "5.1.2.5 Error output pin", page 55
TR8	A fault was detected. On entering 'FAULT_REACTION' state an emergency message is sent out if configured.	If state is 'INIT' or 'FAULT_INIT', state transitions to 'FAULT_HOLD' or 'FAULT_DISABLED' will be redirected to 'FAULT_INIT'. If state is 'DISABLED' or 'FAULT_DISABLED', state transitions to 'FAULT_HOLD' will be redirected to 'FAULT_DISABLED'. Behavior of error output pin: ➪ Chapter "5.1.2.5 Error output pin", page 55
TR9	Transition from 'FAULT_REACTION' to 'FAULT_HOLD' (fault reaction successful).	Depending on #EnableSignal# and the enable behavior. ➪ Chapter "5.1.2.4 Enable behavior", page 54
TR12	Transition from 'FAULT_REACTION' to 'FAULT_DISABLED' (fault reaction successful).	Depending on #EnableSignal# and the enable behavior. ➪ Chapter "5.1.2.4 Enable behavior", page 54
TR13	Transition from 'FAULT_REACTION' to 'FAULT_INIT' (fault reaction successful).	Depending on #EnableSignal# and the enable behavior. ➪ Chapter "5.1.2.4 Enable behavior", page 54

5.1.2.4 Enable behavior

The #EnableSignal# comes from the digital input 0. The #EnableSignal# influences the device state machine and can be used to acknowledge faults.

➪ Chapter "6.5 Digital inputs", page 101

5.1.2.4.1 DSM state transitions depending on the #EnableSignal#

The #EnableSignal# can cause different valve responses ('HOLD' or 'DISABLE'). The response to the #EnableSignal# is specified by the ordered servo valve configuration. 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.

!	State transitions do not necessarily refer to a specific spool position or change of spool position. The specific spool position also depends on the ordered configuration.

Depending on the ordered servo valve configuration, the #EnableSignal# is used to enable the valve. When the #EnableSignal# is switched off the device state machine changes directly to the state 'DISABLED'.

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

In another possible servo valve configuration, the #EnableSignal# is also used to enable the valve. The device state machine changes to the 'HOLD' state when the #EnableSignal# is switched off.

➪ Chapter "5.1.1 DSM states", page 51
➪ Chapter "6.1.3 Q setpoint path", page 65
➪ Chapter "6.1.4 p setpoint path", page 67

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

5.1.2.4.2 Fault confirmation with the #EnableSignal#

Toggling the #EnableSignal# 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.

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

5.1.2.5 Error output pin

The digital outputs 0…2 can be used to indicate fault states (negative logic) according to the Device Profile Fluid Power. To enable a digital output as error output, the corresponding parameter <DigitalOutputConfiguration0…2> (0x5E41) must be set to 2.

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

➪ Chapter "6.6.2 Object 0x5E41: Digital output configuration", page 104

5.1.3 DSM control word and status word

5.1.3.1 Object 0x6040: Device control word

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

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

Value description

Table 28: Possible values of parameter <ControlWord> (0x6040)

<ControlWord>				
Bit	**Description**	**<Control Mode> (0x6043) is set to 1…4**	**<Control Mode> (0x6043) is set to 5 or 14 (p/Q or p/flow control)**	**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…14	Reserved			
15	Ramp stop			Moog DCV

Bits 0, 1, 2, 3: 'DISABLED', 'HOLD', 'ACTIVE', 'RESET FAULTS'
The lower four bits within the control word represent the device state machine's (DSM) control command.
➪ Chapter "5.1.2.1 DSM state transitions caused by the control word", page 53

Bit 8: Enable pressure controller
This bit activates the pressure controller in the p/Q or p/flow control mode (<ControlMode> (0x6043) is set to 5 or 14).

0: Disables the pressure controller
1: Enables the pressure controller
➪ Chapter "7.1 Control modes", page 146

Bit 15: Ramp stop
If this bit is set, the ramp function output in the Q, p, and flow setpoint conditioning are frozen, in case of the ramp functions are enabled.

Q setpoint conditioning: ➪ Chapter "7.2.5 Ramp", page 158
p setpoint conditioning:: ➪ Chapter "7.4.5 Ramp", page 183
flow setpoint conditioning: ➪ Chapter "7.6.5 Ramp", page 213

5.1.3.2 Object 0x6041: Status word

The bit-coded ‹StatusWord› (0x6041) indicates the current device status.

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

Value description

Table 29: Possible values of parameter ‹StatusWord› (0x6041)

‹StatusWord›				
Bit	**‹StatusWord› bit**	**‹Control Mode› (0x6043) is set to 1...4**	**‹Control Mode› (0x6043) is set to 5 or 14 (p/Q or p/flow control)**	**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

Bits 0, 1, 2, 3: 'DISABLED', 'HOLD', 'ACTIVE', 'READY'
These bits indicate the state of the device state machine (DSM).
➪ Chapter "5.1 Device state machine (DSM)", page 50

Bit 4: Indicates that bit ‹Local› (0x604F) is set
The ‹LocalControlWord› (0x4040) is the active control word.
➪ Chapter "5.1.4 DSM control word source selection / local mode", page 58

Bit 8: Enable pressure controller
If the ‹Control Mode› (0x6043) is set to 5 (p/Q control) or 14 (p/flow control), 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 Q or flow controller limits the flow.
1: Output of pressure controller limits the flow.
➪ Chapter "7.1 Control modes", page 146

> **!** This bit is only active if the ‹ControlMode› (0x6043) is set to 5 (p/Q control) or 14 (p/flow control).

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

- Q and/or p and/or flow setpoint conditioning ramp function is active and
- Q and/or p and/or flow setpoint conditioning ramp function is running and
- #ControlWord# bit 15 is set to false.

Q setpoint conditioning: ➪ Chapter "7.2.5 Ramp", page 158
p setpoint conditioning: ➪ Chapter "7.4.5 Ramp", page 183
Flow setpoint conditioning: ➪ Chapter "7.6.5 Ramp", page 213

Bit 10: Limit value reached
This bit indicates that one of the setpoints is limited by the corresponding limit function set with the Q, p or flow setpoint conditioning.

Q setpoint conditioning: ➪ Chapter "7.2.3 Limiting", page 155
p setpoint conditioning: ➪ Chapter "7.4.5 Ramp", page 183
Flow setpoint conditioning: ➪ Chapter "7.6.5 Ramp", page 213

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 "7.9 Monitoring", page 226

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

- Q and/or p and/or flow setpoint conditioning ramp function is active and
- #ControlWord# bit 15 is set to true.

Q setpoint conditioning: ➪ Chapter "7.2.5 Ramp", page 158
p setpoint conditioning: ➪ Chapter "7.4.5 Ramp", page 183
Flow setpoint conditioning: ➪ Chapter "7.6.5 Ramp", page 213

5.1.4 DSM control word source selection / 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.

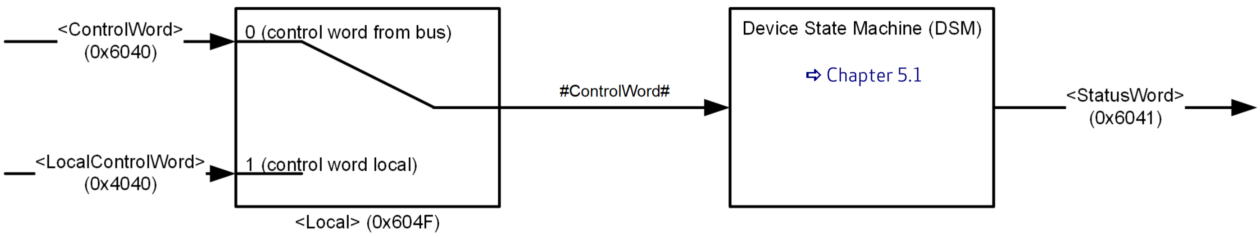


Figure 16: Local mode

❗	The signal #ControlWord# is an internal signal only. It links the signal from the control word source to the device state machine (DSM).

5.1.5 Object 0x604F: Device 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	Per-sistence	Value range	Default
0x604F	0	Local	INT8	rw	Y	0…1	0

Value description

Table 30: Possible values of parameter <Local> (0x604F)

<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.

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

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

➪ Chapter "5.1.3.2 Object 0x6041: Status word", page 57

5.1.6 Object 0x4040: Local control word

Parameter description: ➪ Chapter "5.1.3.1 Object 0x6040: Device control word", page 56

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

Value description

Same values as <ControlWord> (0x6040)
➪ Table 28, page 56

5.1.7 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	Per-sistence	Value range	Default
0x403F	0	LocalControlWordDefault	UINT16	rw	Y	UINT16	0x0107

5.2 DSM further state registers

5.2.1 Object 0x1002: Manufacturer Status Register

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

➪ Chapter "6.5 Digital inputs", page 101

Device							
Index	Subindex	Name	Data type	Access	Per-sistence	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	Per-sistence	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.

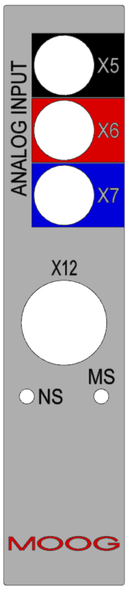


Figure 17: Status display LEDs

5.4.1 Module status LED «MS»

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

➪ Chapter "5.1 Device state machine (DSM)", page 50

Module status LED «MS»	Valve State Machine (status word) (according to Device Profile Fluid Power)	Description
Off		No supply power
Red	'NOT READY'	Unrecoverable error. This state can be caused by fault reactions 'FAULT_INIT' or 'FAULT_STOP'. ➪ Chapter "8.1.3 Fault reaction", page 275
Orange blinking	'INIT'	Servo valve initialisation successful / Servo valve standby mode
Orange	'DISABLED'	Servo valve standby mode
Green blinking	'HOLD'	Normal operation
Green	'ACTIVE'	Normal operation
Red blinking	'FAULT_INIT' or '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", page 275

5.4.2 Network status LED «NS»

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

➪ Chapter "2.10 Network management (NMT) state machine", page 17

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 occurred.

6 Signal routing and scaling

The following picture shows the structure of the signal routing for the setpoint and the physical actual values of the servo valve depending on the <ControlMode> (0x6043) used. Most of the valves are delivered and operated in the configuration shown below. The exact number of analog and digital inputs and outputs as well as the factory settings depend on the valve variant ordered. The blocks with gray backgrounds are described in detail in this chapter.

➪ Chapter "7.1 Control modes", page 146

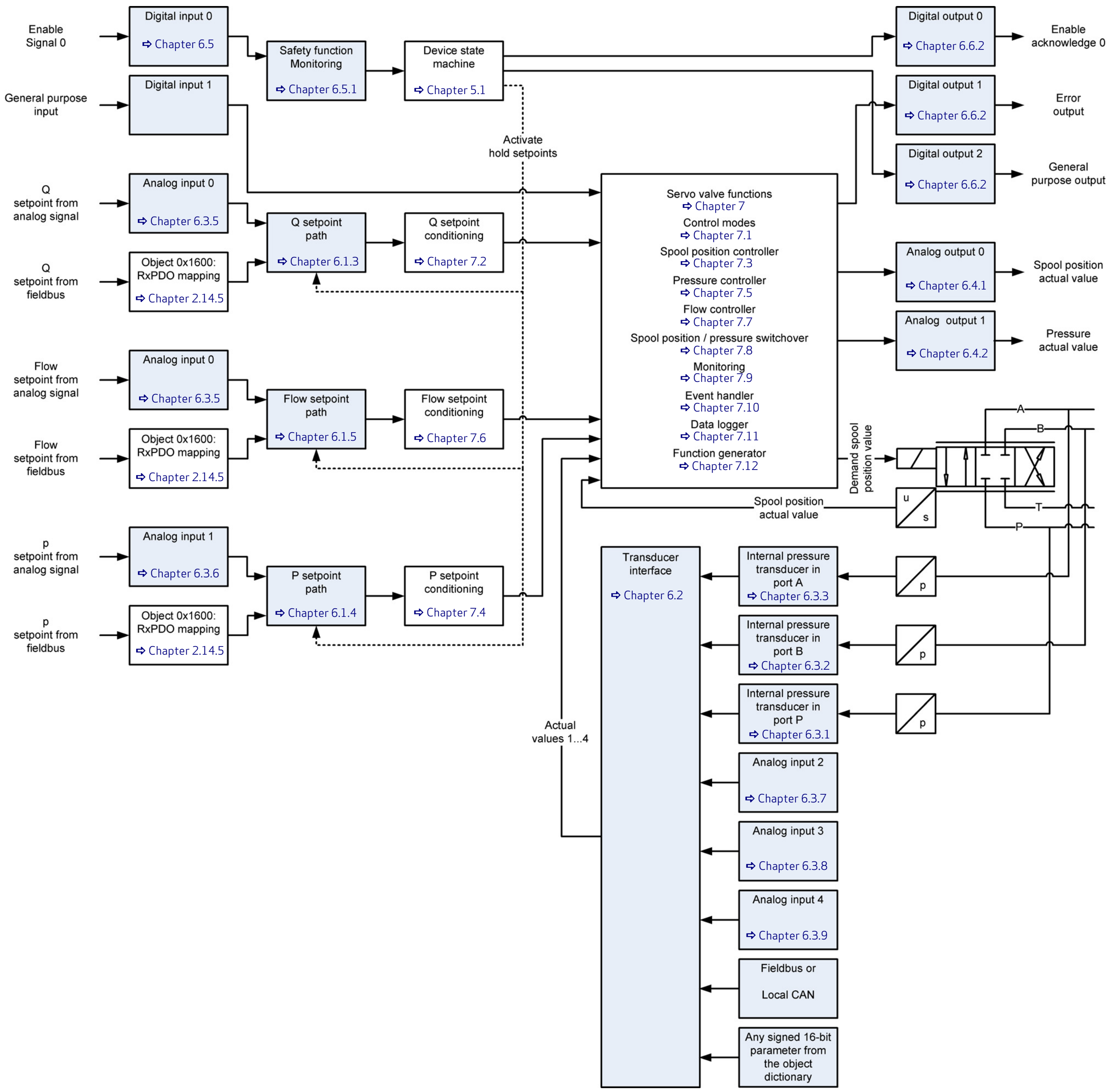


Figure 18: Signal routing

6.1 Setpoint path

The setpoint for spool position, pressure and flow can be received via the fieldbus or via the analog inputs. The current configuration of individual models may differ. Which setpoint is in effect depends on the parameter <DeviceMode> (0x6042) and the <StatusWord> (0x6041) and <DeviceType> (0x1000). Depending on the device type bits 29…30, the 16 or 32 bit setpoint is used.

Setpoints received via fieldbus:

- The Q setpoint parameter <Q_Setpoint> (0x6300), 16 bit.
- The Q setpoint parameter <Q_Setpoint 32 Bit> (0x6308), 32 bit.
- The p setpoint parameter <PrsSetpoint> (0x6380), 16 bit.
- The p setpoint parameter <PrsSetpoint 32 Bit> (0x6388), 32 bit.
- The flow setpoint <FlowSetpoint> (0x6700), 16 bit.
- The flow setpoint <FlowSetpoint 32Bit> (0x6708), 32 bit.

Setpoints from local source (analog input):

- The Q setpoint analog input 0 <AnaInpActualValue0> (0x3204), 16 bit.
- The p setpoint analog input 1 <AnaInpActualValue1> (0x320C), 16 bit.
- The flow setpoint analog input 0 <AnaInpActualValue0> (0x3204), 16 bit.

Examples for 16 bit setpoints:

- <DeviceMode> (0x6042) is set to 1 (setpoint input via bus), <StatusWord> (0x6041) is 0111b ('ACTIVE'):
  Result: The setpoint received via fieldbus is forwarded to the setpoint conditioning.
- <DeviceMode> (0x6042) is set to 2 (setpoint input locally), <StatusWord> (0x6041) is 0111b ('ACTIVE'):
  Result: The setpoint coming from the analog input is forwarded to the setpoint conditioning.
- <StatusWord> (0x6041) is 0011b ('HOLD').
  Result: The hold setpoints are taken as input. These hold setpoints are 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 Q hold setpoint is stored in the parameter <QHoldSetpoint> (0x6314).
The p hold setpoint is stored in the parameter <PrsHoldSetpoint> (0x6394).
The flow hold setpoint is stored in the parameter <FlowHoldSetpoint> (0x6714).

6.1.1 Object 0x6042: Device mode

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6042	0	DeviceMode	INT8	rw	N	1…2	<DeviceMode-Default> (0x4042)

Value description

Table 31: Possible values of parameter <DeviceMode> (0x6042)

<DeviceMode>	**Type of analog input**
1	Setpoint input via the bus
2	Setpoint input locally

The effective <DeviceMode> (0x6042) after power up is defined by the parameter <DeviceModeDefault> (0x4042).
⇨ Chapter "6.1.2 Object 0x4042: Device mode default", page 64

6.1.2 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	Per-sistence	Value range	Default
0x4042	0	DeviceModeDefault	INT8	rw	Y	1…2	1

Value description

⇨ Chapter "6.1.1 Object 0x6042: Device mode", page 64

6.1.3 Q setpoint path

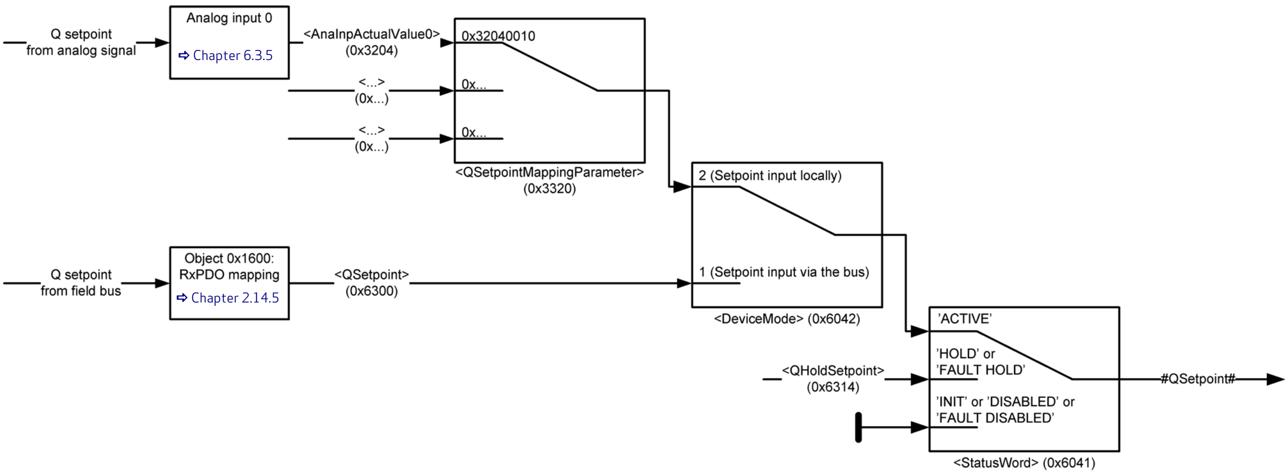


Figure 19: Q setpoint path

6.1.3.1 Object 0x6300: Q setpoint

This parameter contains the Q setpoint which is received from the fieldbus. Depending on the ‹DeviceMode› (0x6042), this parameter is in effect for the following three control modes stored in the parameter ‹ControlMode› (0x6043):

1 Q control open loop
2 Q control closed loop
5 p/Q control

The setpoint ‹QSetpoint› (0x6300) takes only effect if the ‹StatusWord› (0x6041) is 0111_b ('ACTIVE') and the ‹DeviceMode› (0x6042) is set to 1 (setpoint input via bus).

➪ Chapter "5.1.3.2 Object 0x6041: Status word", page 57
➪ Chapter "6.1.1 Object 0x6042: Device mode", page 64

ValveQControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6300	1	QSetpoint	INT16	rw	N	INT16	None
0x6300	2	Unit	UINT8	ro	-	UINT8	0
0x6300	3	Prefix	INT8	ro	-	-4...0	0

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

6.1.3.2 Object 0x3320: Q setpoint source selection parameter

The Q setpoint selection parameter points to the input where the Q setpoint comes from.

The Q setpoint <QSetpoint> (0x6300) is only effective in case the <StatusWord> (0x6041) is 0111b ('ACTIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).

➪ Chapter "5.1.3.2 Object 0x6041: Status word", page 57
➪ Chapter "6.1.1 Object 0x6042: Device mode", page 64

ValveQControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3320	0	QSetpointMappingParameter	UINT32	rw	Y	UINT32	0x63000110

Value description

Table 32: Possible values of parameter <QSetpointMappingParameter> (0x3320)

<QSetpointMappingParameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length: 0x10 or 0x20
Default	0x63	0x00	0x01	0x10

This reference 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 or 0x20.

The default value is 0x63000110, which refers to the <QSetpoint> (0x6300), sub-index 0x01 with a length of 16 bit (16=0x10).

6.1.3.3 Object 0x6314: Q setpoint conditioning hold setpoint

This parameter defines the Q hold setpoint for the <ControlMode> (0x6043):

1 Q control open loop
2 Q control closed loop
5 p/Q control

➪ Chapter "7.1.1 Object 0x6043: Device control mode", page 146

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

➪ Chapter "5.1.3.2 Object 0x6041: Status word", page 57

ValveQControl_SetpointConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6314	1	QHoldSetpoint	INT16	rw	Y	INT16	0
0x6314	2	Unit	UINT8	ro	-	UINT8	0
0x6314	3	Prefix	INT8	ro	-	-4...0	0

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

6.1.3.4 Object 0x6312: Common spool position setpoint

This parameter represents the device internal setpoint which is picked up after the setpoint conditioning and pressure control blocks.

ValveQControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6312	1	SpoolPositionSetpoint	INT16	rw	N	INT16	None
0x6312	2	Unit	UINT8	ro	-	UINT8	0
0x6312	3	Prefix	INT8	ro	-	−4…0	0

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

6.1.4 p setpoint path

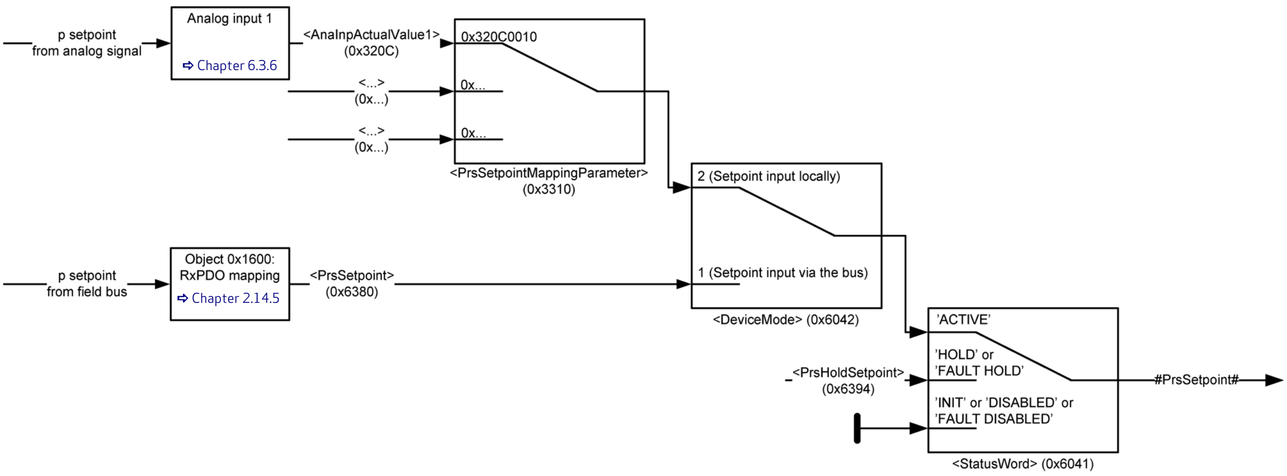


Figure 20: p setpoint path

6.1.4.1 Object 0x6380: Pressure setpoint

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

3 p open loop
4 p closed loop
5 p/Q control
14 Flow control

➪ Chapter "7.1.1 Object 0x6043: Device control mode", page 146

The setpoint 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 "5.1.3.2 Object 0x6041: Status word", page 57
➪ Chapter "6.1.1 Object 0x6042: Device mode", page 64

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6380	1	PrsSetpoint	INT16	rw	N	INT16	None
0x6380	2	Unit	UINT8	ro	-	UINT8	0
0x6380	3	Prefix	INT8	ro	-	-4…0	0

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

6.1.4.2 Object 0x3310: Pressure setpoint source selection parameter

The p setpoint selection parameter points to the input where the p setpoint comes from.

The p setpoint <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	Per-sistence	Value range	Default
0x3310	0	PrsSetpointMappingParameter	UINT32	rw	Y	UINT32	0x63800110

Value description

Table 33: Possible values of parameter <PrsSetpointMappingParameter> (0x3310)

<PrsSetpointMappingParameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length: 0x10 or 0x20
Default	0x63	0x80	0x01	0x10

This reference 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 or 0x20.

The default value is 0x63800110, which refers to the <PrsSetpoint> (0x6380), sub-index 0x01 with the length of 16 bit (16 = 0x10).

6.1.4.3 Object 0x6394: Pressure setpoint conditioning hold setpoint

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

3 p control open loop
4 p control closed loop
5 p/Q control
8 p/Flow control

➪ Chapter "7.1.1 Object 0x6043: Device control mode", page 146

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

➪ Chapter "5.1.3.2 Object 0x6041: Status word", page 57

ValvePressureControl_SetpointConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6394	1	PrsHoldSetpoint	INT16	rw	Y	INT16	0
0x6394	2	Unit	UINT8	ro	-	UINT8	0
0x6394	3	Prefix	INT8	ro	-	-4...0	0

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

6.1.5 Flow setpoint path

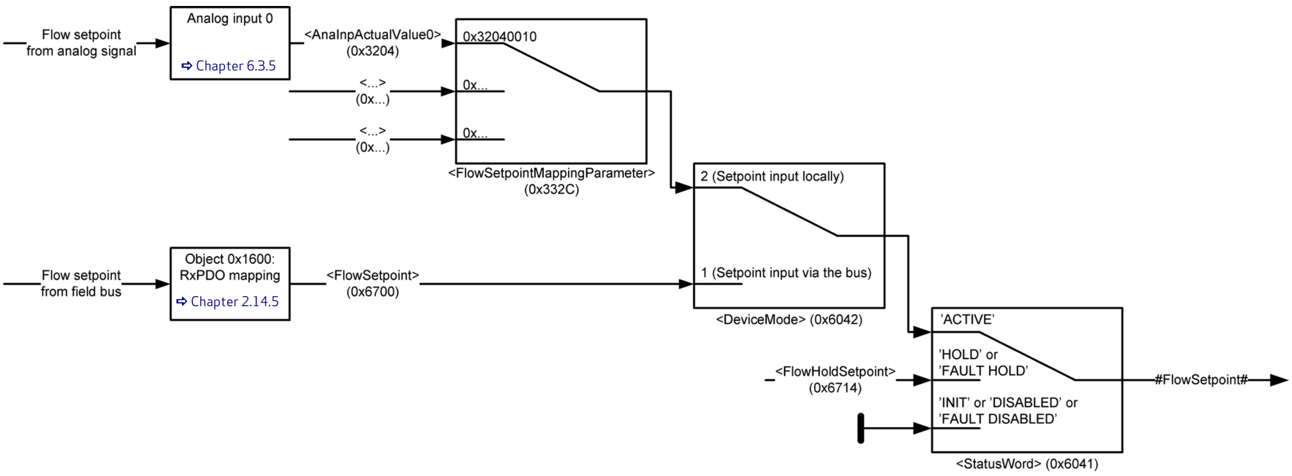


Figure 21: Flow setpoint path

6.1.5.1 Object 0x6700: Flow setpoint

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

13 flow control (not implemented)
14 p/Flow control

➪ Chapter "7.1.1 Object 0x6043: Device control mode", page 146

The setpoint 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 "5.1.3.2 Object 0x6041: Status word", page 57
➪ Chapter "6.1.1 Object 0x6042: Device mode", page 64

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6700	1	FlowSetpoint	INT16	rw	N	INT16	None
0x6700	2	Unit	UINT8	ro	-	UINT8	0
0x6700	3	Prefix	INT8	ro	-	-4…0	0

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

6.1.5.2 Object 0x332C: Flow setpoint source selection parameter

The flow selection mapping parameter points to the input where the flow setpoint comes from.

The flow setpoint <FlowSetpoint> (0x6700) is only effective in case the <StatusWord> (0x6041) is 1111_b ('ACTIVE') and the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus).

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x332C	0	FlowSetpointMappingParameter	UINT32	rw	Y	UINT32	0x67000110

Value description

Table 34: Possible values of parameter <FlowSetpointMappingParameter> (0x332C)

<FlowSetpointMappingParameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length: 0x10 or 0x20
Default	0x63	0x80	0x01	0x10

This reference 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 or 0x20.

The default value is 0x67000110 which refers to the <FlowSetpoint> (0x6700), sub-index 0x01 with the length of 16 bit (16 = 0x10).

6.1.5.3 Object 0x6714: Flow setpoint conditioning hold setpoint

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

13 flow control (not implemented)
14 p/flow control

➪ Chapter "7.1.1 Object 0x6043: Device control mode", page 146

The <FlowHoldSetpoint> (0x6714) acts as setpoint in case of <StatusWord> (0x6041) equals 'HOLD' or 'FAULT_HOLD'.

➪ Chapter "5.1.3.2 Object 0x6041: Status word", page 57

ValveFlowControl_SetpointConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6714	1	FlowHoldSetpoint	INT16	rw	Y	INT16	0
0x6714	2	Unit	UINT8	ro	-	UINT8	0
0x6714	3	Prefix	INT8	ro	-	−4…0	0

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

6.1.5.4 Object 0x2035: Q setpoint size

The parameter indicates the size of the spool position setpoint and actual values. This can be 16 or 32 bits depending on the <device type> (0x1000), bits 29…30.

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2035	0	QSize	UINT8	ro	-	0 for 16 bit values 1 for 32 bit values	Device-specific value

6.1.5.5 Object 0x2036: Pressure setpoint size

The parameter indicates the size of the pressure setpoint and actual values. This can be 16 or 32 bits depending on the <device type> (0x1000), bits 29…30.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2036	0	PressureSize	UINT8	ro	-	0 for 16 bit values 1 for 32 bit values	Device-specific value

6.1.5.6 Object 0x2037: Flow setpoint size

The parameter indicates the size of the flow setpoint and actual values. This can be 16 or 32 bits depending on the <device type> (0x1000), bits 29…30.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2037	0	FlowSize	UINT8	ro	-	0 for 16 bit values 1 for 32 bit values	Device-specific value

6.2 Actual value transducer interface

Depending on the servo valve configuration, up to 5 external analog inputs and 3 internal pressure sensors can be ordered. The following sources for analog values can be mapped and configured via the transducer interface:

- Analog input 0…4
- Internal pressure sensors at port A, B, P
- For special application any parameter can be mapped to an interface e.g. from a fieldbus.

The following figure shows how routing and scaling of the actual value is done for the available inputs.

With the interface type <Type> (0x6102), the type of value conditioning can be adapted to the selected sensor.

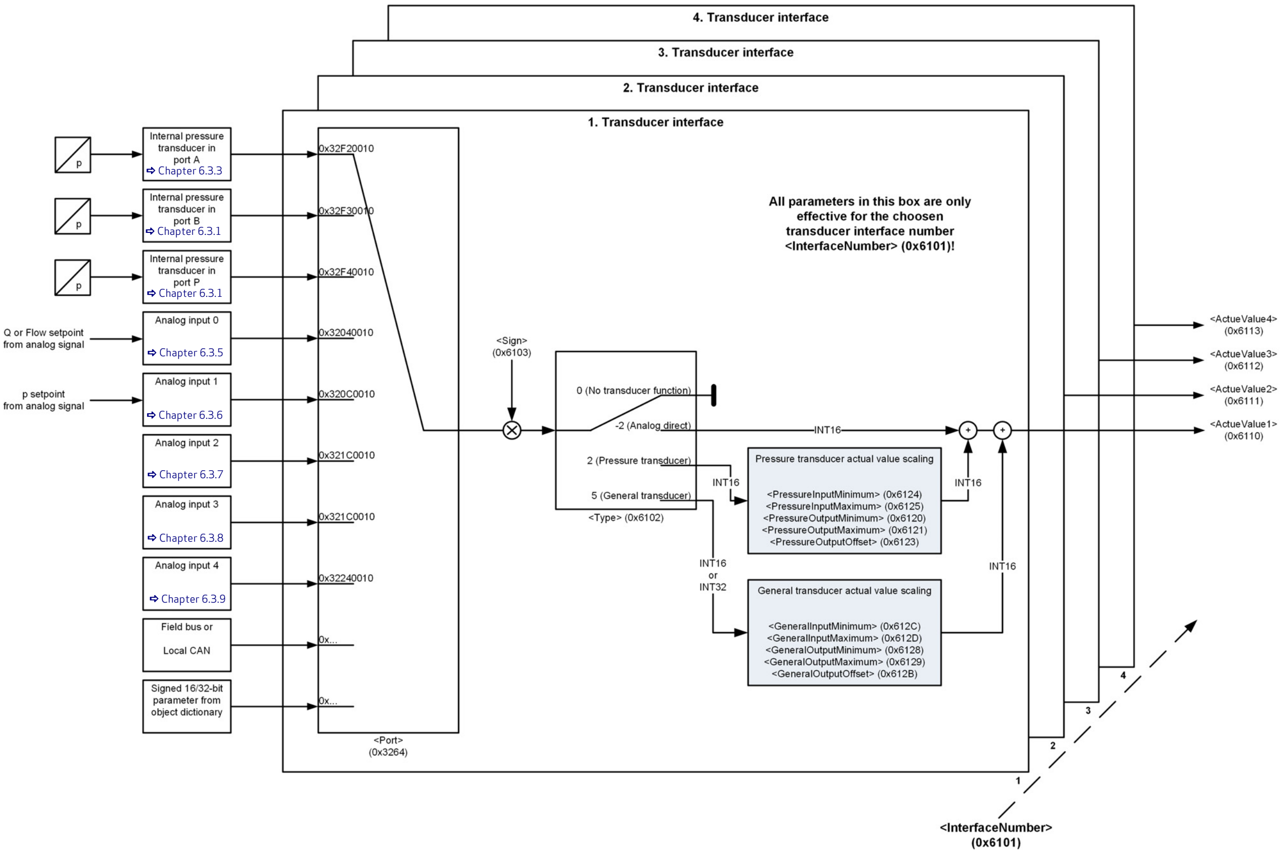


Figure 22: Actual value path

!	Per default, analog input 0 and 1 are used for the analog setpoints. The analog inputs 2…4 are normally used for actual values. ⇨ Chapter "6.3 Analog inputs", page 83

!	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> (0x6101).

When changing the <InterfaceNumber> (0x6101) the following parameters represent the configuration of the selected interface.

Index	Parameter Name	<Type> (0x6102) (The type specifies the way the interface is used.)
0x6100	<InterfaceNumberMax>	All types
0x6101	<InterfaceNumber>	All types
0x6102	<Type>	All types
0x6103	<Sign>	All types
0x6104	<ActualValue>	All types
0x3264	<Port>	All types
0x6120	<PressureOutputMinimum>	2: Pressure input
0x6121	<PressureOutputMaximum>	2: Pressure input
0x6124	<PressureInputlMinimum>	2: Pressure input
0x6125	<PressureInputMaximum>	2: Pressure input
0x6123	<PressureOutputOffset>	2: Pressure input
0x6122	<PressureArea>	2: Pressure input
0x6128	<GeneralOutputMinimum>	5: General input
0x6129	<GeneralOutputMaximum>	5: General input
0x612C	<GeneralInputlMinimum>	5: General input
0x612D	<GeneralInputlMaximum>	5: General input
0x612B	<GeneralOutputOffset>	5: General input

6.2.1 Object 0x6110: Actual value conditioning actual value 1

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6110	1	ActualValue1	INT16	ro	-	INT16	None

6.2.2 Object 0x6111: Actual value conditioning actual value 2

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6111	1	ActualValue2	INT16	ro	-	INT16	None

6.2.3 Object 0x6112: Actual value conditioning actual value 3

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6112	1	ActualValue3	INT16	ro	-	INT16	None

6.2.4 Object 0x6113: Actual value conditioning actual value 4

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6113	1	ActualValue4	INT16	ro	-	INT16	None

6.2.5 Transducer interface definition

The assignment of an actual value source to an interface is done by setting the parameter <Port> (0x3264). The type of the actual value conditioning is configured by the parameter <Type> (0x6102). All parameters configuring the interface only apply to the interface selected by the parameter <InterfaceNumber> (0x6101).

6.2.5.1 Object 0x6100: Actual value conditioning max interface number

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6100	0	InterfaceNumberMax	UINT8	ro	-	UINT8	4

6.2.5.2 Object 0x6101: Actual value conditioning interface number

This parameter defines the actual referenced interface.

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6101	0	InterfaceNumber	UINT8	rw	N	1…4	None

Value description

Table 35: Possible values of parameter <InterfaceNumber> (0x6101)

<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

6.2.5.3 Object 0x6102: Actual value conditioning type

This interface type <Type> (0x6102) defines the method of the value conditioning.
This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6101).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6102	0	Type	INT8	rw	N	−2, 0, 2, 5	None

Value description

Table 36: Possible values of parameter <Type> (0x6102)

<Type>	Description
0 (no transducer function)	Interface deactivated.
2 (pressure transducer)	Pressure sensor actual value conditioning with INT16 input values. ➪ Chapter "6.2.6 Pressure actual value scaling", page 77
5 (general transducer)	General input actual value conditioning with INT16 or INT32 input values. ➪ Chapter "6.2.7 General actual value scaling", page 79
−2 (analog direct)	No further scaling active, INT16 input values.

6.2.5.4 Object 0x6103: Actual value conditioning sign

This parameter defines the sign of the actual value.
This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6101).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6103	0	Sign	INT8	rw	N	−1 or +1	None

Value description

Table 37: Possible values of parameter <Sign> (0x6103)

<Sign>	Description
−1	Negative
1	Positive

6.2.5.5 Object 0x6104: Actual value conditioning actual value

Compared to the objects <Actual value 1…4> (0x6110…0x6113), this parameter contains the output value of the actual selected interface.

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6104	0	ActualValue	INT16	ro	-	INT16	None

6.2.5.6 Object 0x3264: Port

This mapping 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> (0x6101).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3264	0	Port	UINT32	rw	N	UINT32	None

Value description

Table 38: Possible values of parameter <Port> (0x3264)

<Port>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length: 0x10 or 0x20
Default	0x00	0x00	0x00	0x00

This reference 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 or 0x20.

There are 8 analog inputs (0…4 and the internal pressure transducer A, B, P). Each input has its actual value parameter. Each actual value can be assigned to an interface by the <Port> (0x3264) parameter as described in the examples below.

Connector	**Index of actual value**	**Sub-index of - actual value**	**Parameter length**	**Resulting <Port> 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 P	0x32F2	0x00	0x10	0x32F20010
Internal pressure transducer B	0x32F3	0x00	0x10	0x32F30010
Internal pressure transducer A	0x32F4 or 0x3404	0x00	0x10	0x32F40010 or 0x34040010

The actually available analog inputs depend on the ordered valve configuration.
⇨ Chapter "6.3 Analog inputs", page 83

6.2.5.7 Object 0x3270: Data 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.

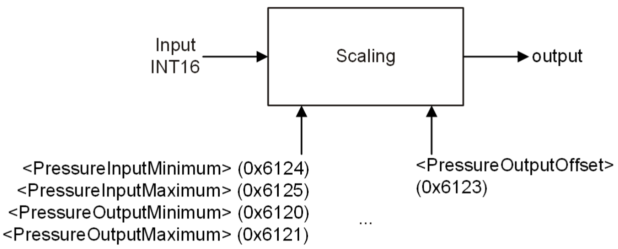
Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3270	0	DataStructure	DOMAIN	rw	Y	DOMAIN	None

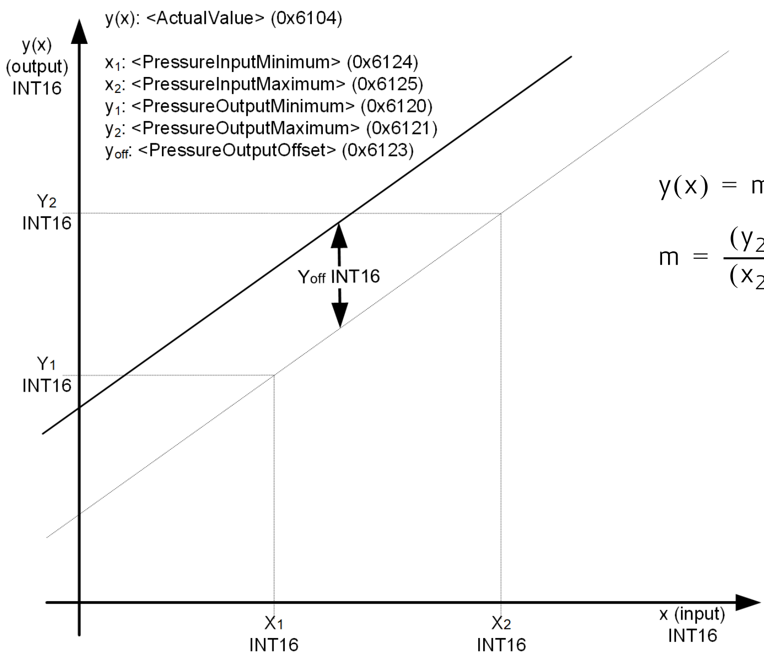
!	This parameter is for Moog internal use only. The structure and length may change in the future.

6.2.6 Pressure actual value scaling

To activate the actual pressure value scaling the interface type needs to be configured to "pressure transducer". This is done by writing the value 2 to the parameter <Type> (0x6102).
This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6101).

➪ Chapter "6.2.5.3 Object 0x6102: Actual value conditioning type", page 75





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

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

Figure 23: Pressure actual value scaling

6.2.6.1 Object 0x6120: Actual value conditioning min pressure for pressure transducer

This parameter defines the minimum transducer interface output signal data point y1 of the two-point scaling function (when pressure offset equals 0).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6120	1	PressureOutputMinimum	INT16	rw	N	INT16	None

6.2.6.2 Object 0x6121: Actual value conditioning max pressure for pressure transducer

This parameter defines the maximum transducer interface output signal data point y2 of the two-point scaling function (when pressure offset equals 0).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6121	1	PressureOutputMaximum	INT16	rw	N	INT16	None

6.2.6.3 Object 0x6124: Actual value conditioning min transducer signal for pressure transducer

This parameter defines the transducer interface input signal data point x1 of the two-point scaling function (when pressure offset equals 0).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6124	1	PressureInputMinimum	INT16	rw	N	INT16	None

6.2.6.4 Object 0x6125: Actual value conditioning max transducer signal for pressure transducer

This parameter defines the transducer interface input signal data point x2 of the two-point scaling function (when pressure offset equals 0).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6125	1	PressureInputMaximum	INT16	rw	N	INT16	None

6.2.6.5 Object 0x6123: Actual value conditioning pressure offset for pressure

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6123	1	PressureOutputOffset	INT16	rw	N	INT16	None

6.2.6.6 Object 0x6122: Actual value conditioning for pressure transducer

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.

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6122	1	PressureArea	INT16	rw	N	INT16	None

6.2.7 General actual value scaling

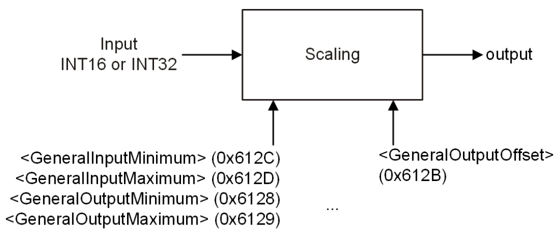
The general actual value scaling is used to scale an INT16 or INT32 input to the internal INT16 data type.

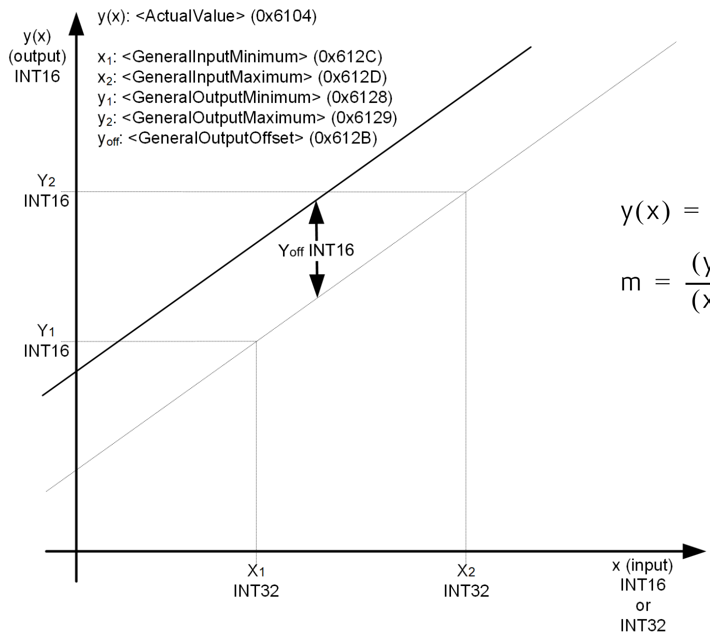
Example: an external pressure transducer with CAN interface is mapped to the receive PDO <INTEGER32> (0x1104).

To activate the general actual value scaling, the interface type needs to be configured to "general input". This is done by writing the value 5 to the parameter <Type> (0x6102).

This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6101).

⇨ Chapter "6.2.5.3 Object 0x6102: Actual value conditioning type", page 75





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

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

Figure 24: General input scaling

6.2.7.1 Object 0x6128: Actual value conditioning min general transducer

This parameter defines the minimum transducer interface output signal data point y1 of the two-point scaling function (when pressure offset equals 0).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6128	1	GeneralOutputMinimum	INT16	rw	N	INT16	None

6.2.7.2 Object 0x6129: Actual value conditioning max general transducer

This parameter defines the maximum transducer interface output signal data point y2 of the two-point scaling function (when pressure offset equals 0).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6129	1	GeneralOutputMaximum	INT16	rw	N	INT16	None

6.2.7.3 Object 0x612C: Actual value conditioning min general signal transducer 32-bit

This parameter defines the transducer interface input signal data point x1 of the two-point scaling function (when pressure offset equals 0).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x612C	1	GeneralInputMinimum	INT32	rw	N	INT32	None

6.2.7.4 Object 0x612D: Actual value conditioning max general signal transducer 32-bit

This parameter defines the transducer interface input signal data point x2 of the two-point scaling function (when pressure offset equals 0).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x612D	1	GeneralInputMaximum	INT32	rw	N	INT32	None

6.2.7.5 Object 0x612B: Actual value conditioning general transducer offset

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x612B	1	GeneralOutputOffset	INT16	rw	N	INT16	None

6.2.8 Parameterization examples

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

6.2.8.1 Get active transducer interface number and output value

The <InterfaceNumber> (0x6101) shows the active transducer interface number. The output value of the active transducer interface can be read from the parameter <ActualValue> (0x6104).

6.2.8.2 Example 1: Enable/disable transducer interface

1. Select the transducer interface which is intended to be enabled or disabled.
   To do this set the <InterfaceNumber> (0x6101).
2. Enable or disable the selected transducer interface and select the method of conditioning.
   To do this set the interface type <Type> (0x6102) to:
   0 (no transducer function) or
   2 (pressure transducer) or
   −2 (analog direct)

6.2.8.3 Example 2: Change sign of the transducer signal

1. Select the transducer interface which is intended to change the sign.
   To do this set the <InterfaceNumber> (0x6101).
2. Change the sign of the transducer interface.
   To do this set the <Sign> (0x6103) to:
   1 (positive) or
   −1 (negative)

6.2.8.4 Example 3: Adjust transducer interface without scaling

1. Select the transducer interface which is intended to be adjusted.
   Therefore set the <InterfaceNumber> (0x6101).
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 <DummyDataS16> (0x2921) should be used as input:

Index:	0x2921
Index MSB:	0x00
Index LSB:	0x03
Sub-index:	0x00
Parameter bit length:	0x10 (Only parameters with a bit length of 0x10 or 0x20 are allowed to be mapped!)

   Build the address value in the following manner:

Byte	3	2	1	0	Result
Description	Index MSB	Index LSB	Sub-index	Parameter length in bit: 0x10	
Example	0x29	0x21	0x00	0x10	0x29210010

   Write the result 0x29210010 into the parameter <Port> (0x3264).
3. Check the transducer interfaces <Sign> (0x6103) and change the value (1 or −1) if needed.
4. Set the transducer interface <Type> (0x6102) to −2 (analog direct).

6.2.8.5 Example 4: Adjust transducer interface with scaling

1. Select the transducer interface which is intended to be adjusted.
   To do this set the <InterfaceNumber> (0x6101).
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> (0x3214) 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 or 0x20 are allowed to be mapped!)

   Build the address value in the following manner:

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

   Write the result 0x32140010 in the parameter <Port> (0x3264).
3. Check transducer interfaces <Sign> (0x6103) and change value (1 or −1) if needed.
4. Set transducer interface <Type> (0x6102) to 2 (pressure transducer).
5. Set the <PressureOutputOffset> (0x6123) 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: <PressureInputMinimum> (0x6124)
   x_2: <PressureInputMaximum> (0x6125)
   y_1: <PressureOutputMaximum> (0x6120)
   y_2: <PressureOutputMaximum> (0x6121)

6.3 Analog inputs

Depending on the servo valve configuration ordered, up to 3 internal pressure transducers and up to 5 external analog inputs are available.

In case the integrated pressure sensor module is used, the values of port P, B, A are represented in the parameters <PressureSensorActualValue0…2> (0x32F2…0x32F4).

In case the integrated single pressure sensor is used, the pressure value of port A is written to <ActualValue> (0x3404).

The type of the analog inputs 0…4 depends on the ordered servo valve configuration. All analog to digital converters have the same resolution of 12 bit. All analog inputs can be operated grounded or potential free.
➪ Chapter "6.2.5 Transducer interface definition", page 74

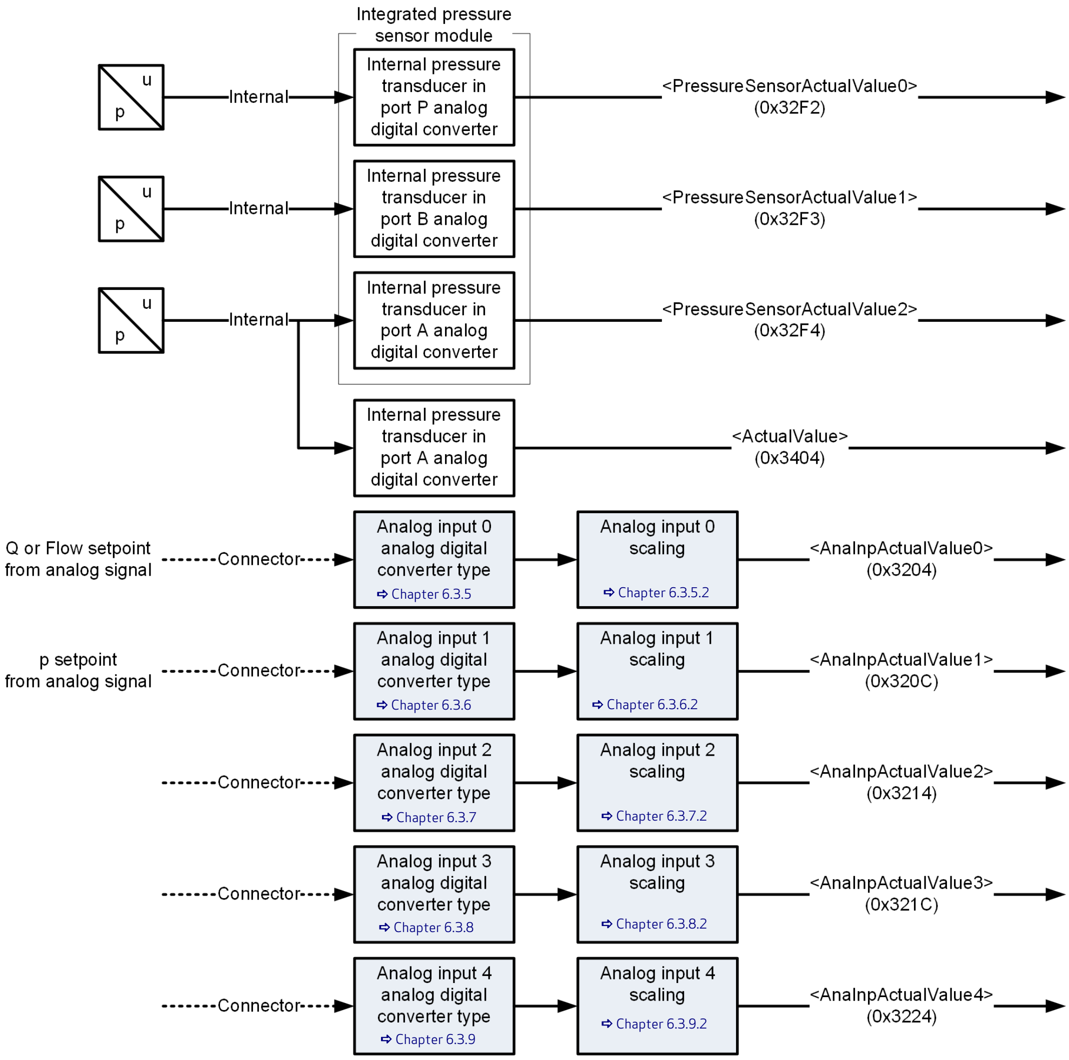


Figure 25: Analog inputs

The following 5 configuration variants are possible, depending on the ordered valve:

1. Analog inputs A, 0, 1, 2, 3, 4
2. Analog inputs A, B, P, 2, 3, 4
3. Analog inputs A, P, 0, 1, 2, 3
4. Analog inputs A, B, 0, 1, 2, 4
5. Analog inputs B, P, 0, 1, 2, 4
6. Analog inputs A, B, P, 0, 1, 2

Variant 1: Pressure actual value of port A is mapped to ‹ActualValue› (0x3404)
Variant 2…5: Pressure actual value of port A is mapped to ‹PressureSensorActualValue2› (0x32F4)

➪ Document CA63420-001 "User Manual Electrical Interfaces"

The analog input 0 is intended for the Q setpoint and the analog input 1 is intended for the p setpoint. The analog inputs 0 and 1 will only be routed to the setpoint conditioning, if the ‹DeviceMode› (0x6042) is set to 2 (setpoint input locally). The setpoints are provided by the fieldbus, 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 "6.1.3 Q setpoint path", page 65

The analog inputs are only effective as setpoint inputs, if the ‹DeviceMode› (0x6042) is set to 2 (setpoint input locally).

6.3.1 Object 0x32F2: Internal pressure transducer at port P

This parameter represents the pressure of the valve's port P in percent of the nominal pressure. This value is only available if the integrated pressure sensor module was ordered.

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x32F2	0	PressureSensorActualValue0	INT16	ro	-	INT16	None

6.3.2 Object 0x32F3: Internal pressure transducer at port B

This parameter represents the pressure of the valve's port B in percent of the nominal pressure. This value is only available if the integrated pressure sensor module was ordered.

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x32F2	0	PressureSensorActualValue1	INT16	ro	-	INT16	None

6.3.3 Object 0x32F4: Internal pressure transducer at port A

This parameter represents the pressure of the valve's port A in percent of the nominal pressure. This value is only available if the integrated pressure sensor module was ordered.

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x32F4	0	PressureSensorActualValue2	INT16	ro	-	INT16	None

6.3.4 Object 0x3404: Internal pressure transducer at port A

This parameter represents the pressure of the valve's port A in percent of the nominal pressure. This value is only available if the integrated single pressure sensor was ordered.

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3404	0	ActualValue	INT16	ro	-	INT16	None

6.3.5 Analog input 0

This analog input can be used for general purposes. By default, it is used as Q or Flow setpoint input.

6.3.5.1 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	Per-sistence	Value range	Default
0x3200	0	AnaInpType0	INT8	rw	Y	1…12	1

Value description

Table 39: Possible values of parameter <AnaInpType0> (0x3200)

<AnaInpType0>		Value range of <AnaInpActualValue0>		
Potential-free	**Grounded**	**0 % or -100 %**	**+100 %**	**Value range of electrical signal**
1	9	-16384	16384	±10 V (±100 %)
2	10	0	16384	0…10 V (0…100 %)
3	6	-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 %)

The available input types (potential-free/grounded) depend on the ordered valve option.
⇨ Document CA63420-001 "User Manual Electrical Interfaces"

6.3.5.2 Object 0x3207: Scaling

The analog input 0 can be scaled directly. If a pressure transducer is connected to analog input 0, the scaling must be selected so that 100 % or 16384 increments corresponds to the ordered nominal pressure of the servo valve.

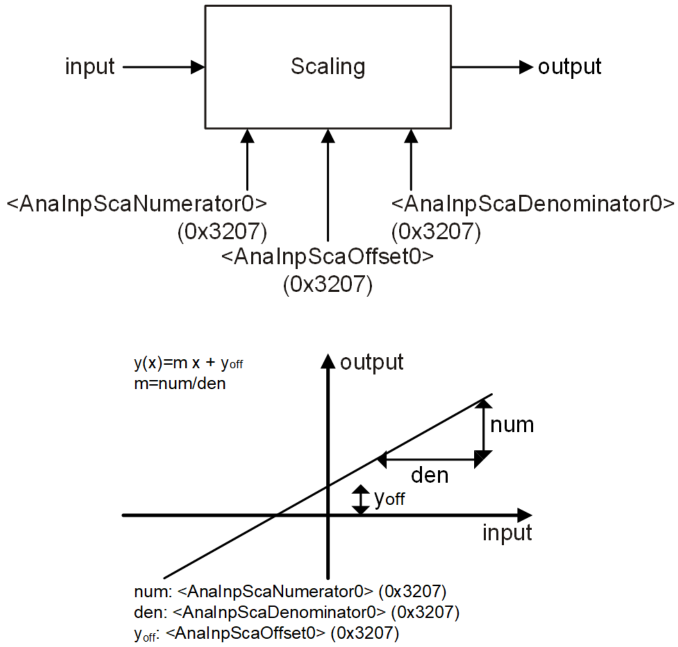


Figure 26: Analog input 0 scaling

AnalogueInput0							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3207	1	AnaInpScaNumerator0	INT16	rw	Y	INT16	16384
0x3207	2	AnaInpScaDenominator0	INT16	rw	Y	INT16	16384
0x3207	3	AnaInpScaOffset0	INT16	rw	Y	INT16	0

6.3.5.3 Object 0x3204: Actual value

Scaled actual value of the analog input 0.

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

6.3.6 Analog input 1

Depending on the ordered servo valve, the analog input 1 is available.

6.3.6.1 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	Per-sistence	Value range	Default
0x3208	0	AnaInpType1	INT8	rw	Y	1…12	2

Value description

➪ Table 39, page 85

6.3.6.2 Object 0x320F: Scaling

The analog input 1 can be scaled directly. If a pressure transducer is connected to analog input 1, the scaling must be selected so that 100 % or 16384 increments correspond to the ordered nominal pressure of the servo valve.

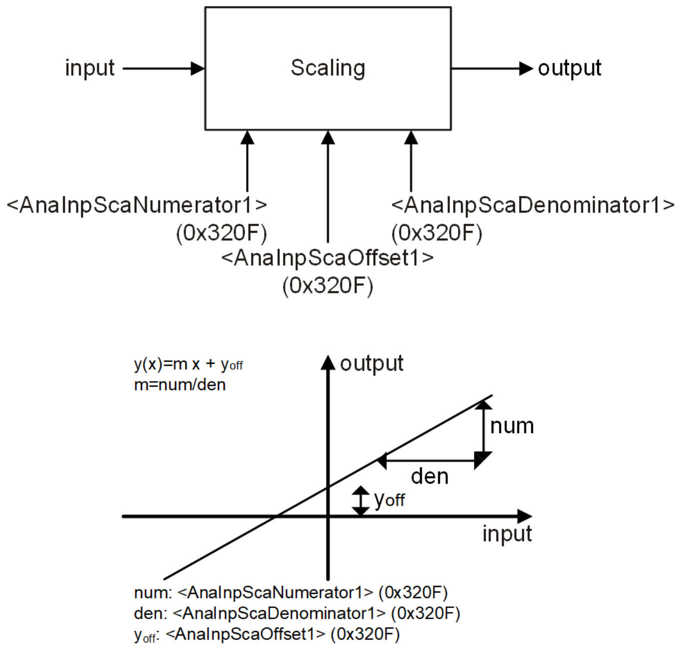


Figure 27: Analog input 1 scaling

AnalogueInput1							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x320F	1	AnaInpScaNumerator1	INT16	rw	Y	INT16	16384
0x320F	2	AnaInpScaDenominator1	INT16	rw	Y	INT16	16384
0x320F	3	AnaInpScaOffset1	INT16	rw	Y	INT16	0

6.3.6.3 Object 0x320C: Actual value

Scaled actual value of the analog input 1.

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

6.3.7 Analog input 2

Depending on the ordered servo valve, the analog input 2 is available.

6.3.7.1 Object 0x3210: Input type

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

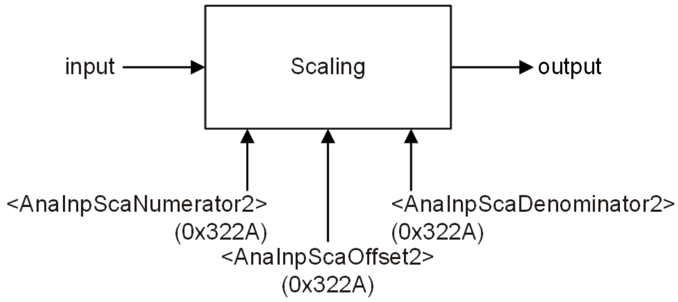
AnalogInput2							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3210	0	AnaInpType2	INT8	rw	Y	1…12	2

Value description

➪ Table 39, page 85

6.3.7.2 Object 0x322A: Scaling

The analog input 2 can be scaled directly. If a pressure transducer is connected to analog input 2, the scaling must be selected so that 100 % or 16384 increments correspond to the ordered nominal pressure of the servo valve.



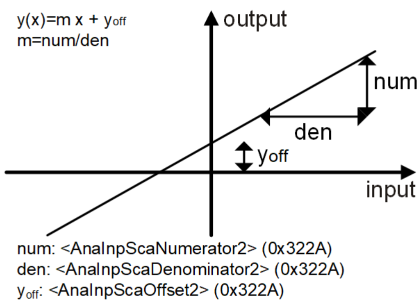


Figure 28: Analog input 2 scaling

AnalogueInput2							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x322A	1	AnaInpScaNumerator2	INT16	rw	Y	INT16	16384
0x322A	2	AnaInpScaDenominator2	INT16	rw	Y	INT16	16384
0x322A	3	AnaInpScaOffset2	INT16	rw	Y	INT16	0

6.3.7.3 Object 0x3214: Actual value

Scaled actual value of the analog input 2.

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

Value description

➪ Table 39, page 85

6.3.8 Analog input 3

Depending on the ordered servo valve, the analog input 3 is available.

6.3.8.1 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	Per-sistence	Value range	Default
0x3218	0	AnaInpType3	INT8	rw	Y	1…12	2

Value description

➪ Table 39, page 85

6.3.8.2 Object 0x321F: Scaling

The analog input 3 can be scaled directly. If a pressure transducer is connected to analog input 3, the scaling must be selected so that 100 % or 16384 increments correspond to the ordered nominal pressure of the servo valve.

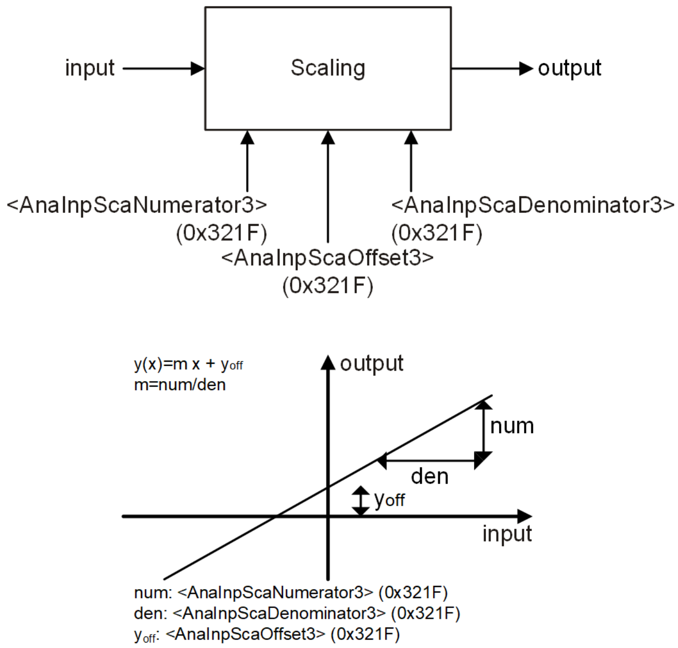


Figure 29: Analog input 3 scaling

AnalogueInput3							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x321F	1	AnaInpScaNumerator3	INT16	rw	Y	INT16	16384
0x321F	2	AnaInpScaDenominator3	INT16	rw	Y	INT16	16384
0x321F	3	AnaInpScaOffset3	INT16	rw	Y	INT16	0

6.3.8.3 Object 0x321C: Actual value

Scaled actual value of the analog input 3.

AnalogInput3							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x321C	0	AnalnpActualValue3	INT16	ro	-	INT16	None

6.3.9 Analog input 4

Depending on the ordered servo valve, the analog input 4 is available.

6.3.9.1 Object 0x3220: Input type

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

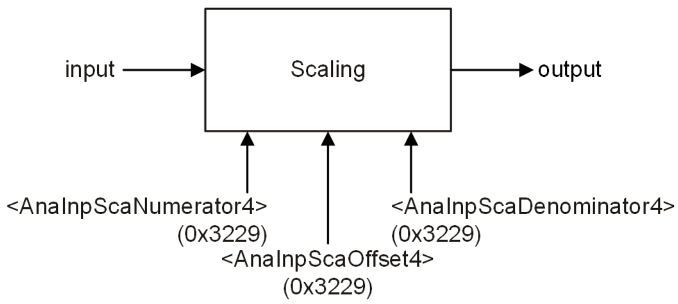
AnalogInput4							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3220	0	AnalnpType4	INT8	rw	Y	1…12	2

Value description

⇨ Table 39, page 85

6.3.9.2 Object 0x3229: Scaling

The analog input 4 can be scaled directly. If a pressure transducer is connected to analog input 4, the scaling must be selected so that 100 % or 16384 increments correspond to the ordered nominal pressure of the servo valve.



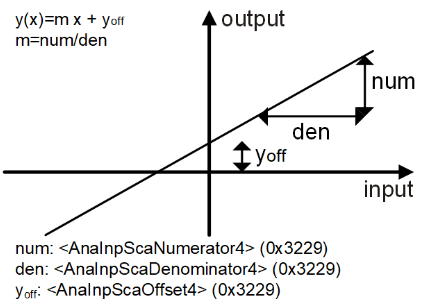


Figure 30: Analog input 4 scaling

AnalogueInput4							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3229	1	AnaInpScaNumerator4	INT16	rw	Y	INT16	16384
0x3229	2	AnaInpScaDenominator4	INT16	rw	Y	INT16	16384
0x3229	3	AnaInpScaOffset4	INT16	rw	Y	INT16	0

6.3.9.3 Object 0x3224: Actual value

Scaled actual value of the analog input 4.

AnalogInput4							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3224	0	AnaInpActualValue4	INT16	ro	-	INT16	None

6.3.10 Internal pressure transducer offset correction

If the integrated pressure sensor module has been ordered, an automatic offset adjustment is available for the internal pressure transducers at port A, B, P. For this purpose, transducers <PressureSensorCorrectionOffset0…2> can be determined.

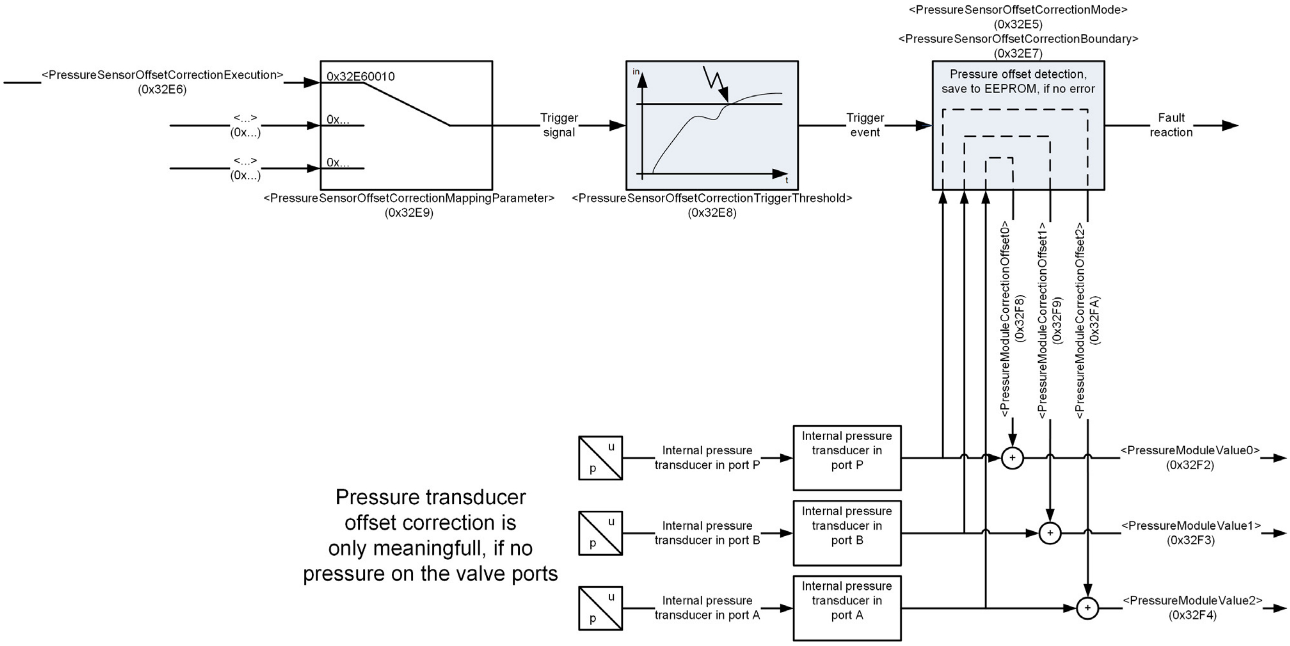


Figure 31: Internal pressure transducer offset correction

The automatic calibration of the pressure transducers can be performed for one or all pressure transducers at the same time. The calibration can be started with a freely definable trigger signal. The trigger signal is selected with the mapping parameter <PressureSensorOffsetCorrectionMappingParameter>. The trigger threshold <PressureSensorOffsetCorrectionTriggerThreshold> determines the value from which the trigger calibration is triggered. Which pressure sensor is to be calibrated is defined with the parameter <PressureSensorOffsetCorrectionMode>. To be able to detect a malfunction, a maximum possible correction value <PressureSensorOffsetCorrectionBoundary> is defined. If this value is exceeded, fault code 125 (0x7D) is set. Otherwise the determined offset is inverted and then stored in the EEPROM. After successful correction, the inverted pressure sensor offset is added to the measured value.

➪ Chapter "8.1 Fault reaction", page 261

6.3.10.1 Object 0x32E6: Offset correction execution

This parameter can be mapped to <PressureSensorOffsetCorrectionMappingParameter> and then serves as a signal to trigger the pressure offset correction. Value is set automatically to zero after determination of offset correction is finished.

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x32E6	0	PressureSensorOffsetCorrectionExecution	INT8	ro	N	0…1	0

6.3.10.2 Object 0x32E9: Offset correction mapping parameter

This parameter defines source of the trigger signal where the trigger are coming from. The trigger signal is defined by a parameter index, sub-index and length.

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x32E9	0	PressureSensorOffsetCorrectionMappingParameter	UINT32	rw	Y	UINT32	0x32E60010

Value description

Table 40: Possible values of parameter <PressureSensorOffsetCorrectionMappingParameter> (0x32E9)

<PressureSensorOffsetCorrectionMappingParameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length: 0x10
Default	0x32	0xE6	0x00	0x10

This reference 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 0x32E60010, which refers to the <PressureSensorOffsetCorrectionExecution> (0x32E6), sub-index 0x00 with a length of 16 bit (16 = 0x10).

6.3.10.3 Object 0x32E8: Offset correction trigger threshold

This parameter defines the trigger level that the trigger signal must reach in order to start the automatic pressure transducer calibration. For digital signals the parameter must be set to 1.

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x32E8	0	PressureSensorOffsetCorrectionTriggerThreshold	INT16	rw	Y	INT16	1

6.3.10.4 Object 0x32E7: Offset correction boundary

To avoid incorrect offset corrections, e.g. by a hydraulic system that is not pressureless, the pressure signal correction is limited to the value of this parameter. The unit is given as a percentage of the value range. This means that for a 160 bar (ordered nominal pressure) servo valve 1.5 % corresponds to 2.4 bar. If the determined offset exceeds the value of this parameter in percent, the fault code 125 (0x7D) is set.

➪ Chapter "8.1 Fault reaction", page 261

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x32E7	0	PressureSensorOffsetCorrectionBoundary	REAL32	ro	-	0.00… 100.00	1.50

6.3.10.5 Object 0x32E5: Offset correction mode

It can be selected whether the pressure transducer in connection A, B, P or whether all pressure transducers at the same time should be calibrated automatically.

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x32E5	0	PressureSensorOffsetCorrectionMode	INT8	rw	Y	0…3	None

Value description

Table 41: Possible values of parameter <PressureSensorOffsetCorrectionMode> (0x32E5)

<PressureSensorOffsetCorrectionMode>	Description
0	Automatic offset correction of all internal pressure transducers
1	Automatic offset correction of the internal pressure transducer in port P
2	Automatic offset correction of the internal pressure transducer in port B
3	Automatic offset correction of the internal pressure transducer in port A

6.3.10.6 Object 0x32F8: Correction offset 0

This read only parameter contains the correction value that was determined by the automatic pressure offset correction for the pressure sensor at port P.

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x32F8	0	PressureSensorCorrectionOffset0	INT16	ro	-	INT16	0

6.3.10.7 Object 0x32F9: Correction offset 1

This read only parameter contains the correction value that was determined by the automatic pressure offset correction for the pressure sensor at port B.

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x32F9	0	PressureSensorCorrectionOffset1	INT16	ro	-	INT16	0

6.3.10.8 Object 0x32FA: Correction offset 2

This read only parameter contains the correction value that was determined by the automatic pressure offset correction for the pressure sensor at port A.

PressureModule							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x32FA	0	PressureSensorCorrectionOffset2	INT16	ro	-	INT16	0

6.4 Analog outputs

Most of the servo valves are delivered and operated in the configuration shown below. Analog output 0 is always present. The presence of analog output 1 depends on the servo valve configuration ordered. The type of electrical interface depends on the servo valve configuration. All digital to analog converters have the same resolution of 12 bit. There are three possible hardware interface types available:

- 4…20 mA
- 0…10 V or −10…+10 V (referenced to supply ground)
- 0…10 V or −10…+10 V (referenced to supply ground with 10 kOhm resistor)

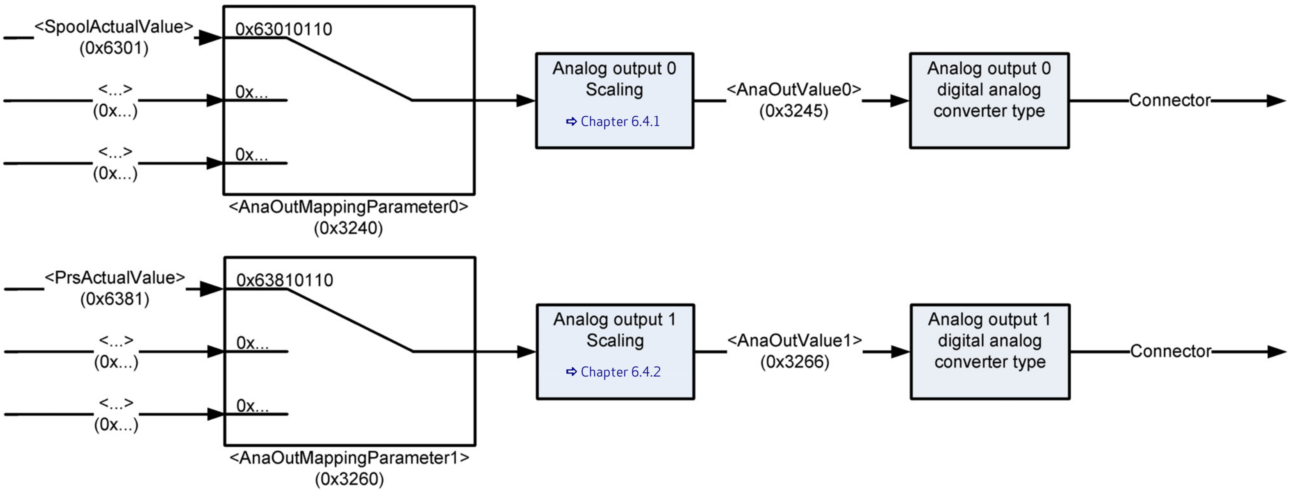
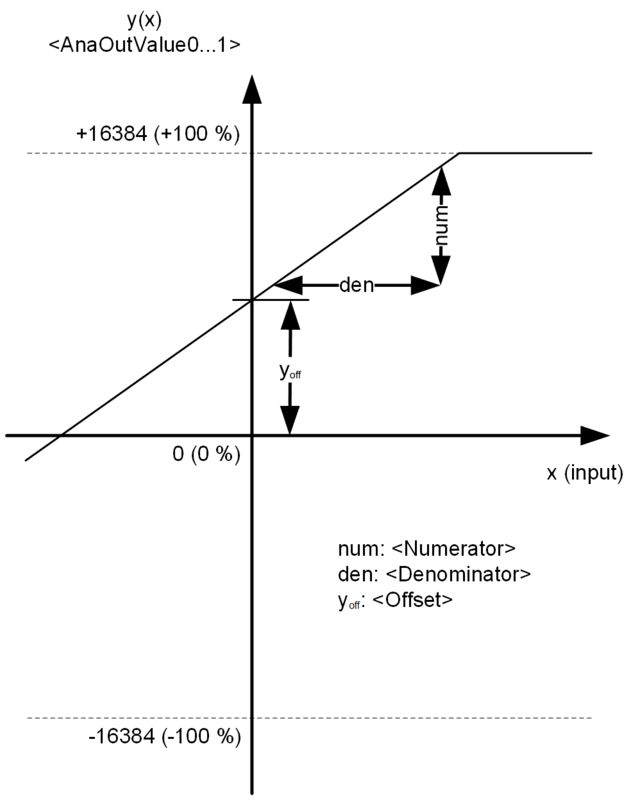


Figure 32: Analog outputs

➪ Document CA63420-001 "User Manual Electrical Interfaces"

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



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

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

Figure 33: Analog output scaling

6.4.1 Analog output 0

By default, the output 0 is used to provide the current spool position value. The user can also use this output to output any arbitrary value. This is done by mapping the corresponding parameter by means of the parameter <AnaOutMappingParameter0> (0x3240) to this arbitrary value. The value itself can be scaled with the parameters <AnaOutScaNumerator0> (0x3244), <AnaOutScaDenominator0> (0x3244) and <AnaOutScaOffset0> (0x3244). The scaled value can be queried via the parameter <AnaOutValue0> (0x3245).

6.4.1.1 Object 0x3240: Mapping parameter

This parameter defines the source parameter of the 16 bit output signal. The output signal is referenced by the corresponding parameter index and sub-index.

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

Value description

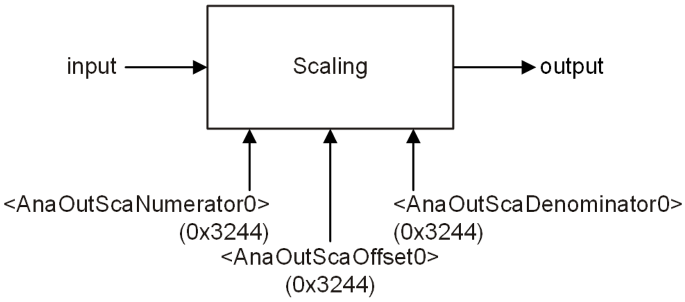
Table 42: Possible values of parameter <AnaOutMappingParameter0> (0x3240)

<AnaOutMappingParameter0>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length: 0x10 or 0x20
Default	0x63	0x01	0x01	0x10

This reference 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 or 0x20.

The default value is 0x63010110, which refers to the <SpoolActualValue> (0x6301), sub-index 0x01 with a length of 16 bit (16 = 0x10).

6.4.1.2 Object 0x3244: Scaling



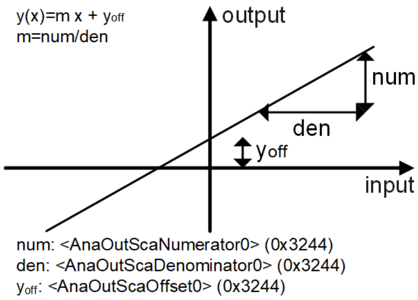


Figure 34: Analog output 0 scaling

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3244	1	AnaOutScaNumerator0	INT16	rw	Y	INT16	16384
0x3244	2	AnaOutScaDenomina-tor0	INT16	rw	Y	INT16	16384
0x3244	3	AnaOutScaOffset0	INT16	rw	Y	INT16	0

6.4.1.3 Object 0x3245: Actual value

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

6.4.1.4 Object 0x3243: Type

The electrical interface type depends on the ordered servo valve configuration. In case the current variant has been ordered, it is reasonable to be switched over as in table 43 described.

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

Value description

Table 43: Possible values of parameter <AnaOutType0> (0x3243)

<AnaOutType0>	<ActualOutputValue0> (0x3245) range	Output signal range
0	–16384…16384	4…20 mA / –10…+10 V (depending on hardware version)
1	0…16384	4…20 mA / –10…+10 V (depending on hardware version)

6.4.2 Analog output 1

By default, the output 1 is used to provide the current pressure value. The user can also use this output to output any arbitrary value. This is done by mapping the corresponding parameter by means of the parameter <AnaOutMappingParameter1> (0x3260) to this arbitrary value. The value itself can be scaled with the parameters <AnaOutScaNumerator1> (0x3265), <AnaOutScaDenominator1> (0x3265) and <AnaOutScaOffset1> (0x3265). The scaled value can be queried via the parameter <AnaOutValue1> (0x3266).

6.4.2.1 Object 0x3260: Mapping parameter

This parameter defines source of the 16 bit output signal where the output are coming from. The output signal is defined by a parameter index, sub-index.

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

Value description

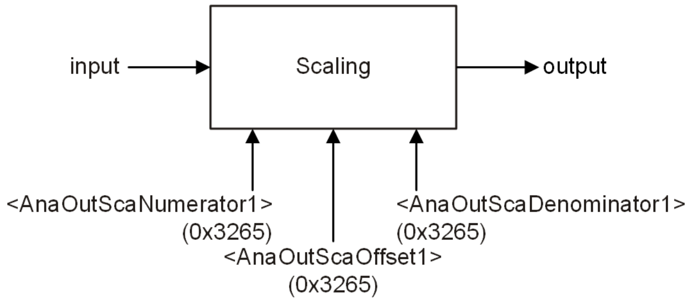
Table 44: Possible values of parameter <AnaOutMappingParameter1> (0x3260)

<AnaOutMappingParameter1>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length: 0x10 or 0x20
Default	0x63	0x81	0x01	0x10

This reference 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 or 0x20.

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

6.4.2.2 Object 0x3265: Scaling



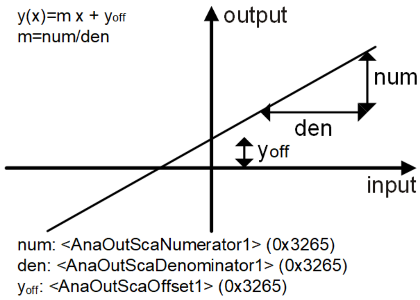


Figure 35: Analog output 1 scaling

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

6.4.2.3 Object 0x3266: Actual value

AnalogOutput1							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3266	0	AnaOutValue1	INT16	ro	-	Depending on <AnaOutType1> (0x3263)	None

6.4.2.4 Object 0x3263: Type

The electrical interface type depends on the ordered servo valve configuration. In case the current variant has been ordered, it is reasonable to be switched over as in table 43 described.

AnalogOutput1							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3263	0	AnaOutType1	UINT8	rw	Y	0…1	0

Value description

Table 45: Possible values of parameter <AnaOutType1> (0x3263)

<AnaOutType1>	**<AnaOutValue1> (0x3266) range**	**Output signal range**
0	–16384…16384	4…20 mA / –10…+10 V (depending on hardware version)
1	0…16384	4…20 mA / –10…+10 V (depending on hardware version)

6.5 Digital inputs

Depending on the servo valve configuration ordered, there are no digital inputs, a digital enable input and additionally a second input. An internal two-channel monitoring of the enable signal (digital input 0) can also be ordered to increase machine safety. The second input can be mapped to at least one customer accessible mapping parameter via the non-customer accessible parameter <MonitoringEnable2> (0x2542).
The status of the enable signal can be queried via the parameter <ManufacturerStatusRegister> (0x1002).

➪ Chapter "5.2.1 Object 0x1002: Manufacturer Status Register", page 60

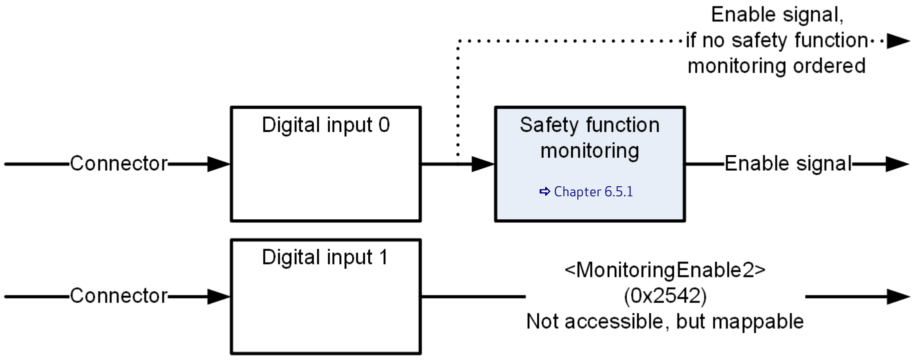


Figure 36: Digital inputs

➪ Document CA63420-001 "User Manual Electrical Interfaces"

6.5.1 Safety function monitoring

Depending on the ordered servo valve configuration the safety function monitoring is active or not. If safety function monitoring is active, the enable signal (digital input 0) is routed through two separate hardware channels and the signals are monitored.

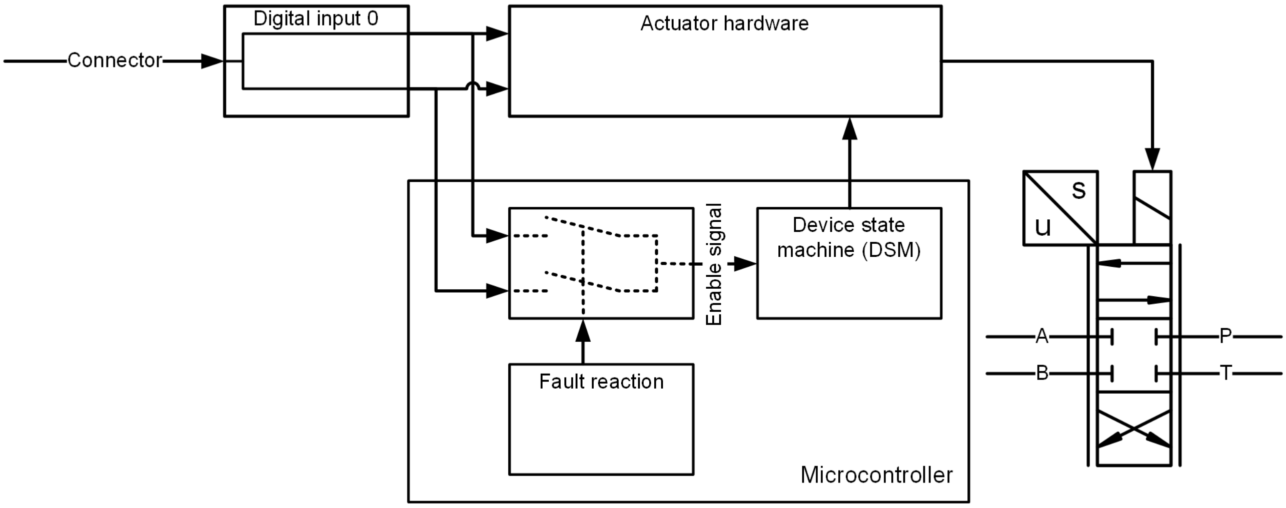


Figure 37: Safety function monitoring

When the monitoring of the safety function is active, the error reactions of the following errors are configured:

- Different levels on the hardware redundant digital input lines lead to an error reaction with fault code 124 (0x7C) and with the fault response 'FAULT_DISABLED'.
- Different levels on the hardware-redundant H-bridge controls lead to an error response with fault code 45 (0x2D) and with the fault response 'FAULT_DISABLED'.
- Overcurrent in the solenoid coils leads to a fault response with fault code 16 (0x10) and with the fault response 'FAULT_DISABLED'.
- Excessive system deviation of the current controller leads to a fault response with fault code 66 (0x42) and with the fault response 'FAULT_DISABLED'.

If the safety function monitoring is active and one of the four above-mentioned errors occurs, the device state machine (DSM) changes to the 'FAULT_DISABLED' state and the power supply for the actuator is switched off.

➪ Chapter "5.1.2.4.1 DSM state transitions depending on the #EnableSignal#", page 55

Likewise, the safe state is no longer output at the digital output with a high-level signal, as the safe function of the valve can no longer be guaranteed.

➪ Chapter "6.6 Digital outputs", page 103

!	The safety function is not certified. However, the safety concept has been implemented in accordance with internal specifications and in coordination with a certification authority. A valve with a safety function alone does not achieve a safe machine function, but the degree of safety (performance level in ISO13849) is determined by the safety concept of the entire machine function. No personal safety can be guaranteed with the safety concept described in this chapter. With the safety logic of a valve alone, only the requirements for machine safety can be achieved. For personal safety, two redundant valves must be installed and the safe status must be returned to the safety control via the digital outputs

6.6 Digital outputs

The number of available digital outputs and the assignment of the functionalities to the outputs depends on the ordered valve configuration.

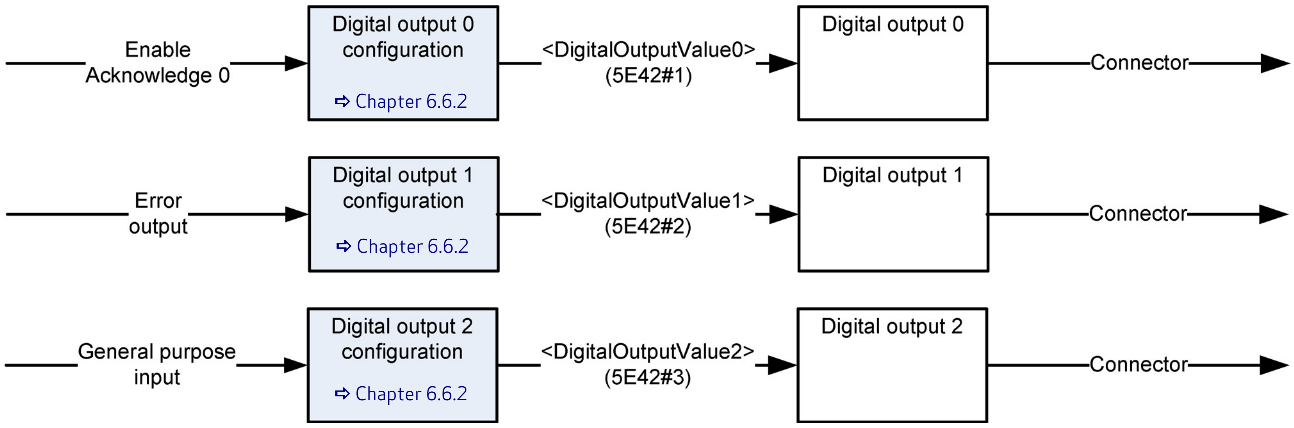


Figure 38: Digital outputs

➪ Document CA63420-001 "User Manual Electrical Interfaces"

6.6.1 Object 0x5E42: Digital output values

The <DigitalOutpuValue0…2> controls the state of the digital outputs in case the <DigitalOutputConfiguration0…2> is set to 0.

DigitalOutput							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5E42	1	DigitalOutputValue0	INT8	rw	-	0…1	0
0x5E42	2	DigitalOutputValue1	INT8	rw	-	0…1	0
0x5E42	3	DigitalOutputValue2	INT8	rw	-	0…1	0

6.6.2 Object 0x5E41: Digital output configuration

This parameter can be used to read the configuration of the digital outputs. Depending on the ordered servo valve the configuration is set at the factory.

DigitalOutput							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5E41	1	DigitalOutputConfiguration0	INT8	ro	-	0…5	3
0x5E41	2	DigitalOutputConfiguration1	INT8	ro	-	0…5	0
0x5E41	3	DigitalOutputConfiguration2	INT8	ro	-	0…5	3

Value description

Table 46: Possible values of parameter <DigitalOutputConfiguration0…2> (0x5E41)

<DigitalOutputConfiguration0…2>	**Description**
0	The servo valve's digital outputs are controlled by the parameter <DigitalOutpuValue0…2> (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 "7.9.13 Failsafe monitoring", page 239
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 "5.1.2.5 Error output pin", page 55
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 "7.9 Monitoring", page 226
5	The digital output gets high (24 V) if the safe state is reached. Low (0 V) on digital output means unsafe state. The safe state is reached, if the enable signal is Low (0 V) and none of the safety function relevant faults occurred. ➪ Chapter "6.5.1 Safety function monitoring", page 102

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!

6.6.3 Object 0x5E44: Digital output state

The <DigitalOutputState0…2> (0x5E44) shows the state of the digital outputs, independent of the chosen configuration <DigitalOutputConfiguration0…2> (0x5E41).

DigitalOutput							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5E44	1	DigitalOutputState0	UINT8	ro	-	0…1	None
0x5E44	2	DigitalOutputState1	UINT8	ro	-	0…1	None
0x5E44	2	DigitalOutputState2	UINT8	ro	-	0…1	None

6.7 Local CAN

Depending on the valve model, a local CANopen interface is available. This interface is intended to be used to connect the Moog Valve and Pump Service Tool or to be used as a local sensor interface. The Local CAN has implemented the same CANopen protocols as the Fieldbus CAN. Furthermore Local CAN has the same number and access rights of the valve parameters as the Fieldbus CAN.

It is not galvanic isolated from the valves electronic! For bus termination, a valve internal resistor can be activated.

6.7.1 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 profile for the hydraulic valves CiA 408 is built on top of this basic profile.

6.7.2 Device profiles

The device profiles describe the application parameters and the functional behavior of the devices including the device class-specific state machines.

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 fieldbus in a standardized, fieldbus independent format across manufacturers. This profile is implemented in all Moog servo valves with fieldbus interface.

The CiA organization transformed this 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". Other device profiles are for example generic I/O modules (CiA 401), drives and motion control (CiA 402).

6.7.3 CANopen 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 because such functionalities are not required in a real-time fieldbus system. The CANopen communication concept can be described similar to the ISO/OSI reference model.

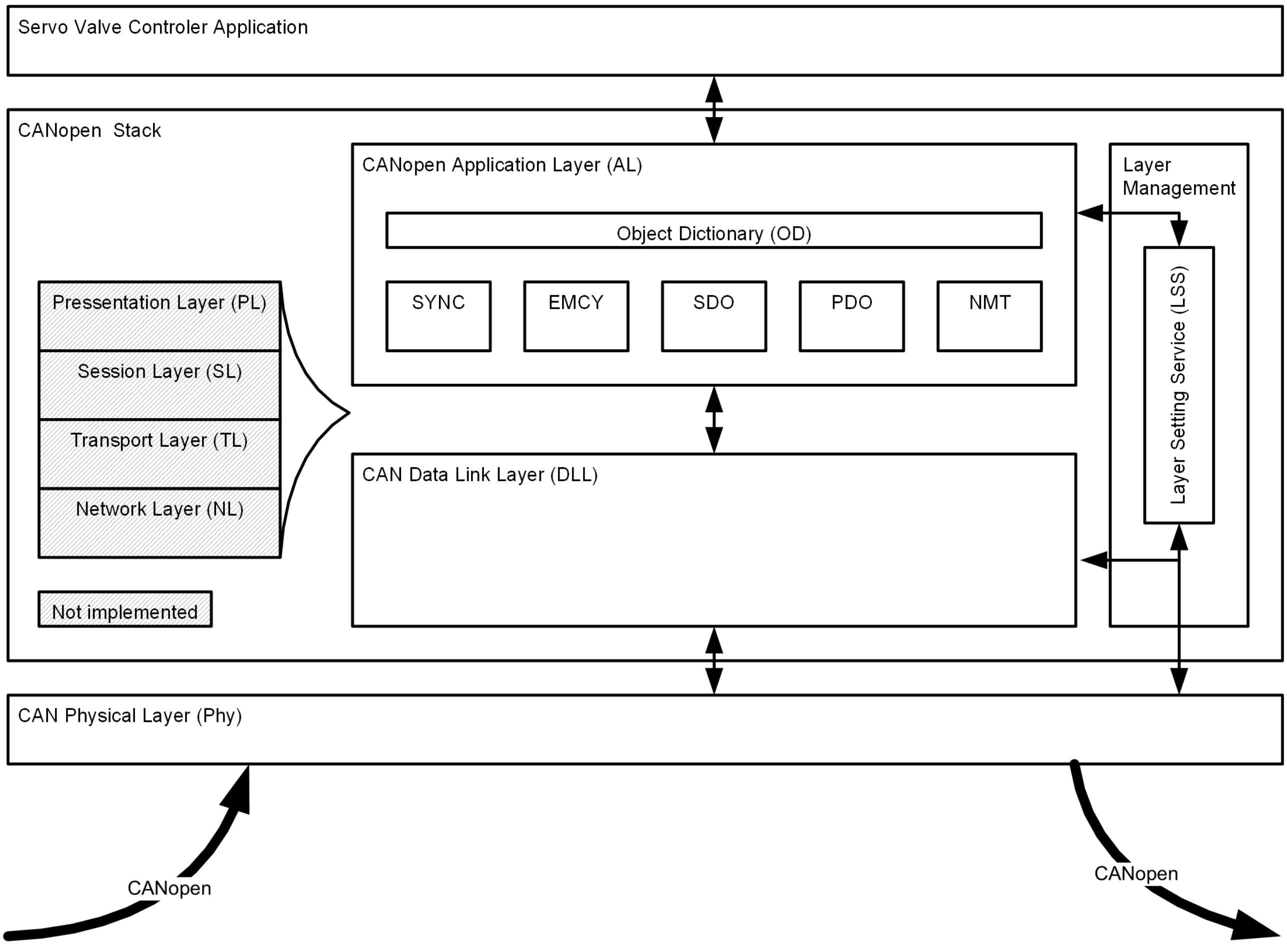


Figure 39: CANopen reference model

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.

Table 47: CANopen reference model

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)

6.7.4 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.

6.7.4.1 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.

6.7.4.2 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 "Object dictionary" you will find a form like the following one in which the attributes are explained in detail.
➪ Chapter "10 Object dictionary", page 290

Index	Sub-index	PDO-mapping	Short name	Specification	Block, object and parameter name	Data type	Access	Per-sistence	Value range	Default

Table 48: Fieldbus independent attributes

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. **CiA 301 data type** – **Short name** – **Description** INTEGERn – INTn – Signed integer with n bits UNSIGNEDn – UINTn – Unsigned integer with n bits REAL32 – REAL32 – Floating point with 32 bit according CiA 301 VISIBLE_STRING – STRING(n) – String of n ASCII characters DOMAIN – 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 parameter.
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 287
Specification	Related (fieldbus) standard defining the parameter. Possible entries: CiA 301 – Parameters correspond to CiA 301. 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.

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

6.7.4.3 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.

Table 49: Unit representation

Name of unit	International symbol	Notation index (hex)	Notation index (dec)
none	dimensionless or iR	0x00	0
meter	m	0x01	1
second	s	0x03	3
hertz	Hz	0x20	32
liter	l or L	0x44	68
minute (time)	min	0x47	71
hour	h	0x48	72
day	d	0x49	73
year	a	0x4A	74
bar	bar	0x4E	78
meter per square second	m/s^2	0x55	85

Table 50: Prefix representation

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

6.7.5 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.

Table 51: Structure of the CANopen object dictionary (OD)

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	Note: Communication profile area used for the fieldbus CAN. For configuring the local CAN interface, the objects 0x5000 and following are used.	CiA 301 / IEC 61158-5-12
0x2000…0x5FFF	Manufacturer-specific area	Moog DCV
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	

6.7.6 Electronic data sheet (EDS)

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. The digital servo valve EDS file is provided by Moog and can be downloaded from the Moog website
http://www.moogsoftwaredownload.com.

6.7.7 CANopen communication protocols

CANopen communication protocols are classified as follows:

- Real-time data are transferred with the process data object (PDO) protocol.
- 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.

Table 52: CANopen communication objects

Protocol	COB-ID	Description	SDO objects used for protocol configuration	Reference	Chapter
NMT	0x000	Network Management (NMT) protocol (Broadcast)		CiA 301	➪ Chapter "2.10 Network management (NMT) state machine", page 17
NMT	0x000+Node-ID	Network Management (NMT) protocol		CiA 301	
SYNC	0x080	Synchronization (Broadcast) protocol	0x1005	CiA 301	➪ Chapter "2.18 Synchronization (SYNC)", page 38
EMCY	0x080+Node-ID	Emergency protocol	0x1014	CiA 301	➪ Chapter "2.19 Emergency (EMCY)", page 40
TPDO	0x180+Node-ID	1st Transmit PDO protocol	0x1800, 0x1A00	CiA 301	➪ Chapter "2.13 Process data object (PDO)", page 23 ➪ Chapter "2.15 Transmit process data object (TPDO)", page 30
RPDO	0x200+Node-ID	1st Receive PDO protocol	0x1400, 0x1600	CiA 301	
TPDO	0x280+Node-ID	2nd Transmit PDO protocol	0x1801, 0x1A01	CiA 301	
RPDO	0x300+Node-ID	2nd Receive PDO protocol	0x1401, 0x1601	CiA 301	
TPDO	0x380+Node-ID	3rd Transmit PDO protocol	0x1802, 0x1A02	CiA 301	
RPDO	0x400+Node-ID	3rd Receive PDO protocol	0x1402, 0x1602	CiA 301	
TPDO	0x480+Node-ID	4th Transmit PDO protocol	0x1803, 0x1A03	CiA 301	
RPDO	0x500+Node-ID	4th Receive PDO protocol	0x1403, 0x1603	CiA 301	
SDO	0x580+Node-ID	Transmit SDO protocol	0x1200	CiA 301	➪ Chapter "2.16 Service data object (SDO)", page 37
SDO	0x600+Node-ID	Receive SDO protocol	0x1200	CiA 301	
NMT	0x700+Node-ID	Network Management (NMT) error control protocol (Bootup, Node Guarding, Hearbeat)	0x100C, 0x100D, 0x1016, 0x1017	CiA 301	➪ Chapter "2.10 Network management (NMT) state machine", page 17 ➪ Chapter "2.12 Network management (NMT) node guarding", page 22 ➪ Chapter "2.11 Network management (NMT) heartbeat", page 21
LSS	0x7E4	Transmit Layer Setting Services (LSS) protocol	0x1018	CiA 305	➪ Chapter "2.8 Bit rate and Node-ID configuration using Layer Setting Services (LSS)", page 15
LSS	0x7E5	Receive Layer Setting Services (LSS) protocol			

6.7.8 Bit rate and Node-ID configuration using Layer setting services (LSS)

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:

Table 53: CANopen bit rates

<Bitrate>	
Bit rate	**Maximum cable length**
1000 kBit/s	< 25 m
500 kBit/s	< 100 m
250 kBit/s	< 250 m
125 kBit/s	< 500 m
100 kBit/s - not supported	-
50 kBit/s	< 1000 m
20 kBit/s	< 2500 m

➪ For details, see document CA63420 001 "User Manual Electrical Interfaces"

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 valves local CAN interface can be configured by an LSS master.

To configure the local CAN interface with the LSS Service, the CAN bus must be connected with this local interface.

The implemented LSS service provides four methods to change the Node-ID or bit rate of the servo valve:

1. LSS configuration service
   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. LSS inquiry service
   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 (0x1018) is worldwide unique and will be programmed during production. Information about the identification object (0x1018) is available from the name plate (LSS) of the device or can be inquired by using SDO.
3. LSS identification services
   The third method corresponds to the second with the possibility to find devices that match a certain VendorID and product code in the identification object (0x1018). This allows newly installed bus devices with changed revision number and serial number to be found.
4. LSS fastscan service
   As the LSS identify remote slave service, the LSS fastscan service may be used to scan for unconfigured LSS slaves. The value range of the active node-ID is from 0x01 to 0x7F. In addition to this range, the value range for pending Node-ID as well as the persistent Node-ID includes the value 0xFF which indicates an invalid setting.

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 <IdentityObject> (0x1018) or LSS address (VendorId / ProductCode / RevisionNumber / SerialNumber) can be found on the name plate of the servo valve. For more details about the LSS service see CiA 305.

➪ Chapter "4.1.5 Object 0x1018: Identity object", page 45

Figure 40: Nameplate of the device with identification object address

6.7.9 Bit rate and Node-ID configuration using service data object (SDO)

In case a network master does not support the LSS services, the bit rate and the Node-ID in the servo valve local CAN interface 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 an NMT-reset communication are initiated. That makes sure that the CAN communication will not get interrupted by changing these basic settings.
➪ Chapter "2.10 Network management (NMT) state machine", page 17

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.

6.7.9.1 Object 0x500B: Actual module identifier (Node-ID)

This parameter shows the currently used Node-ID of the local CAN fieldbus.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x500B	0	LocalCanActualModuleIdentifier	UINT32	ro	-	1...127	None

6.7.9.2 Object 0x5102: Module identifier (Node-ID)

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

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5102	0	LocalCanModuleIdentifier	UINT8	rw	Y	1…127	127

6.7.9.3 Object 0x5103: Bit rate

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

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5103	0	LocalCanBitrate	UINT32	rw	Y	10000… 1000000	500000

Value description

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

Table 54: CANopen bit rates

<LocalCanBitrate>		
Bit rate	**Maximum cable length**	**Parameter value**
1000 kBit/s	< 25 m	1000000
500 kBit/s	< 100 m	500000
250 kBit/s	< 250 m	250000
125 kBit/s	< 500 m	125000
100 kBit/s - not supported	-	-
50 kBit/s	< 1000 m	50000
20 kBit/s	< 2500 m	20000
10 kBit/s	< 5000 m	10000

6.7.10 Network management (NMT) state machine

The CANopen NMT state machine is used to control the network communication of the local CAN interface. 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 CANopen network management (NMT) state machine must not be mistaken with the device state machine.

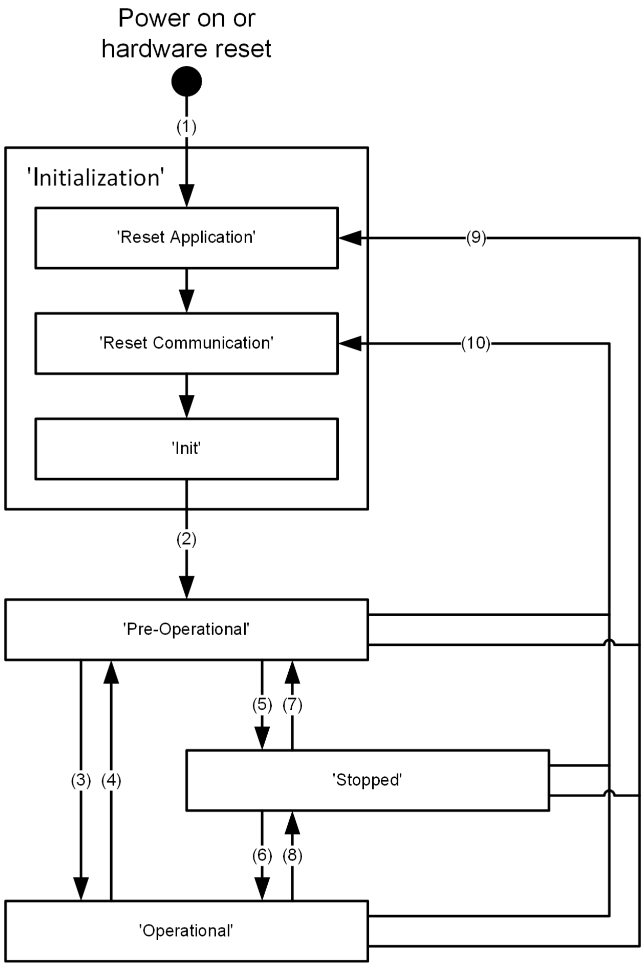


Figure 41: Network management (NMT) state machine

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. On transition to the 'Pre-Operational' state, a boot-up message is sent to signal successful booting to the master. 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. For further information on the node control protocol see CiA 301.

NMT state transitions are caused by

- Reception of an NMT node control command from the CANopen bus
- Hardware reset, or
- Node control services locally initiated by application events.

Table 55: NMT state transitions

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".

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

Table 56: NMT states

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

6.7.10.1 Object 0x5104: Network management (NMT) state machine status

This parameter contains the network management (NMT) state machine status in bit coded form.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5104	0	LocalCANNmtState	UINT16	ro	-	UINT16	None

Value description

Table 57: Object 0x5104: Network Management (NMT) state machine status

<NmtState>		
Status (hex)	**Status (dec)**	**Status**
0x0000	0	Invalid State
0x0001	1	State Power On or Hardware Reset
0x0002	2	Reset Application
0x0004	4	Reset Communication
0x0008	8	Init
0x0010	16	Pre-Operational
0x0020	32	Stopped
0x0040	64	Operational
0x8000	32768	Severe Error State

6.7.10.2 Object 0x5105: Network management (NMT) state machine control

This parameter is used to control the Network management (NMT) state machine. The possible status change depends on the current status.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5105	0	LocalCANNmtSetState	UINT16	rw	N	1…130	None

Value description
⇨ Table 57, page 117

6.7.10.3 Object 0x5029: Network management (NMT) error behavior

If a serious CANopen device failure is detected in NMT state 'Operational', the CANopen device shall enter by default autonomously the NMT state 'Pre-operational'. The error behavior of the NMT can be changed via this parameter to NMT state 'Stopped' or remain in the current NMT state.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5029	0	LocalCANErrorBehavior	UINT8	rw	Y	UINIT8	0

Value description

<LocalCANErrorBehavior>	
0	Change to NMT state 'Pre-Operational' (only if currently in NMT state 'Operational')
1	No change of the NMT state
2	Change to NMT state 'Stopped'
3…256	No change of the NMT state

6.7.10.4 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.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5B02	0	LocalCanStartRemoteNode	UINT8	rw	Y	0…127	0

6.7.11 Network management (NMT) heartbeat

The Heartbeat Protocol consists of two separate services:

1. The monitoring of a periodically received heartbeat message, called consumer heartbeat. The Node-ID of the heartbeat transmitter and the time window until the next heartbeat message must arrive is set by the parameter <LocalCanConsumerHeartbeatTime> (0x5016). If the heartbeat transmitter does not respond within the specified time, error code 110 (0x6E) is triggered.
2. Periodically sending a heartbeat message, called producer heartbeat. For this purpose, the interval between two heartbeat messages can be set with the parameter <LocalCanProducerHeartbeatTime> (0x5017). The heartbeat protocol starts on the transition from the NMT state 'Initialization' to the NMT state 'Pre-Operational'. In this case the boot-up message is regarded as first heartbeat message. With the heartbeat message the servo valve NMT state is transmitted.

> It is not allowed to use both error control mechanisms guarding protocol and heartbeat protocol on one NMT slave at the same time.

6.7.11.1 Object 0x1016: Consumer heartbeat time

The parameter <ConsumerHeartbeatTime> (0x5016) defines the expected heartbeat cycle time. It additionally defines the node-ID of the heartbeat producer to be monitored.

> The consumer heartbeat time should be higher than the corresponding producer heartbeat time.

To enable the heartbeat message, a time greater zero must be set and a valid node-ID (between 1…127) must be configured. If the heartbeat time is 0 or the node-ID is 0 or greater than 127, the corresponding object entry shall be not used. The heartbeat time shall be given in multiples of 1 ms.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5016	1	LocalCanConsumerHeartbeatTime	UINT32	rw	Y	UINT32	0

Value description

<LocalCanConsumerHeartbeatTime>			
Bit	31…24	23…16	15…0
Description	Reserved	Node-Id	Heartbeat time in ms

6.7.11.2 Object 0x5017: Producer heartbeat time

The parameter <LocalCanProducerHeartbeatTime> (0x5017) specifies the time in ms at which the servo valve sends a heartbeat message. The value 0 disables the producer heartbeat.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5017	0	LocalCanProducerHeartbeatTime	UINT16	rw	Y	UINT16	0

6.7.12 Network management (NMT) node guarding

The node guarding is the periodical monitoring of certain network nodes. Each node can be checked by the NMT master with a certain period <LocalCanGuardTime> (0x500C). A second parameter <LocalCanLifeTimeFactor> (0x500D) 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 transmit 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 (0x6E) 'CAN life guard error or heartbeat error' is triggered is calculated as follows:
Monitoring time = <LocalCanGuardTime> (0x500C) · <LocalCanLifeTimeFactor> (0x500D)

> It is not allowed to use both error control mechanisms guarding protocol and heartbeat protocol on one NMT slave at the same time.

If the network master supports the Heartbeat Protocol, this should be preferred to Node Guarding, as it allows better monitoring. The guarding protocol is still supported for compatibility reasons.

6.7.12.1 Object 0x500C: Guard time

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

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x500C	0	GuardTime	UINT16	rw	Y	UINT16	0

6.7.12.2 Object 0x500D: Life time factor

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

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x500D	0	LifeTimeFactor	UINT8	rw	Y	UINT8	1

6.7.13 Process data object (PDO)

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

Four RPDOs and four TPDOs are implemented:

- Receive process data object (RPDO), described in this chapter.
- Transmit process data object (TPDO), ⇨ Chapter "6.7.14 Transmit process data object (TPDO)", page 127

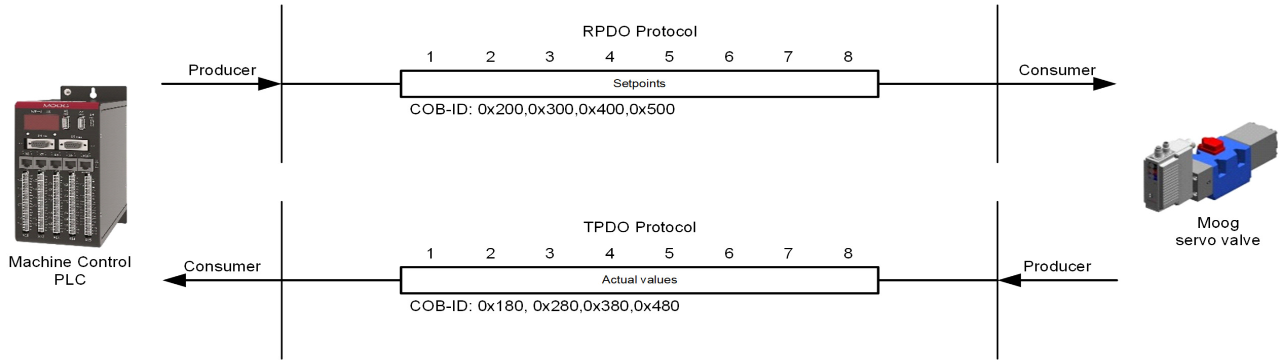


Figure 42: PDO Real Time Data Transmission

With the receive process data object (RPDO) mapping most object dictionary entries can be mapped to RPDO. To enable receive process data object (RPDO) transmission, the local application parameters must be mapped to the RPDO and a transmission type must be selected for each RPDO channel. Four RPDO channels are available.

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 bits can be mapped within one RPDO. An arbitrary combination of different data types is possible if the sum of the mapped RPDO data is less or equal 8 bytes.

Two different kind of parameters are used to configure the transmission.

- Parameters 0x5400…0x5402 are used to configure the type of transmission.
- Parameters 0x5600…0x5603 are used to configure which application data is transmitted.

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

- Device state machine (DSM) Control Word <ControlWord> (0x6040) (sub-index 0x00)
  ⇨ Chapter "5.1 Device state machine (DSM)", page 50
- Q setpoint <QSetpoint> (0x6300) (sub-index 0x01)
  ⇨ Chapter "6.1.3 Q setpoint path", page 65
- Pressure setpoint <PrsSetpoint> (0x6380) (sub-index 0x01)
  ⇨ Chapter "6.1.4 p setpoint path", page 67

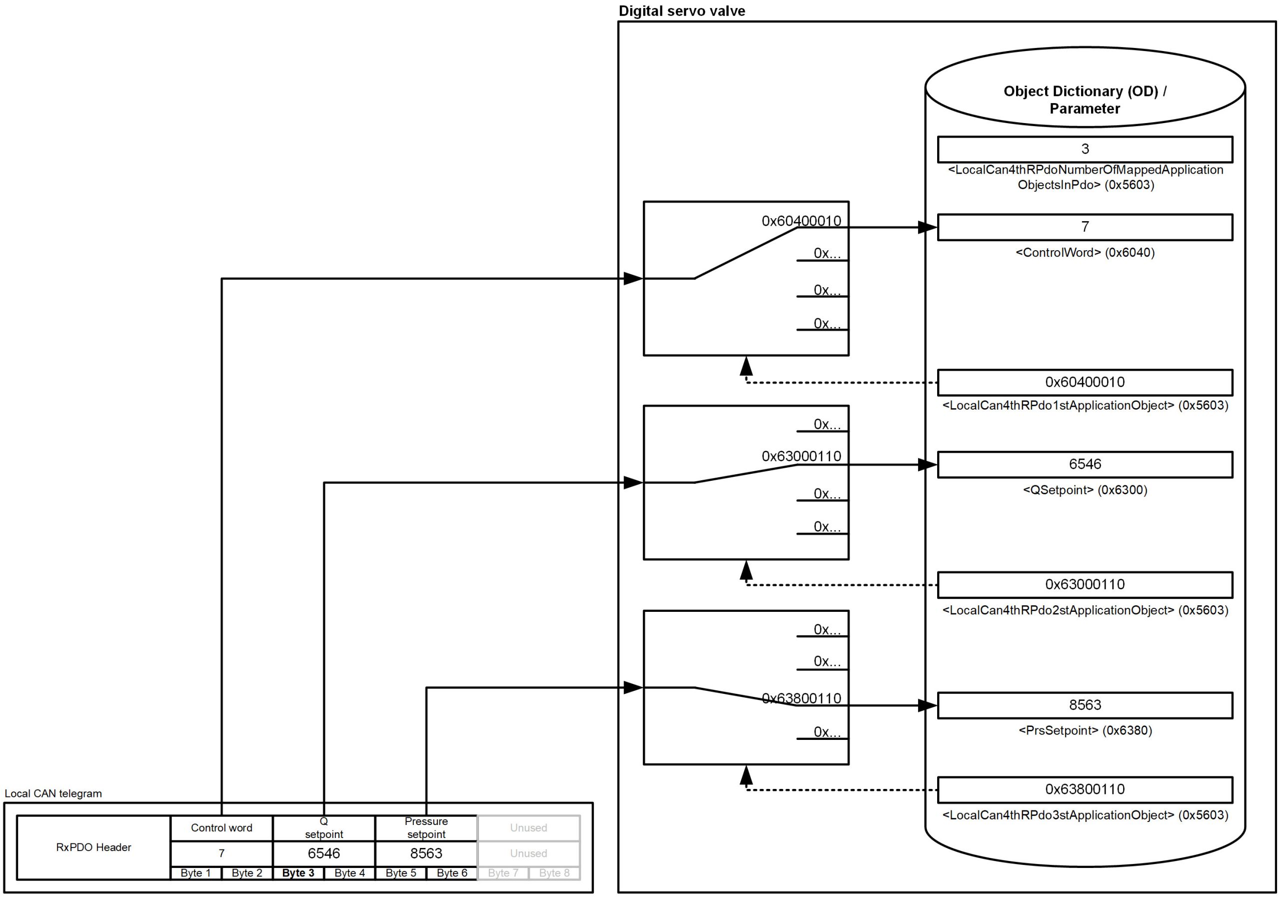


Figure 43: Receive process data object (RPDO) mapping

For example, the parameter <LocalCan4thRPdoNumberOfMappedApplicationObjectsInPdo> (0x5603) defines the number of mapped values for the fourth RPDO. The second sub-index is a reference to the <ControlWord> (0x6040) (sub-index 0x00). The references to the parameters <QSetpoint> (0x6300) (sub index 0x01) and <PrsSetpoint> (0x6380) (sub-index 0x01) are defined in the same manner.

For detailed information how the PDO protocol is working, please take a look into the actual CiA 301 CANopen application layer and communication profile.

6.7.13.1 Object 0x5400: 1st RPDO configuration

First RPDO channel configuration.

LocalCAN							
Index	Sub-in-dex	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5400	1	LocalCan1stRPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0200+127
0x5400	2	LocalCan1stRPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x5400	5	LocalCan1stRPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description

Table 58: Object 0x5400: 1st RPDO configuration

Parameter	Description
<LocalCan1stRPdoCobIdUsedByPdo>	COB-ID of the 1st RPDO.
<LocalCan1stRPdoTransmissionType>	Different trigger can be configured: ⇨ Table 60, page 122
<LocalCan1stRPdoEventTimer>	This parameter defines the timeout in milliseconds for the RPDO time-out monitoring. The event timer is used to recognize the expiration of the RPDO. 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 59: Possible values of parameter <LocalCan1stRPdoCobIdUsedByPdo> (0x5400)

<LocalCan1stRPdoCobIdUsedByPdo>				
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 60: Possible values of parameter <LocalCan1stRPdoTransmissionType> (0x5400)

<LocalCan1stRPdoTransmissionType>	
0	The last incoming PDO will be processed synchronously to the next in-coming SYNC telegram.
1…240	The last incoming PDO will be processed synchronously to every <LocalCan1stRPdoTransmissionType> (0x5400) SYNC telegram.
241…253	Reserved
254…255	PDO will be processed immediately after reception.

6.7.13.2 Object 0x5401: 2nd RPDO configuration

Second RPDO channel configuration.

LocalCAN							
Index	Sub-in-dex	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5401	1	LocalCan2ndRPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0300+127
0x5401	2	LocalCan2ndRPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x5401	5	LocalCan2ndRPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
➪ Table 58, page 122

6.7.13.3 Object 0x5402: 3rd RPDO configuration

Third RPDO channel configuration.

LocalCAN							
Index	Sub-in-dex	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5402	1	LocalCan3rdRPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0400+127
0x5402	2	LocalCan3rdRPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x5402	5	LocalCan3rdRPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
➪ Table 58, page 122

6.7.13.4 Object 0x5403: 4th RPDO configuration

Fourth RPDO channel configuration.

LocalCan							
Index	Sub-in-dex	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5403	1	LocalCan4thRPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0500+127
0x5403	2	LocalCan4thRPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x5403	5	LocalCan4thRPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
➪ Table 58, page 122

6.7.13.5 Object 0x5600: 1st RPDO mapping

With the parameter <LocalCan1stRPdoNumberOfMappedApplicParaInPdo> (0x5600) 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.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5600	0	LocalCan1stRPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	1
0x5600	1	LocalCan1stRPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60400010
0x5600	2	LocalCan1stRPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0
0x5600	3	LocalCan1stRPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x5600	4	LocalCan1stRPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x5600	5	LocalCan1stRPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x5600	6	LocalCan1stRPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x5600	7	LocalCan1stRPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x5600	8	LocalCan1stRPdo8thApplicationObject	UINT32	rw	Y	UINT32	0

Value description

Table 61: Object 0x5600: 1st RPDO mapping

Parameter	Description
<LocalCan1stRPdoNumberOfMappedApplicationObjectsInPdo>	Numbers of mapped application parameters
<LocalCan1stRPdo1stApplicationObject>	Mapping of 1st application parameter
<LocalCan1stRPdo2ndApplicationObject>	Mapping of 2nd application parameter
<LocalCan1stRPdo3rdApplicationObject>	Mapping of 3rd application parameter
<LocalCan1stRPdo4thApplicationObject>	Mapping of 4th application parameter
<LocalCan1stRPdo5thApplicationObject>	Mapping of 5th application parameter
<LocalCan1stRPdo6thApplicationObject>	Mapping of 6th application parameter
<LocalCan1stRPdo7thApplicationObject>	Mapping of 7th application parameter
<LocalCan1stRPdo8thApplicationObject>	Mapping of 8th application parameter

Table 62: Value description of mapping parameter <LocalCan1stRPdo1stApplicationObject>…<LocalCan1stRPdo8thApplicationObject>

<LocalCan1stRPdo1stApplicationObject>…<LocalCan1stRPdo8thApplicationObject>				
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

This reference 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).

6.7.13.6 Object 0x5601: 2nd RPDO mapping

Second RPDO mapping.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5601	0	LocalCan2ndRPdoNumberOfMappedApplicationOb-jectsInPdo	UINT8	rw	Y	0…8	2
0x5601	1	LocalCan2ndRPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60400010
0x5601	2	LocalCan2ndRPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63000110
0x5601	3	LocalCan2ndRPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x5601	4	LocalCan2ndRPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x5601	5	LocalCan2ndRPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x5601	6	LocalCan2ndRPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x5601	7	LocalCan2ndRPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x5601	8	LocalCan2ndRPdo8thApplicationObject	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 ‹QSetpoint› (0x6300).

Value description
➪ Table 61, page 124

6.7.13.7 Object 0x5602: 3rd RPDO mapping

Third RPDO mapping.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5602	0	LocalCan3rdRPdoNumberOfMappedApplicationOb-jectsInPdo	UINT8	rw	Y	0…8	2
0x5602	1	LocalCan3rdRPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60400010
0x5602	2	LocalCan3rdRPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63800110
0x5602	3	LocalCan3rdRPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x5602	4	LocalCan3rdRPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x5602	5	LocalCan3rdRPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x5602	6	LocalCan3rdRPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x5602	7	LocalCan3rdRPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x5602	8	LocalCan3rdRPdo8thApplicationObject	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 61, page 124

6.7.13.8 Object 0x5603: 4th RPDO mapping

Fourth RPDO mapping.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5603	0	LocalCan4thRPdoNumberOfMappedApplicationOb-jectsInPdo	UINT8	rw	Y	0…8	3
0x5603	1	LocalCan4thRPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60400010
0x5603	2	LocalCan4thRPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63000110
0x5603	3	LocalCan4thRPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0x63800110
0x5603	4	LocalCan4thRPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x5603	5	LocalCan4thRPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x5603	6	LocalCan4thRPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x5603	7	LocalCan4thRPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x5603	8	LocalCan4thRPdo8thApplicationObject	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 <QSetpoint> (0x6300) and the <PrsSetpoint> (0x6380).

Value description
⇨ Table 61, page 124

6.7.13.9 Object 0x5112: RPDO counter

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

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5112	1	LocalCan1stRPdoCounter	UINT32	rw	N	UINT32	0
0x5112	2	LocalCan2ndRPdoCounter	UINT32	rw	N	UINT32	0
0x5112	3	LocalCan3rdRPdoCounter	UINT32	rw	N	UINT32	0
0x5112	4	LocalCan4thRPdoCounter	UINT32	rw	N	UINT32	0

6.7.14 Transmit process data object (TPDO)

The transmit process data object (TPDO) protocol must be configured and the transmit values must be mapped to the local parameters. Four transmit TPDO 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.

With the transmit process data object (TPDO) mapping the most object dictionary entries can be mapped to a TPDO. 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 TPDO. An arbitrary combination of different data types is possible, if the sum of the mapped TPDO 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 TPDO (default mapping):

- Device state machine (DSM) Status Word <StatusWord> (0x6041) (sub-index 0x00)
  ➪ Chapter "5.1 Device state machine (DSM)", page 50
- Spool position actual value <SpoolActualValue> (0x6301) (sub-index 0x01)
  ➪ Chapter "7.3 Spool position controller", page 163
- Pressure actual value <PrsActualValue> (0x6381) (sub-index 0x01)
  ➪ Chapter "7.5.11 Actual pressure value filter", page 196

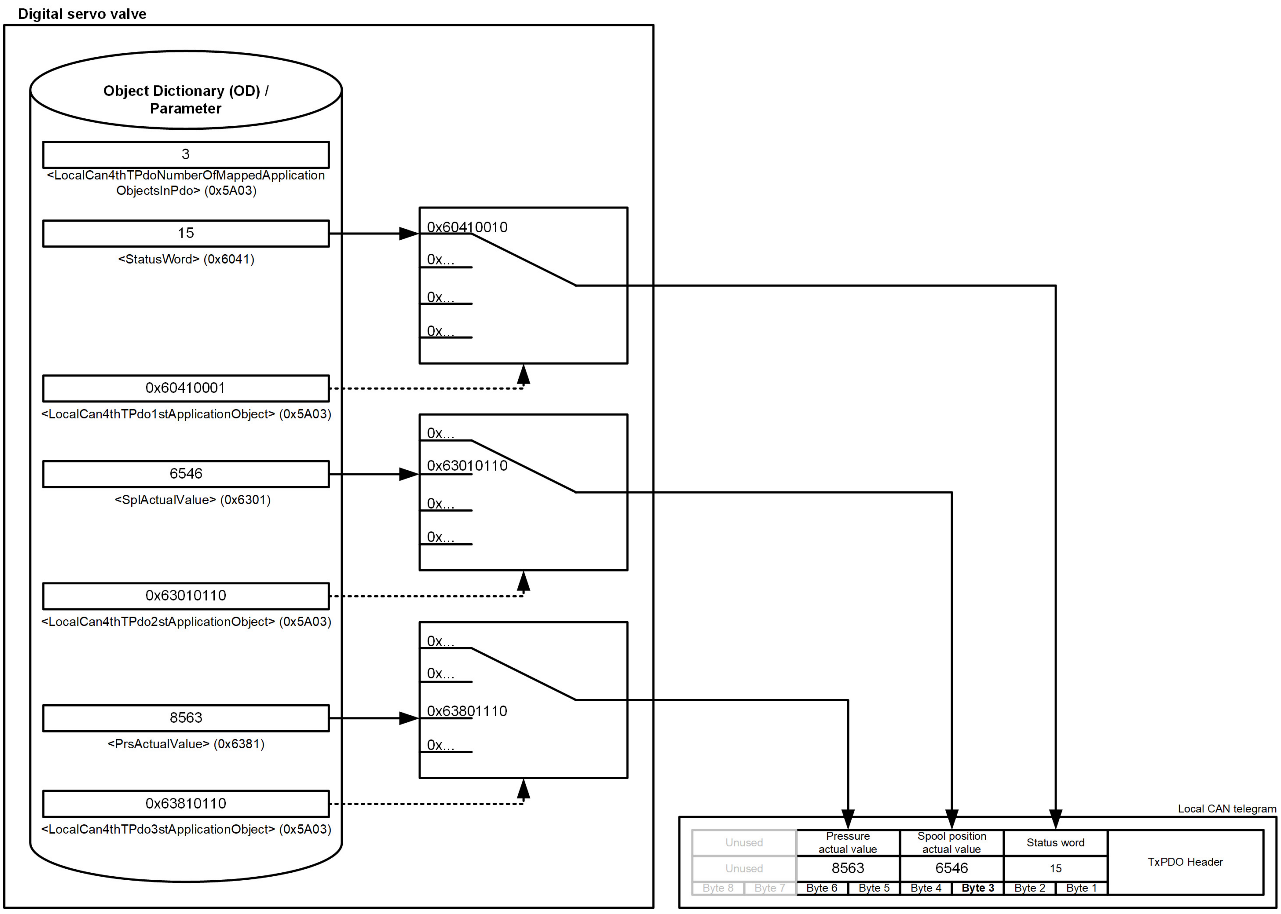


Figure 44: Transmit process data object (TPDO) mapping

With the parameter <LocalCan4thTPdoNumberOfMappedApplicationObjectsInPdo> 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 <SpoolActualValue> (0x6301) (sub-index 0x01) and <PrsActualValue> (0x6381) (sub-index 0x01) are done in the same manner.

For detailed information how the PDO protocol is working, please take a look into the actual CiA 301 CANopen application layer and communication profile.

6.7.14.1 Object 0x5800: 1st TPDO configuration

First TPDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5800	1	LocalCan1stTPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF| 0x80000000	0x0180+127
0x5800	2	LocalCan1stTPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x5800	3	LocalCan1stTPdoInhibitTime	UINT16	rw	Y	UINT16	0
0x5800	5	LocalCan1stTPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description

Table 63: Object 0x5800: 1st TPDO configuration

Parameter	Description
‹LocalCan1stTPdoCobIdUsedByPdo›	Defines the COB-ID for the 1st TPDO.
‹LocalCan1stTPdoTransmissionType›	Defines the transmission behavior for the 1st TPDO.
‹LocalCan1stTPdoInhibitTime›	Not implemented.
‹LocalCan1stTPdoEventTimer›	This parameter defines the event time in milliseconds. This time defines the cycle time of the TPDO timer event. If the event time elapsed, a trigger to transmit a TPDO is initiated and the event timer will be restarted.

Table 64: Possible values of parameter ‹LocalCan1stTPdoCobIdUsedByPdo› (0x5800)

‹LocalCan1stTPdoCobIdUsedByPdo›				
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 65: Possible values of parameter ‹LocalCan1stTPdoTransmissionType› (0x5800)

‹LocalCan1stTPdoTransmissionType›	
0	The process data for the TPDO will be updated and sent immediately after the next incoming SYNC telegram.
1…240	The process data for the TPDO will be updated and sent immediately every ‹LocalCan1stTPdoTransmissionType› SYNC telegram.
241…253	Reserved
254…255	The TPDO will be sent after the event time is elapses (when the event time is nonzero).

To initiate a single request only, the ‹LocalCan1stTPdoEventTimer› (0x5800)…‹LocalCan4thTPdoEventTimer› (0x5803) must be set to 0. The transmission then can be triggered by the parameter ‹LocalCanTPdoTrigger› (0x5111).

➪ Chapter "6.7.14.9 Object 0x5111: TPDO trigger", page 133

6.7.14.2 Object 0x5801: 2nd TPDO configuration

Second TPDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5801	1	LocalCan2ndTPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF|0x80000000	0x0280+127
0x5801	2	LocalCan2ndTPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x5801	3	LocalCan2ndTPdoInhibitTime	UINT16	rw	Y	UINT16	0
0x5801	5	LocalCan2ndTPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
➪ Table 63, page 129

6.7.14.3 Object 0x5802: 3rd TPDO configuration

Third TPDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5802	1	LocalCan3rdTPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF|0x80000000	0x0380+127
0x5802	2	LocalCan3rdTPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x5802	3	LocalCan3rdTPdoInhibitTime	UINT16	rw	Y	UINT16	0
0x5802	5	LocalCan3rdTPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
➪ Table 63, page 129

6.7.14.4 Object 0x5803: 4th TPDO configuration

Fourth TPDO channel configuration.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5803	1	LocalCan4thTPdoCobIdUsedByPdo	UINT32	rw	Y	1…0x000007FF|0x80000000	0x0480+127
0x5803	2	LocalCan4thTPdoTransmissionType	UINT8	rw	Y	UINT8	255
0x5803	3	LocalCan4thTPdoInhibitTime	UINT16	rw	Y	UINT16	0
0x5803	5	LocalCan4thTPdoEventTimer	UINT16	rw	Y	UINT16	0

Value description
➪ Table 63, page 129

6.7.14.5 Object 0x5A00: 1st TPDO mapping

With the parameter <LocalCan1stTPdoNumberOfMappedApplicationObjectsInPdo> (0x5A00) 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.

LocalCan							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5A00	0	LocalCan1stTPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	1
0x5A00	1	LocalCan1stTPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60410010
0x5A00	2	LocalCan1stTPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0
0x5A00	3	LocalCan1stTPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x5A00	4	LocalCan1stTPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A00	5	LocalCan1stTPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A00	6	LocalCan1stTPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A00	7	LocalCan1stTPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A00	8	LocalCan1stTPdo8thApplicationObject	UINT32	rw	Y	UINT32	0

Value description

Table 66: Object 0x5A00: 1st TPDO mapping

Parameter	Description
<LocalCan1stTPdoNumberOfMappedApplicationObjectsInPdo>	Numbers of mapped application parameters
<LocalCan1stTPdo1stApplicationObject>	1st application parameter
<LocalCan1stTPdo2ndApplicationObject>	2nd application parameter
<LocalCan1stTPdo3rdApplicationObject>	3rd application parameter
<LocalCan1stTPdo4thApplicationObject>	4th application parameter
<LocalCan1stTPdo5thApplicationObject>	5th application parameter
<LocalCan1stTPdo6thApplicationObject>	6th application parameter
<LocalCan1stTPdo7thApplicationObject>	7th application parameter
<LocalCan1stTPdo8thApplicationObject>	8th application parameter

Table 67: Possible values of parameter <LocalCan1stTPdo1stApplicationObject>…<LocalCan1stTPdo8thApplicationObject>

<LocalCan1stTPdo1stApplicationObject>…<LocalCan1stTPdo8thApplicationObject>				
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

This reference 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).

6.7.14.6 Object 0x5A01: 2nd TPDO mapping

Second TPDO mapping.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5A01	0	LocalCan2ndTPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	2
0x5A01	1	LocalCan2ndTPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60410010
0x5A01	2	LocalCan2ndTPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63010110
0x5A01	3	LocalCan2ndTPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x5A01	4	LocalCan2ndTPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A01	5	LocalCan2ndTPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A01	6	LocalCan2ndTPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A01	7	LocalCan2ndTPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A01	8	LocalCan2ndTPdo8thApplicationObject	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 ‹SpoolActualValue› (0x6301).

Value description
⇨ Table 66, page 131

6.7.14.7 Object 0x5A02: 3rd TPDO mapping

Third TPDO mapping.

LocalCan							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5A02	0	LocalCan3rdTPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	2
0x5A02	1	LocalCan3rdTPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60410010
0x5A02	2	LocalCan3rdTPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63810110
0x5A02	3	LocalCan3rdTPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x5A02	4	LocalCan3rdTPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A02	5	LocalCan3rdTPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A02	6	LocalCan3rdTPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A02	7	LocalCan3rdTPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A02	8	LocalCan3rdTPdo8thApplicationObject	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 66, page 131

6.7.14.8 Object 0x5A03: 4th TPDO mapping

Fourth TPDO mapping.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5A03	0	LocalCan4thTPdoNumberOfMappedApplicationObjectsInPdo	UINT8	rw	Y	0…8	2
0x5A03	1	LocalCan4thTPdo1stApplicationObject	UINT32	rw	Y	UINT32	0x60410010
0x5A03	2	LocalCan4thTPdo2ndApplicationObject	UINT32	rw	Y	UINT32	0x63010110
0x5A03	3	LocalCan4thTPdo3rdApplicationObject	UINT32	rw	Y	UINT32	0x63810110
0x5A03	4	LocalCan4thTPdo4thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A03	5	LocalCan4thTPdo5thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A03	6	LocalCan4thTPdo6thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A03	7	LocalCan4thTPdo7thApplicationObject	UINT32	rw	Y	UINT32	0
0x5A03	8	LocalCan4thTPdo8thApplicationObject	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 <SpoolActualValue> (0x6301) and the <PrsActualValue> (0x6381).

Value description
⇨ Table 66, page 131

6.7.14.9 Object 0x5111: TPDO trigger

Writing this parameter triggers a single TPDO. This can be used to transmit a PDO on request only. To trigger one of the four TPDOs, the parameter <LocalCanTPdoTrigger> (0x5111) value must be set to the number of the PDO channel to be sent.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5111	0	LocalCanTPdoTrigger	UINT8	rw	N	1…4	None

6.7.15 Service data object (SDO)

The service data objects are used to configure the settings for the communication and to set or read (upload) application parameters. The SDO data object is connected to the application data via the CANopen object directory. The SDOs are transmitted non real-time with low priority.

If an SDO request cannot be processed, e.g., the value is too big or the client has not the access rights to write (download) a parameter, an SDO abort message with an abort code is sent back from the server to the client.
➪ Chapter "8.3 Abort SDO Transfer Protocol", page 285

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

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

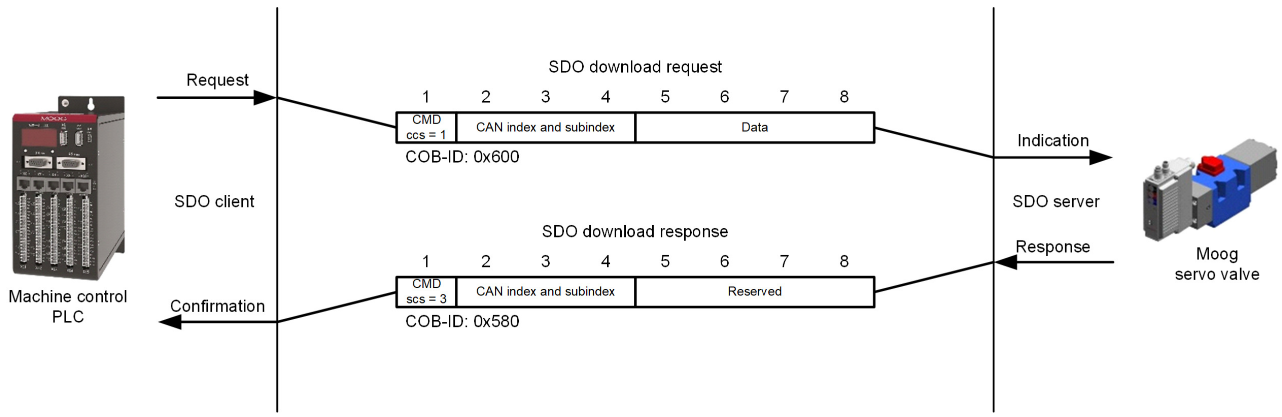


Figure 45: Write service data object request

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

Figure 46: 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 (download) a parameter, an SDO abort transfer message is sent back from the server to the client.

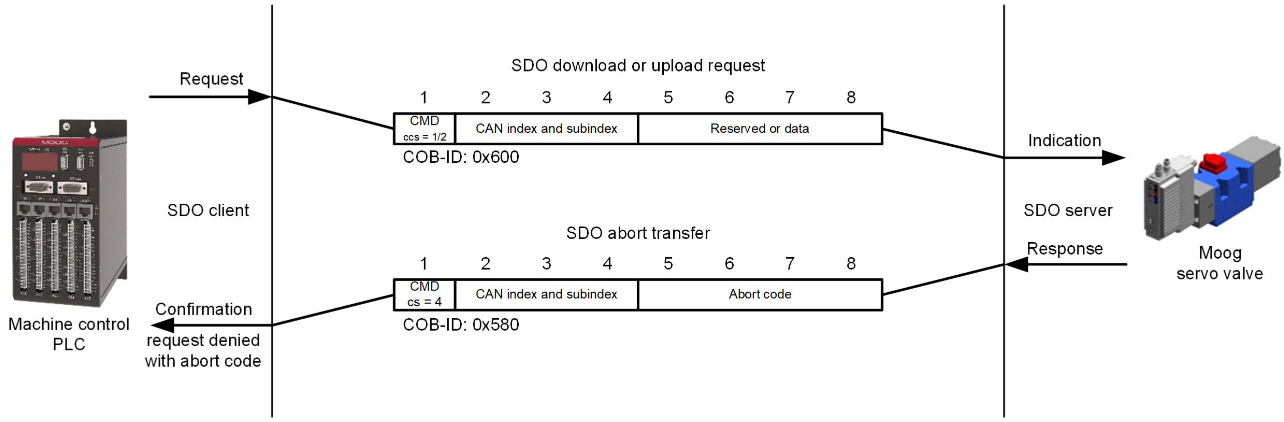


Figure 47: Typical SDO abort message

6.7.15.1 Object 0x5200: SDO client/server parameter

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x5200	1	LocalCanCobIdClientServer	UINT32	ro	-		0x0600+127
0x5200	2	LocalCanCobIdServerClient	UINT32	ro	-	UINT8	0x0580+127

Value description

Table 68: Object 0x5200: SDO client/server parameter

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

6.7.16 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 fieldbus master via the SDO gateway. The local CAN has an 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 + <LocalCANRemoteModuleIdentifier> (0x5B12)	Transmit service data object (TxSDO)	CiA 301
0x600	0x600 + <LocalCANRemoteModuleIdentifier> (0x5B12)	Receive service data object (RxSDO)	CiA 301

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

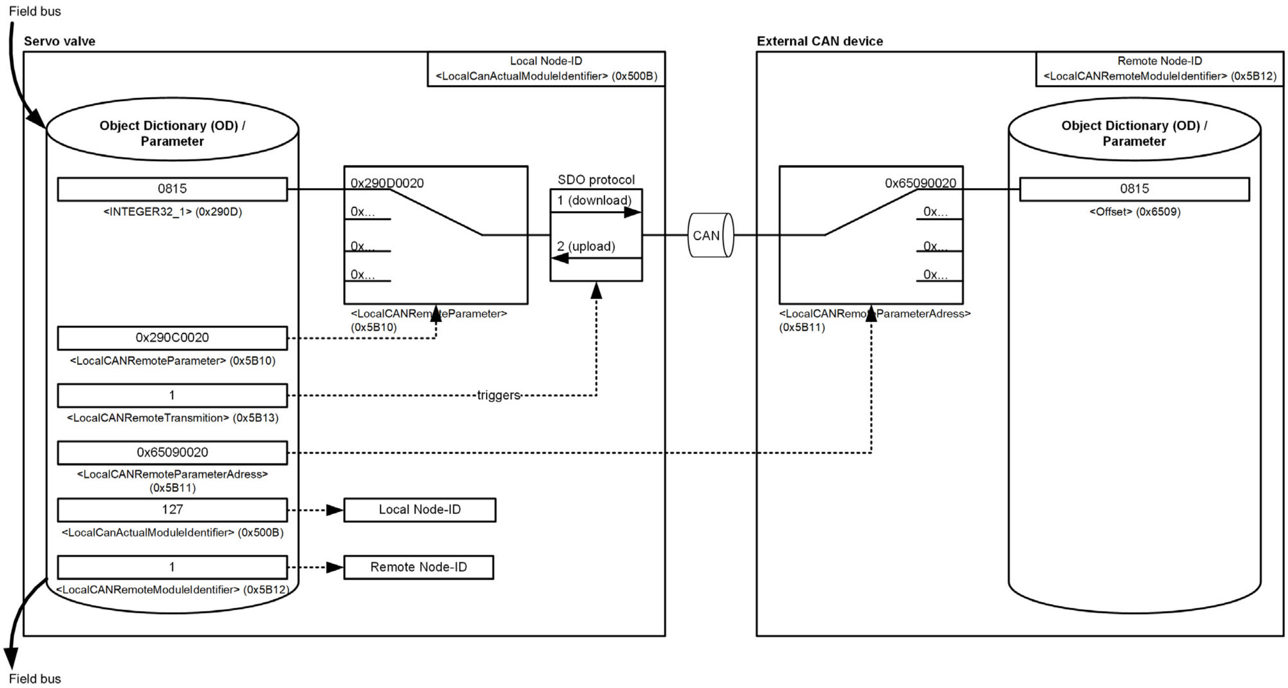


Figure 48: Local CAN service data object (SDO) gateway

6.7.16.1 Object 0x5B10: Remote parameter

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

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

➪ Chapter "6.8 Free to use parameters", page 140

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

Value description

Table 69: Possible values of parameter <LocalCANRemoteParameter> (0x5B10)

<LocalCANRemoteParameter>				
Byte	**3**	**2**	**1**	**0**
Description	Index MSB	Index LSB	Sub-index	Parameter bit length 0x10 or 0x20
Default	0x00	0x00	0x00	0x00

This reference 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 or 0x20.

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

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

Value description

Table 70: Possible values of parameter <LocalCANRemoteParameterAdress> (0x5B11)

<LocalCANRemoteParameterAdress>				
Byte	**3**	**2**	**1**	**0**
Description	Index MSB	Index LSB	Sub-index	Parameter bit length 0x10 or 0x20
Default	0x00	0x00	0x00	0x00

This reference 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 or 0x20.

6.7.16.3 Object 0x5B12: Remote Node-ID

This parameter represents the Node-ID of the external device.

This parameter must be different from the Node-ID <LocalCanActualModuleIdentifier> (0x500B) of the local CAN bus of the servo valve.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5B12	0	LocalCANRemoteModuleIdentifier	UINT8	rw	N	0...127	None

6.7.16.4 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> (0x5B100). If <LocalCANRemoteTransmission> (0x5B13) is set to 2 (upload), the local value at the address <LocalCANRemoteParameter> (0x5B10) will be written to the remote parameter referenced by <LocalCANRemoteParameterAdress> (0x5B11).

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5B13	0	LocalCANRemoteTransmission	INT8	rw	N	-1...2	None

Value description

Table 71: Possible values of parameter <LocalCANRemoteTransmission> (0x5B13)

<LocalCANRemoteTransmission>	Description
-1	Operation was not successful.
0	Operation was successful.
1	Execute an 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.

6.7.17 Synchronization (SYNC)

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 "6.7.13.1 Object 0x5400: 1st RPDO configuration", page 122
➪ Chapter "6.7.14.1 Object 0x5800: 1st TPDO configuration", page 129
➪ Chapter "6.7.14.9 Object 0x5111: TPDO trigger", page 133

6.7.17.1 Object 0x5005: SYNC protocol COB-ID configuration

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

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5005	0	LocalCanCobIdSyncMessage	UINT32	rw	Y	1…0x000007FF |0x40000000	0x0080 +127

Value description

Table 72: Possible values of parameter <LocalCanCobIdSyncMessage> (0x5005)

<LocalCanCobIdSyncMessage>				
Bit	**31**	**30**	**29…11**	**10…0**
Description	Reserved	0: SYNC consumer 1: SYNC producer	Reserved	11 bit COB-ID

6.7.17.2 Object 0x5006: SYNC protocol period

If <LocalCanCobIdSyncMessage> (0x5005) is set to 1, a sync signal is produced by the local CAN fieldbus. The cycle time of the sync signal is set with this parameter in milliseconds.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5006	0	LocalCanCommunicationCyclePeriod	UINT32	rw	Y	UINT32	0

6.7.17.3 Object 0x5019: SYNC protocol counter overflow value

This object shall indicate the configured highest value the synchronous counter supports. If the value is greater than 1, the SYNC message shall have a data length of 1 byte. An EMCY message (error code 0x8240) may be transmitted by a SYNC consumer in the case the configured data length of the SYNC message does not meet the data length of a received SYNC message. The value used shall be the least common multiple of all the TPDO transmission types (1 < n <= 240) used. This ensures that periodic SYNC events always happen in the SYNC cycles with the same counter value.

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5019	0	LocalCANSynchronousCounterOverflowValue	UINT8	rw	Y	UINT8	0

Value description

Table 73: Possible values of parameter <LocalCANSynchronousCounterOverflowValue> (0x5019)

<LocalCANSynchronousCounterOverflowValue>	**Description**
0	The SYNC message shall be transmitted as a CAN message of data length 0.
1	Reserved
2…240	The SYNC message shall be transmitted as a CAN message of data length 1. The first data byte contains the counter.
241…255	Reserved

6.7.18 Emergency (EMCY)

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 specified in the CANopen communication profiles CiA 301 and CiA 408. They are described in the chapter Diagnostics.

➪ Chapter "8 Diagnostics", page 261

6.7.18.1 Object 0x5014: EMCY protocol COB-ID configuration

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

LocalCAN							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5014	0	LocalCanCobIdEmergencyMessage	UINT32	rw	Y	1…0x000007FF	0x0080 +127

Value description

Table 74: Possible values of parameter <LocalCanCobIdEmergencyMessage> (0x5014)

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

6.7.19 CAN Interface features

Depending on the servo valve ordered, a bus supply voltage and/or a switchable terminating resistor is supplied. There is no electrical isolation of the signal lines.

6.7.19.1 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. But not all servo valve variants contain a switchable terminating resistor, others have a fix terminating resistor. If switching with the valve variant is not possible, the following CiA fault code is messaged:
0x0047 - General internal incompatibility in the device.
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	INT8	rw	Y	0…1	0

6.8 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.

6.8.1 Object 0x0002: Signed one byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x0002	0	INTEGER8	INT8	rw	N	INT8	None

6.8.2 Object 0x0003: Signed two byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x0003	0	INTEGER16	INT16	rw	N	INT16	None

6.8.3 Object 0x0004: Signed four byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x0004	0	INTEGER32	INT32	rw	N	INT32	None

6.8.4 Object 0x0005: Unsigned one byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x0005	0	UNSIGNED8	UINT8	rw	N	UINT8	None

6.8.5 Object 0x0006: Unsigned two byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x0006	0	UNSIGNED16	UINT16	rw	N	UINT16	None

6.8.6 Object 0x0007: Unsigned four byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x0007	0	UNSIGNED32	UINT32	rw	N	UINT32	None

6.8.7 Object 0x0008: Four byte real

DataType							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x0008	0	REAL32	REAL32	rw	N	REAL32	None

6.8.8 Object 0x0009: Visible string

DataType							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x0009	0	VISIBLE_STRING	String	rw	N	64 byte	None

6.8.9 Object 0x001B: Unsigned eight byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x001B	0	UNSIGNED64	UINT64	rw	N	UINT64	None

6.8.10 Object 0x290B: Signed one byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x290B	1	Integer08	INT8	rw	N	INT8	None
0x290B	2	Integer08	INT8	rw	N	INT8	None
0x290B	3	Integer08	INT8	rw	N	INT8	None
0x290B	4	Integer08	INT8	rw	N	INT8	None
0x290B	5	Integer08	INT8	rw	N	INT8	None
0x290B	6	Integer08	INT8	rw	N	INT8	None
0x290B	7	Integer08	INT8	rw	N	INT8	None
0x290B	8	Integer08	INT8	rw	N	INT8	None

6.8.11 Object 0x290C: Signed two byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x290C	1	Integer16	INT16	rw	N	INT16	None
0x290C	2	Integer16	INT16	rw	N	INT16	None
0x290C	3	Integer16	INT16	rw	N	INT16	None
0x290C	4	Integer16	INT16	rw	N	INT16	None
0x290C	5	Integer16	INT16	rw	N	INT16	None
0x290C	6	Integer16	INT16	rw	N	INT16	None
0x290C	7	Integer16	INT16	rw	N	INT16	None
0x290C	8	Integer16	INT16	rw	N	INT16	None
0x290C	9	Integer16	INT16	rw	N	INT16	None
0x290C	10	Integer16	INT16	rw	N	INT16	None
0x290C	11	Integer16	INT16	rw	N	INT16	None
0x290C	12	Integer16	INT16	rw	N	INT16	None
0x290C	13	Integer16	INT16	rw	N	INT16	None
0x290C	14	Integer16	INT16	rw	N	INT16	None
0x290C	15	Integer16	INT16	rw	N	INT16	None
0x290C	16	Integer16	INT16	rw	N	INT16	None

6.8.12 Object 0x290D: Signed four byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x290D	1	Integer32	INT32	rw	N	INT32	None
0x290D	2	Integer32	INT32	rw	N	INT32	None
0x290D	3	Integer32	INT32	rw	N	INT32	None
0x290D	4	Integer32	INT32	rw	N	INT32	None
0x290D	5	Integer32	INT32	rw	N	INT32	None
0x290D	6	Integer32	INT32	rw	N	INT32	None
0x290D	7	Integer32	INT32	rw	N	INT32	None
0x290D	8	Integer32	INT32	rw	N	INT32	None

6.8.13 Object 0x290E: Unsigned one byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x290E	1	Unsigned08	UINT8	rw	N	UINT8	None
0x290E	2	Unsigned08	UINT8	rw	N	UINT8	None
0x290E	3	Unsigned08	UINT8	rw	N	UINT8	None
0x290E	4	Unsigned08	UINT8	rw	N	UINT8	None
0x290E	5	Unsigned08	UINT8	rw	N	UINT8	None
0x290E	6	Unsigned08	UINT8	rw	N	UINT8	None
0x290E	7	Unsigned08	UINT8	rw	N	UINT8	None
0x290E	8	Unsigned08	UINT8	rw	N	UINT8	None

6.8.14 Object 0x290F: Unsigned two byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x290F	1	Unsigned16	UINT16	rw	N	UINT16	None
0x290F	2	Unsigned16	UINT16	rw	N	UINT16	None
0x290F	3	Unsigned16	UINT16	rw	N	UINT16	None
0x290F	4	Unsigned16	UINT16	rw	N	UINT16	None
0x290F	5	Unsigned16	UINT16	rw	N	UINT16	None
0x290F	6	Unsigned16	UINT16	rw	N	UINT16	None
0x290F	7	Unsigned16	UINT16	rw	N	UINT16	None
0x290F	8	Unsigned16	UINT16	rw	N	UINT16	None

6.8.15 Object 0x2910: Unsigned four byte integer array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2910	1	Unsigned32	UINT32	rw	N	UINT32	None
0x2910	2	Unsigned32	UINT32	rw	N	UINT32	None
0x2910	3	Unsigned32	UINT32	rw	N	UINT32	None
0x2910	4	Unsigned32	UINT32	rw	N	UINT32	None
0x2910	5	Unsigned32	UINT32	rw	N	UINT32	None
0x2910	6	Unsigned32	UINT32	rw	N	UINT32	None
0x2910	7	Unsigned32	UINT32	rw	N	UINT32	None
0x2910	8	Unsigned32	UINT32	rw	N	UINT32	None

6.8.16 Object 0x2911: Four byte real array

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2911	1	Real32	REAL32	rw	N	REAL32	None
0x2911	2	Real32	REAL32	rw	N	REAL32	None
0x2911	3	Real32	REAL32	rw	N	REAL32	None
0x2911	4	Real32	REAL32	rw	N	REAL32	None
0x2911	5	Real32	REAL32	rw	N	REAL32	None
0x2911	6	Real32	REAL32	rw	N	REAL32	None
0x2911	7	Real32	REAL32	rw	N	REAL32	None
0x2911	8	Real32	REAL32	rw	N	REAL32	None

7 Servo valve functions

This chapter describes how the servo valve operates:

- Controller (spool position, pressure, and flow controller) depending on the <ControlMode> (0x6043)
- Setpoint conditioning (spool position, pressure and flow setpoint conditioning)
- Monitoring functions

The following structure shows the controllers and the signal conditioning blocks and how they are embedded into the whole system.

The gray highlighted blocks are described in detail in this chapter.

➪ Chapter "7.1 Control modes", page 146

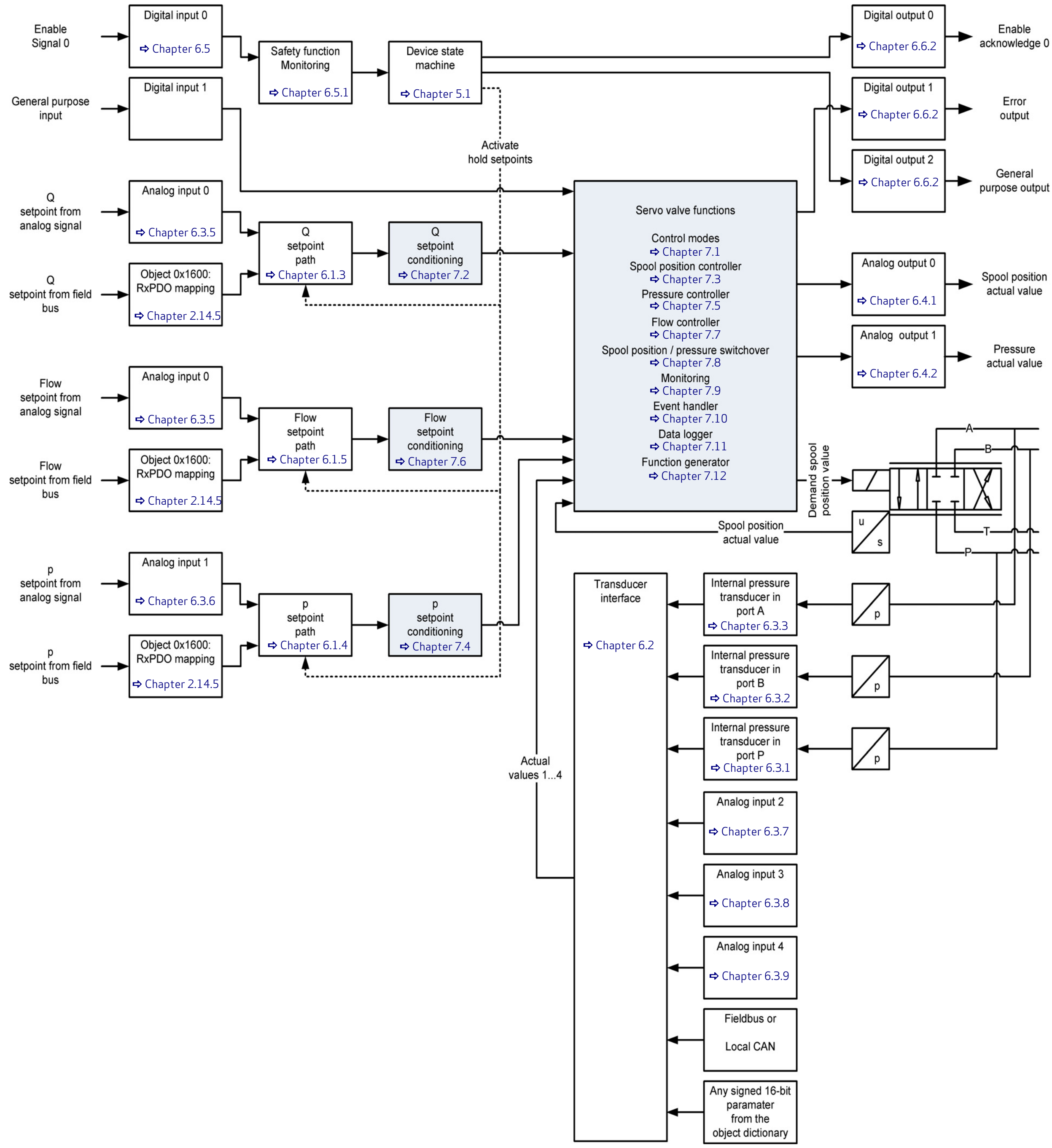


Figure 49: Servo valve controller and signal conditioning

The servo valve can be used to control the spool position (Q), the flow and/or the pressure (p). For these tasks the following control structures are implemented:

- Spool position controller (configured by Moog)
  ➪ Chapter "7.3 Spool position controller", page 163
- Pressure controller
  ➪ Chapter "7.5 Pressure controller", page 189
- Flow controller
  ➪ Chapter "7.7 Flow controller", page 218
- Pressure demand signal polarity
  ➪ Chapter "7.4.6 Pressure demand signal sign", page 188
- Switchover between spool position (Q) or flow and pressure (p)
  ➪ Chapter "7.8 Spool position (Q or flow) / pressure (p) switchover", page 221

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 is set with the parameter <ControlMode> (0x6043).

7.1.1 Object 0x6043: Device control mode

This parameter selects the servo valve control mode.

!	The available control modes are defined by the <DeviceType> (0x1000) parameter. ➪ Chapter "4.1.1 Object 0x1000: Device Type", page 43

Device							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6043	0	ControlMode	INT8	rw	N	1…5, 14	<ControlModeDefault> (0x4043)

Value description

Table 75: Possible values of parameter <ControlMode> (0x6043)

<ControlMode>	Meaning
1	Spool position control open loop Used for tests ➪ Chapter "7.1.3 Spool position control open loop", page 148
2	Spool position control closed loop Spool position control ➪ Chapter "7.1.4 Spool position control closed loop", page 149
3	Pressure control open loop Used for tests. Behaves like a closed loop Q control. ➪ Chapter "7.1.5 Pressure control open loop", page 150
4	Pressure control closed loop Pressure / force control ➪ Chapter "7.1.6 Pressure control closed loop", page 151
5	p/Q control In many applications the p/Q controller is used as Q controller with pressure/force limiting. ➪ Chapter "7.1.7 p/Q control closed loop", page 152
13	Flow control (not yet implemented) - use control mode 14 instead.
14	p/flow Control In many applications the p/flow controller is used as flow controller with pressure/force limiting. ➪ Chapter "7.1.9 p/flow control closed loop", page 153

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…5, 14	2

7.1.3 Spool position control open loop

The spool position open loop mode is selected by setting the parameter <ControlMode> (0x6043) to 1 (spool position control open loop).

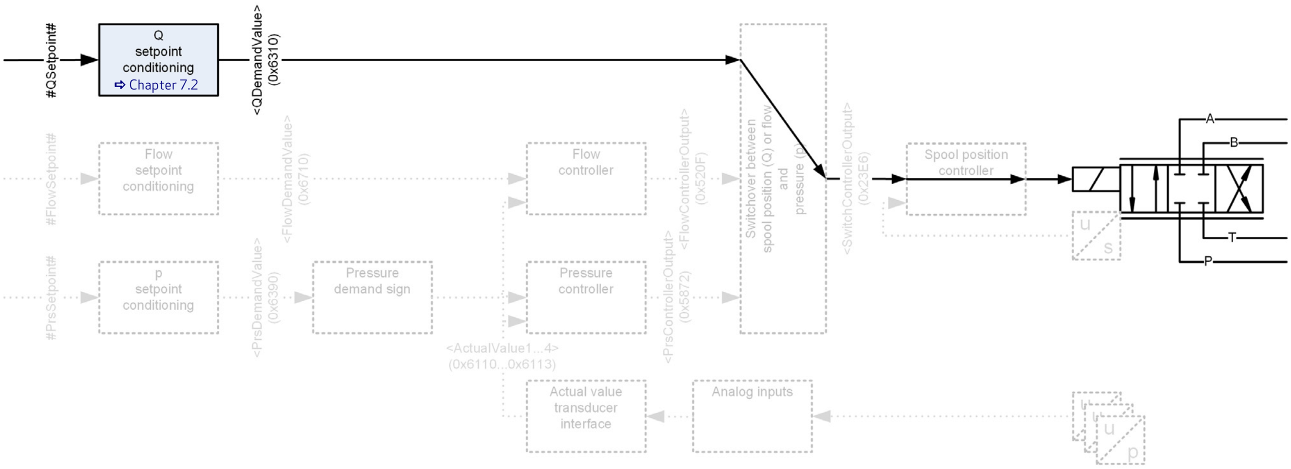


Figure 50: Spool position control open loop for single stage valves

➪ Chapter "7.1.1 Object 0x6043: Device control mode", page 146
➪ Chapter "7.2.1 Object 0x6310: Demand value", page 154

!	This control mode is meant to be used only for commissioning or diagnostic purposes.

!	The signal #FlowSetpoint# is an internal signal only. It links the signal from the Flow setpoint path to the Flow setpoint conditioning. ➪ Chapter "6.1.5 Flow setpoint path", page 69

In case of a dual stage valve, which is not shown in Figure 50, the pilot valve controller is still active.

7.1.4 Spool position control closed loop

The spool position closed loop mode is selected by setting the parameter <ControlMode> (0x6043) to 2 (spool position control closed loop).

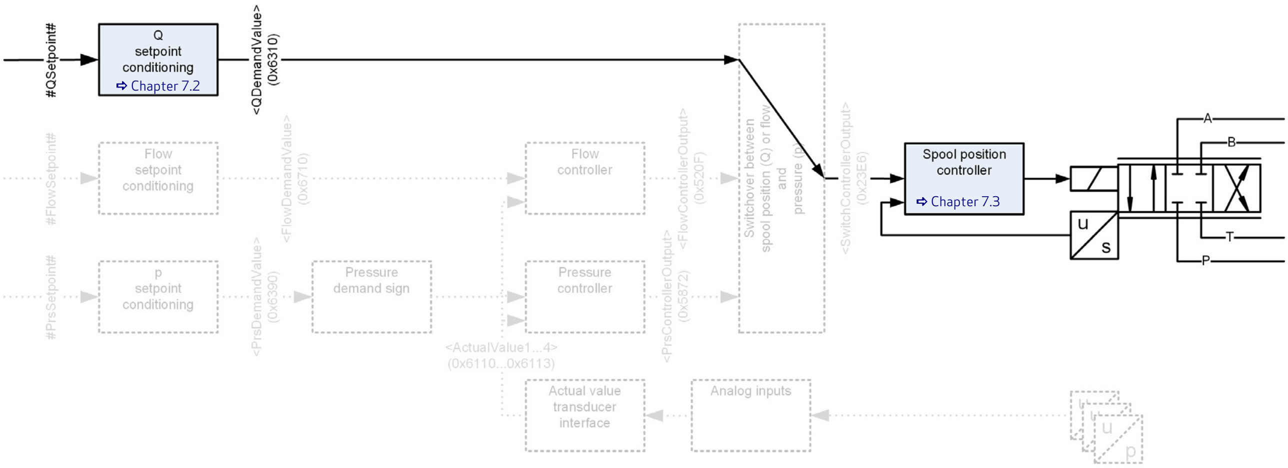


Figure 51: Spool position control closed loop for single stage and dual stage valves

➪ Chapter "7.1.1 Object 0x6043: Device control mode", page 146
➪ Chapter "7.2.1 Object 0x6310: Demand value", page 154

!	The signal #FlowSetpoint# is an internal signal only. It links the signal from the Flow setpoint path to the Flow setpoint conditioning. ➪ Chapter "6.1.5 Flow setpoint path", page 69

7.1.5 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. In contrast to the spool position closed loop control mode, the pressure setpoint is used as input value.

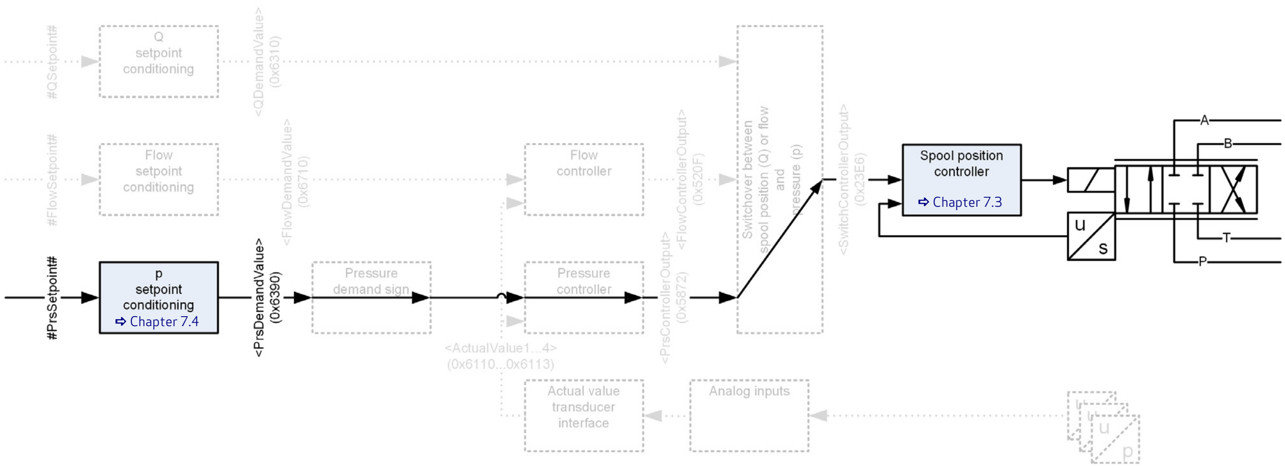


Figure 52: Pressure control open loop

➪ Chapter "7.1.4 Spool position control closed loop", page 149

This mode is for commissioning of the servo valve only and should only be used by Moog staff..

7.1.6 Pressure control closed loop

The pressure control closed loop mode is selected by setting the parameter <ControlMode> (0x6043) to 4 (pressure control closed loop).

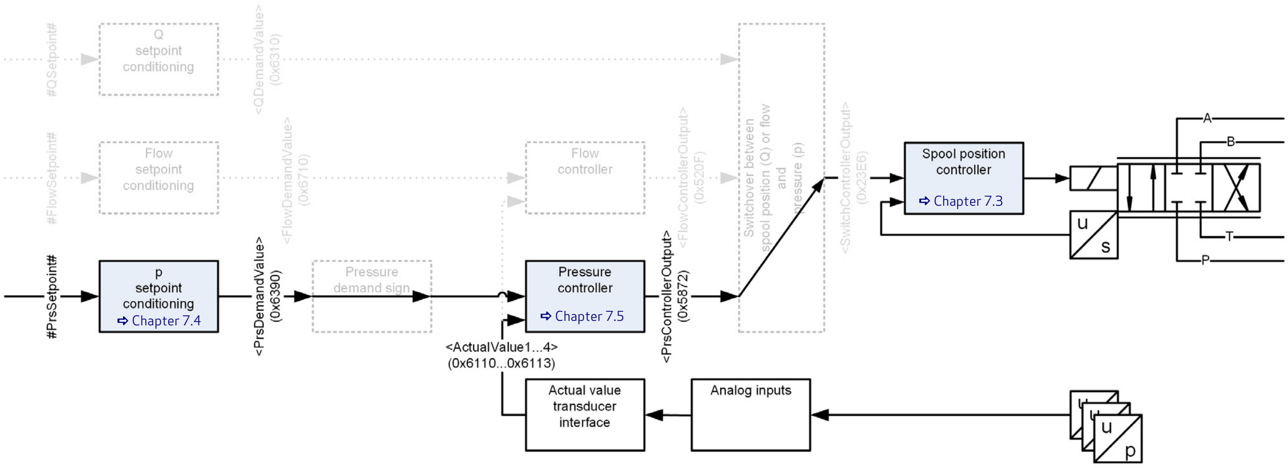


Figure 53: Pressure control closed loop

Description of the feedback signal <ActualValue1…8> (0x6110…0x6113):

➪ Chapter "6.2 Actual value transducer interface", page 72
➪ Chapter "7.1.1 Object 0x6043: Device control mode", page 146
➪ Chapter "7.4.1 Object 0x6390: Setpoint conditioning demand value", page 179
➪ Chapter "7.5.7 Object 0x5872: Pressure controller output", page 191

❗	The signal #PrsSetpoint# is an internal signal only. It links the signal from the p setpoint path to the p setpoint conditioning. ➪ Chapter "6.1.4 p setpoint path", page 67

7.1.7 p/Q control closed loop

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

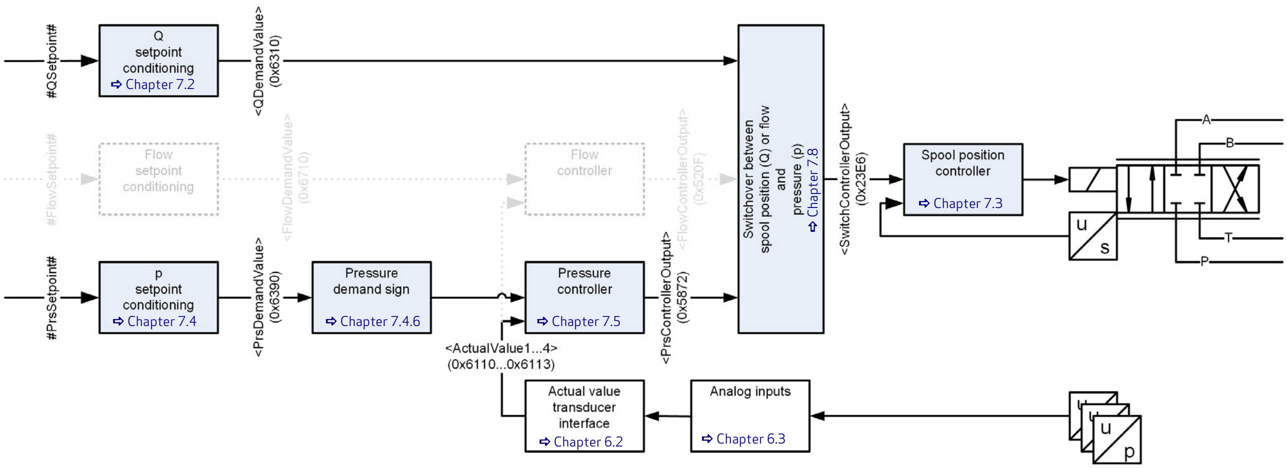


Figure 54: p/Q control closed loop

Description of the feedback signal <ActualValue1…4> (0x6110…0x6113):
➪ Chapter "6.2 Actual value transducer interface", page 72
➪ Chapter "7.2.1 Object 0x6310: Demand value", page 154
➪ Chapter "7.4.1 Object 0x6390: Setpoint conditioning demand value", page 179
➪ Chapter "7.5.7 Object 0x5872: Pressure controller output", page 191
➪ Chapter "7.8.1.1 Object 0x23E6: Switch controller output", page 222

❗	The signal #PrsSetpoint# is an internal signal only. It links the signal from the p setpoint path to the p setpoint conditioning. ➪ Chapter "6.1.4 p setpoint path", page 67

❗	The signal #FlowSetpoint# is an internal signal only. It links the signal from the Flow setpoint path to the Flow setpoint conditioning. ➪ Chapter "6.1.5 Flow setpoint path", page 69

7.1.8 Flow control closed loop

The flow control mode is not yet implemented. The control mode p/flow control can be used instead.

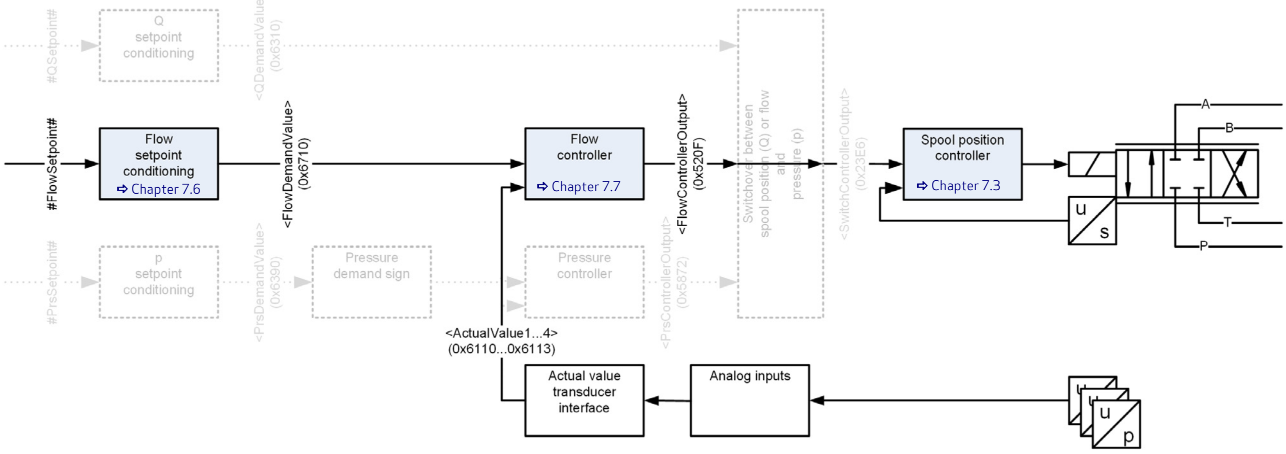


Figure 55: Flow control closed loop

7.1.9 p/flow control closed loop

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

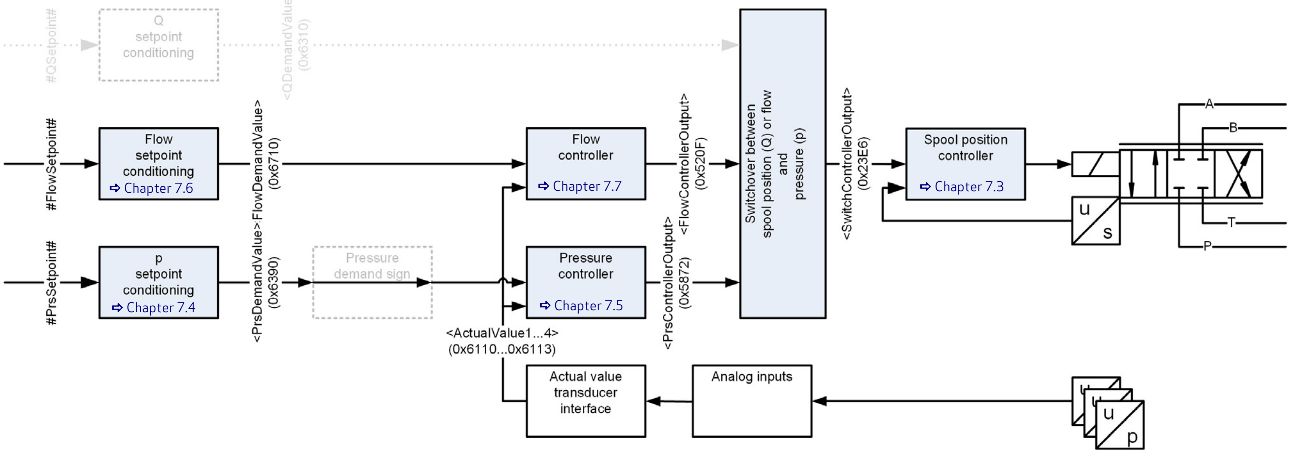


Figure 56: p/flow control closed loop

Description of the feedback signal <ActualValue1...4> (0x6110...0x6113):

➪ Chapter "6.2 Actual value transducer interface", page 72
➪ Chapter "7.6.1 Object 0x6710: Setpoint conditioning demand value", page 209
➪ Chapter "7.4.1 Object 0x6390: Setpoint conditioning demand value", page 179
➪ Chapter "7.5.7 Object 0x5872: Pressure controller output", page 191
➪ Chapter "7.8.1.1 Object 0x23E6: Switch controller output", page 222
➪ Chapter "7.7.4 Object 0x520F: Flow controller output", page 219

!	The signal #PrsSetpoint# is an internal signal only. It links the signal from the p setpoint path to the p setpoint conditioning. ➪ Chapter "6.1.4 p setpoint path", page 67

!	The signal #FlowSetpoint# is an internal signal only. It links the signal from the Flow setpoint path to the Flow setpoint conditioning. ➪ Chapter "6.1.5 Flow setpoint path", page 69

7.2 Q setpoint conditioning

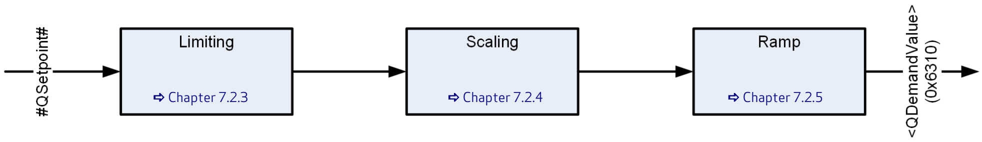


Figure 57: Q setpoint conditioning

!	The internal signal #QSetpoint# is used to link the Q setpoint to the Q setpoint conditioning. ➪ Chapter "6.1.3 Q setpoint path", page 65

7.2.1 Object 0x6310: Demand value

The demand value is generated from the #QSetpoint# by means of the functions in the Q setpoint conditioning and forwarded to the spool position controller.

ValveQControl_SetpointConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6310	1	QDemandValue	INT16	ro	-	INT16	None
0x6310	2	Unit	UINT8	ro	-	UINT8	0
0x6310	3	Prefix	INT8	ro	-	-4…0	0

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

7.2.2 Object 0x6311: Setpoint conditioning reference value

The reference value corresponds to 100 % of the Q input signal. This means that a 100 % Q 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 read by the fieldbus master to scale the setpoint. The reference value is a read only object.

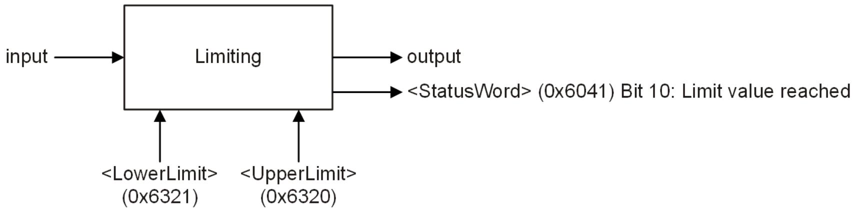
ValveQControl_SetpointConditioning_ReferenceValue							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6311	1	QReferenceValue	INT16	ro	-	INT16	16384
0x6311	2	Unit	UINT8	ro	-	UINT8	0
0x6311	3	Prefix	INT8	ro	-	-4…0	0

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

7.2.3 Limiting

This function limits the value range of the Q 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 "5.1.3.2 Object 0x6041: Status word", page 57



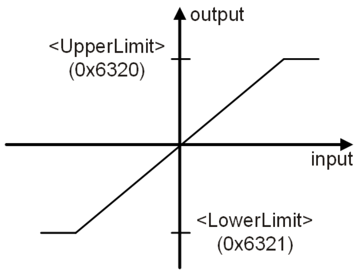


Figure 58: Limit function

!	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: Setpoint conditioning upper setpoint limit

ValveQControl_SetpointConditioning_SetpointLimit							
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	-	-4...0	0

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

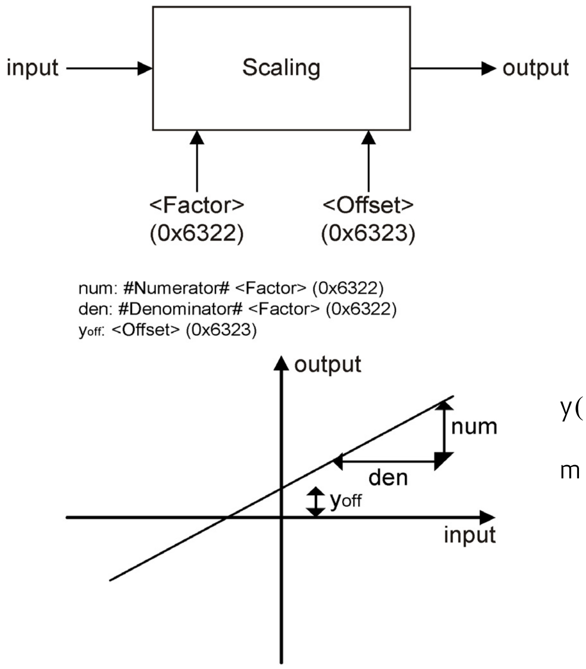
7.2.3.2 Object 0x6321: Setpoint conditioning lower setpoint limit

ValveQControl_SetpointConditioning_SetpointLimit							
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	-	-4…0	0

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

7.2.4 Scaling

This function is used to scale the Q setpoint, e.g., to influence the input signal's range. The output signal is calculated by multiplication of the input signal with a scaling factor and a subsequent addition of an offset according to the following figure.



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

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

Figure 59: Scaling function

7.2.4.1 Object 0x6322: Setpoint conditioning scaling 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).

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

Value description

Table 76: Data structure of the slope factor

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

7.2.4.2 Object 0x6323: Setpoint conditioning scaling offset

This parameter is the offset of the linear output function.

ValveQControl_SetpointConditioning_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	-	-4…0	0

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

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-, two- or four-quadrant ramp function or to deactivate it.

If the signal slope of the input signal is limited by configured slew rate (ramp running), the <StatusWord> (0x6041) bit 9 is set.

If the running ramp is stopped by the bit 15 (ramp stop) of the #ControlWord# signal, then the <StatusWord> (0x6041) bit 15 signalizes that the output value of the ramp signal is hold (ramp frozen).

➪ Chapter "5.1.3.2 Object 0x6041: Status word", page 57
➪ Chapter "5.1.5 Object 0x604F: Device local", page 59
➪ Chapter "5.1.3.1 Object 0x6040: Device control word", page 56
➪ Chapter "5.1.6 Object 0x4040: Local control word", page 59

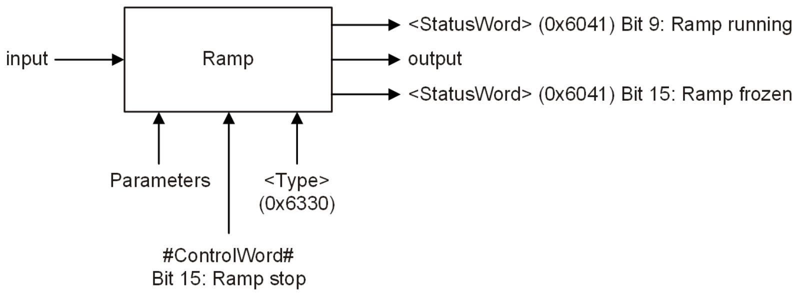


Figure 60: Ramp function

7.2.5.1 Object 0x6330: Setpoint conditioning ramp type

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

ValveQControl_SetpointConditioning_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

Table 77: Possible values of parameter <Type> (0x6330)

<Type>	Description
0	No ramp
1	One-quadrant ramp
2	Two-quadrant ramp
3	Four-quadrant ramp

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.

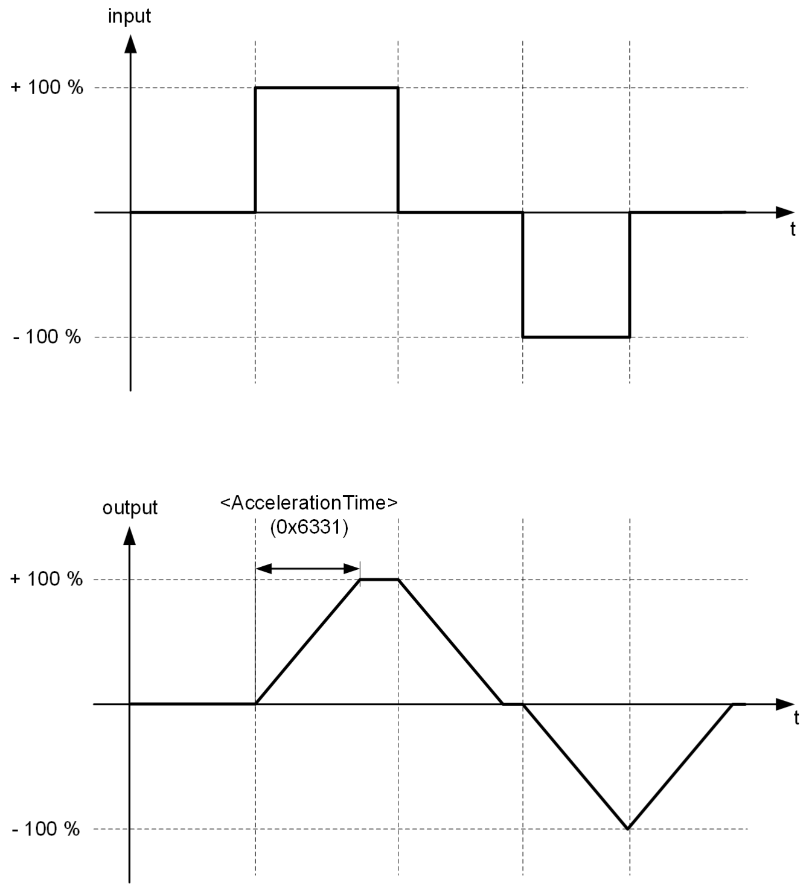


Figure 61: Ramp type 1

7.2.5.2.1 Object 0x6331: Setpoint conditioning ramp 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 61, page 159. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValveQControl_SetpointConditioning_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	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 <DecelerationTime> (0x6332).

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

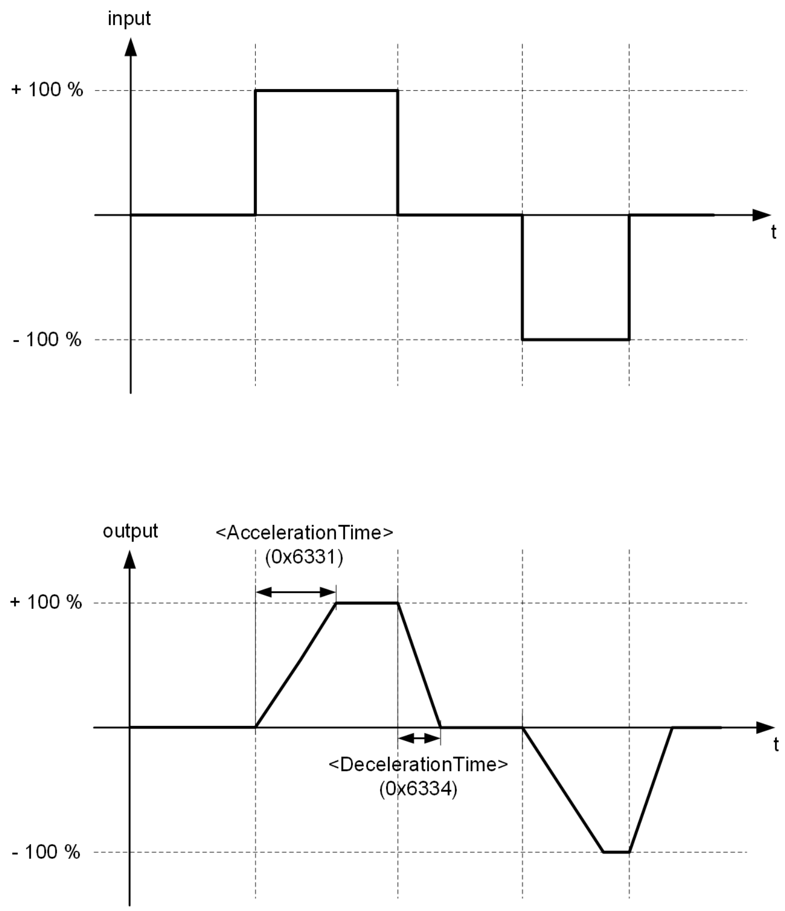


Figure 62: Ramp type 2

7.2.5.3.1 Object 0x6331: Setpoint conditioning ramp acceleration time

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

➪ Chapter "7.2.5.2.1 Object 0x6331: Setpoint conditioning ramp acceleration time", page 159

7.2.5.3.2 Object 0x6334: Setpoint conditioning ramp 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.

ValveQControl_SetpointConditioning_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	Prefix	INT8	rw	Y	-4...0	-3

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

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.

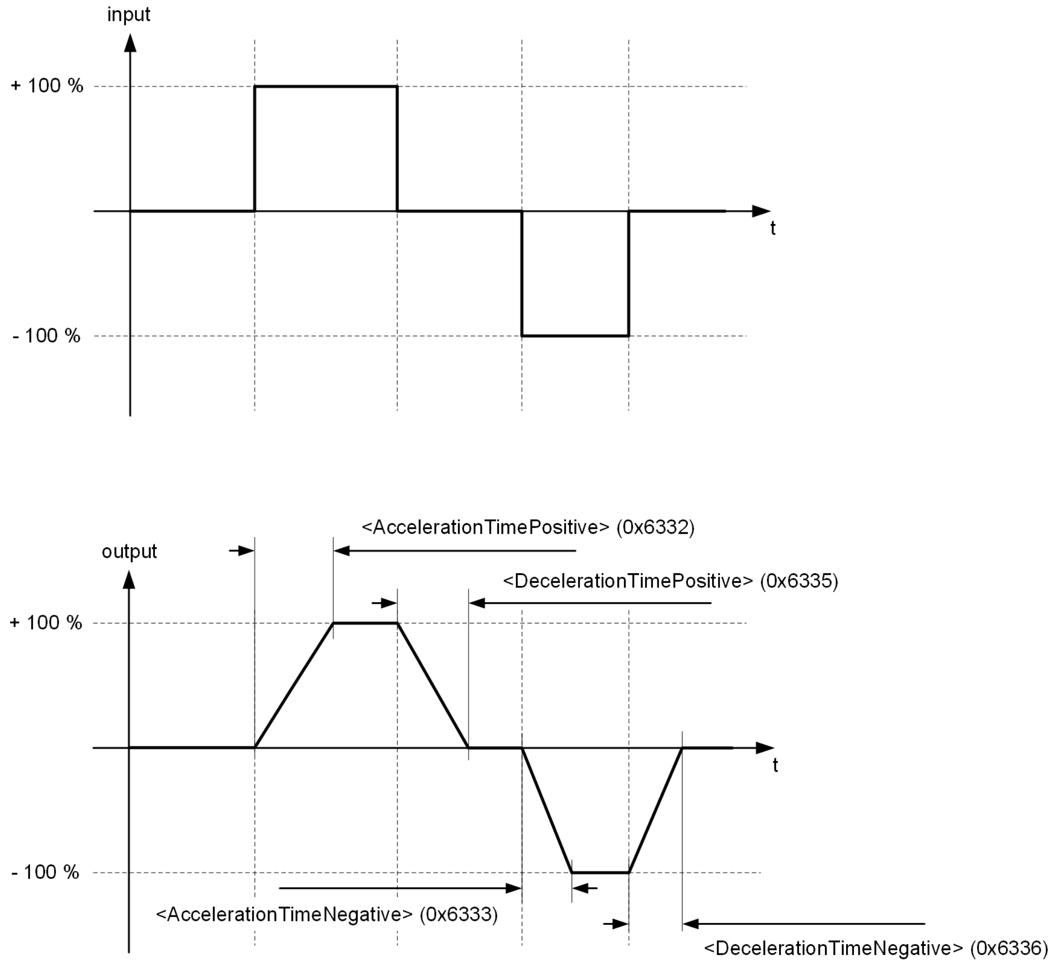


Figure 63: Ramp type 3

7.2.5.4.1 Object 0x6332: Setpoint conditioning ramp 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.

ValveQControl_SetpointConditioning_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	Prefix	INT8	rw	Y	-4...0	-3

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

7.2.5.4.2 Object 0x6333: Setpoint conditioning ramp 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.

ValveQControl_SetpointConditioning_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	Prefix	INT8	rw	Y	-4...0	-3

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

7.2.5.4.3 Object 0x6335: Setpoint conditioning ramp 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.

ValveQControl_SetpointConditioning_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	Prefix	INT8	rw	Y	-4...0	-3

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

7.2.5.4.4 Object 0x6336: Setpoint conditioning ramp 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.

ValveQControl_SetpointConditioning_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	Prefix	INT8	rw	Y	-4...0	-3

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

7.3 Spool position controller

The spool position controller controls the spool position and thus the flow of the oil. 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 closed loop control. For a single stage servo valve, there is only one spool position controller which is the pilot stage controller. If the spool position control open loop is selected by setting the parameter <ControlMode> (0x6043) to 1, the spool position actuator conditioning and the spool position controller are bypassed.

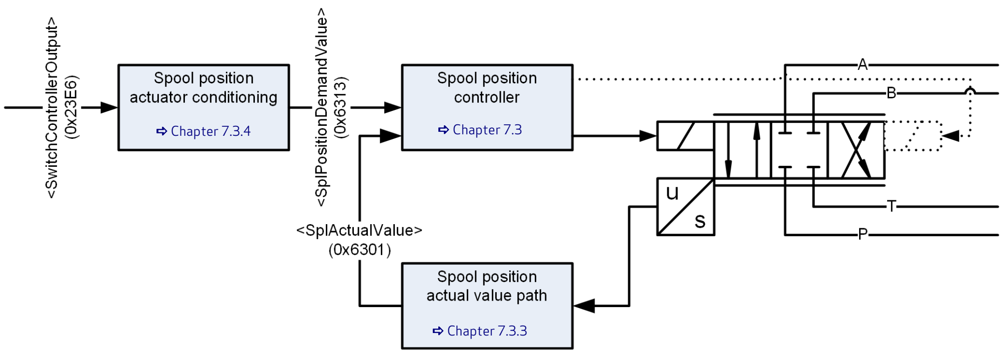


Figure 64: Single stage servo valve

7.3.2 Dual stage servo valve

Control structure of a dual stage servo valve closed loop control. If the spool position control open loop is selected by setting the parameter <ControlMode> (0x6043) to 1, the main stage spool position controller is bypassed.

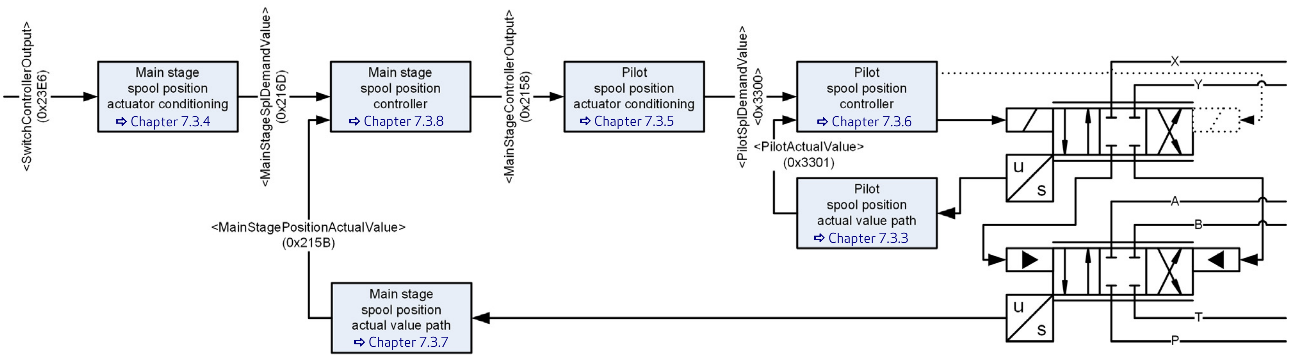


Figure 65: Dual stage servo valve for closed loop control

7.3.3 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).

7.3.3.1 Object 0x6301: Spool position 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 <SpoolActualValue> (0x6301).

For a dual stage servo valve, the spool position value of the main stage <MainStagePositionActualValue>(0x215B) is scaled and mapped to the same signal <SpoolActualValue> (0x6301).
➪ Chapter "7.3.7.1 Object 0x215B: Main stage position actual value", page 177

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6301	1	SpoolActualValue	INT16	ro	-	INT16	None
0x6301	2	Unit	UINT8	ro	-	UINT8	0
0x6301	3	Prefix	INT8	ro	-	−4…0	0

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

7.3.3.2 Object 0x3301: Pilot spool position actual value

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

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

7.3.4 Spool position / main stage spool position actuator conditioning

Actuator conditioning of a servo valve.

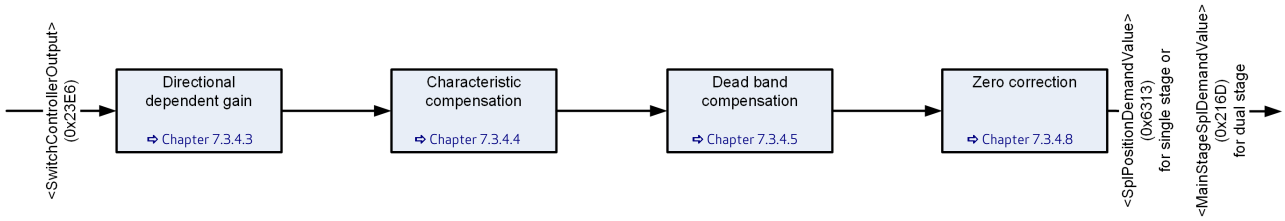


Figure 66: Spool position / main stage spool position actuator conditioning

The following parameters contain the spool position setpoint and demand value depending on the servo valve hardware.

➪ Chapter "7.8.1.1 Object 0x23E6: Switch controller output", page 222

7.3.4.1 Object 0x6313: Spool position demand value

This object provides the output of the actuator conditioning of a single or dual stage servo valve.

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6313	1	SplPositionDemandValue	INT16	ro	-	INT16	None
0x6313	2	Unit	UINT8	ro	-	UINT8	0
0x6313	3	Prefix	INT8	ro	-	-4…0	0

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

7.3.4.2 Object 0x216D: Main stage spool position demand value

This object provides the output of the actuator conditioning of the main stage.

ValveMainStageControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x216D	1	MainStageSplDemandValue	INT16	ro	-	INT16	None

7.3.4.3 Directional dependent gain

This function reduces the input signal's gain by a <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 favors one direction. This compensation need only be approximate but it helps ensure that the closed loop response in both extend and retract directions is symmetrical.

❗	The actual flow depends not only on servo valve opening but on pressure loss in the servo valve and the system load.

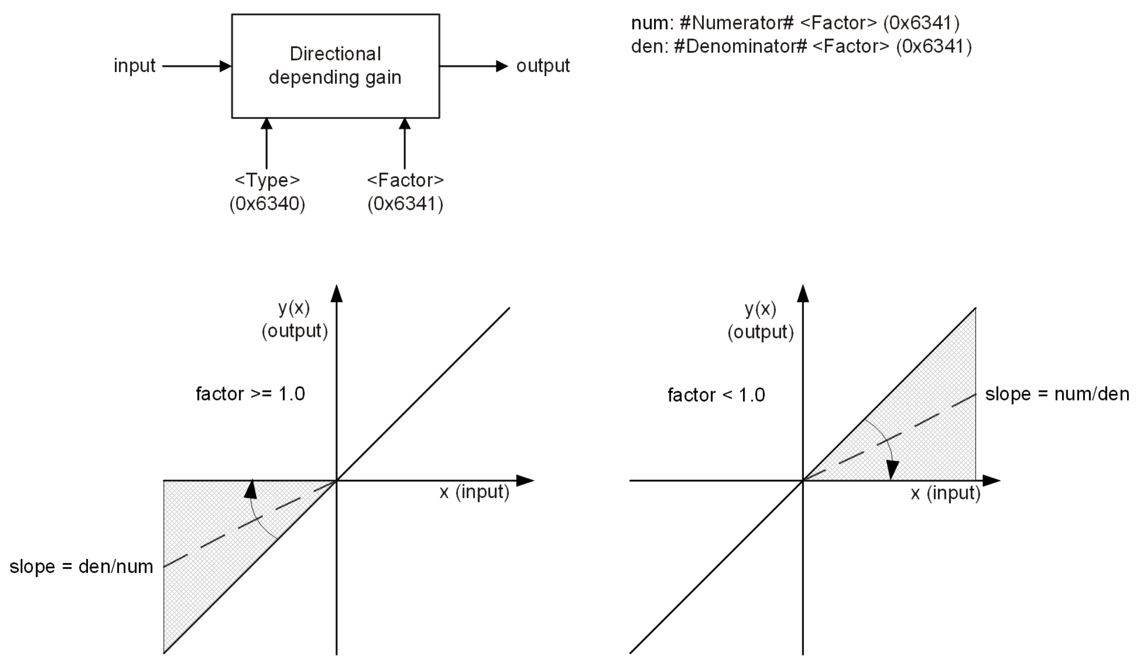


Figure 67: Direction dependent gain

Table 78: Definition of the directional dependent gain factor values

<Factor>	Input	Resulting output
$\lvert Factor \rvert < 1$	Input positive	$\text{Output} = \text{Input} \times \lvert \text{Factor} \rvert$
	Input negative	$\text{Output} = \text{Input}$
$\lvert Factor \rvert \geq 1$	Input positive	$\text{Output} = \text{Input}$
	Input negative	$\text{Output} = \frac{\text{Input}}{\lvert \text{Factor} \rvert}$

7.3.4.3.1 Object 0x6340: Actuator conditioning directional dependent gain type

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

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

Value description

Table 79: Possible values of parameter <Type> (0x6340)

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

7.3.4.3.2 Object 0x6341: Actuator conditioning directional dependent gain 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 <Factor> is only effective, if the parameter <Type> (0x6340) is set to 1 (directional dependent gain function activated). The default value 0x00010001 corresponds to the factor 1.0.

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

Value description

Table 80: Data structure of the directional dependent gain factor

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

7.3.4.4 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 <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.

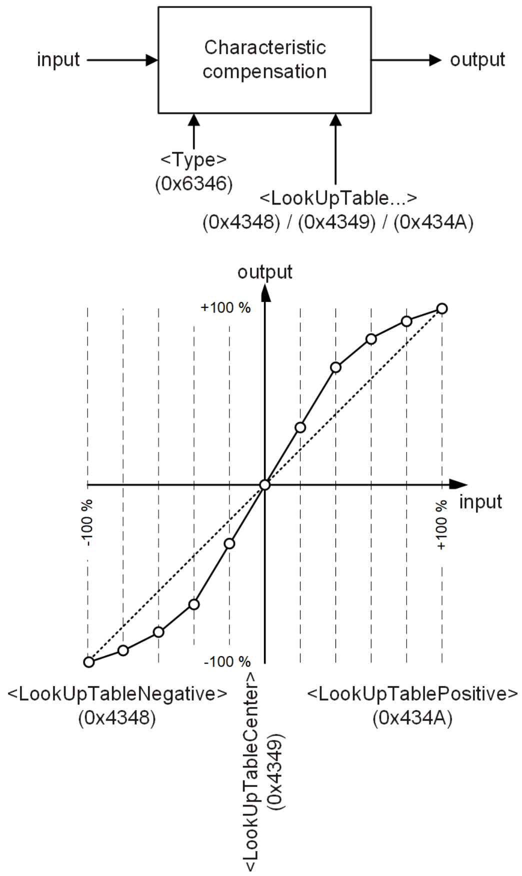


Figure 68: Characteristic compensation

7.3.4.4.1 Object 0x6346: Actuator conditioning characteristic compensation type

This parameter switches the characteristic compensation function on or off.

Valve_ActuatorConditioning_CharacteristicCompensation							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6346	0	Type	INT8	rw	Y	−1…0	0

Value description

Table 81: Possible values of parameter <Type> (0x6346)

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

7.3.4.4.2 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 <Type> (0x6346) set to −1.

Index	Sub-index	Input values (fixed values)	E.g. linear output values (parameter values)
<LookUpTable> (0x4348)	1	-16384	-16384
<LookUpTable> (0x4348)	2	-16256	-16256
<LookUpTable> (0x4348)	3…127	…	…
<LookUpTable> (0x4348)	128	-128	-128
<LookUpTable> (0x4349)	0	0	0
<LookUpTable> (0x434A)	1	128	128
<LookUpTable> (0x434A)	2	256	256
<LookUpTable> (0x434A)	3…127	…	…
<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 <LookUpTable> (0x4348), sub-index 1. When a value is greater than 16384, the output value equals the value of sampling point <LookUpTable> (0x434A), sub-index 128.

7.3.4.4.2.1 Object 0x4347: Look-up table

This parameter is for Moog internal use only.

Valve_ActuatorConditioning_CharacteristicCompensation							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x4347	0	LookUpTable	DOMAIN	rw	Y	DOMAIN	

7.3.4.4.2.2 Object 0x4348: Look-up table

This object contains the output values corresponding to the negative input values.

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

7.3.4.4.2.3 Object 0x4349: Look-up table

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

Valve_ActuatorConditioning_CharacteristicCompensation							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x4349	0	LookUpTableCenter	INT16	rw	Y	INT16	0

7.3.4.4.2.4 Object 0x434A: Look-up table

This object contains the output values corresponding to the positive input values.

Valve_ActuatorConditioning_CharacteristicCompensation							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x434A	1…128	LookUpTablePositive	INT16	rw	Y	INT16	0

7.3.4.5 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 <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 <ASide> (0x6343) and <BSide> (0x6344).

The parameter <ASide> (0x6343) specifies the step height on the positive side and the parameter <BSide> (0x6344) the step height on the negative side. The border where the dead band compensation is effective is set by the parameter <Threshold> (0x6345).

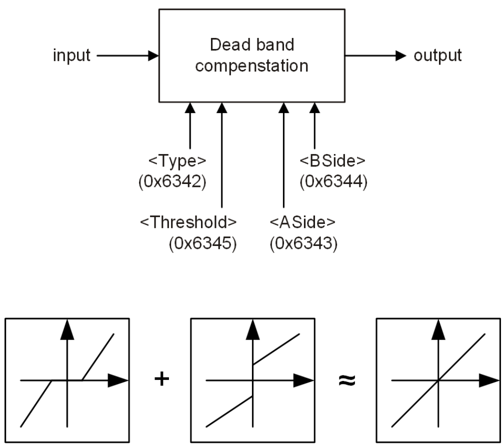


Figure 69: Dead band compensation

7.3.4.5.1 Object 0x6342: Actuator conditioning dead band compensation type

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

Valve_ActuatorConditioning_DeadbandCompensation							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6342	0	Type	INT8	rw	Y	0…2	0

Value description

Table 82: Possible values of parameter <Type> (0x6342)

<Type>	Description
0	Dead band compensation switched off.
1	Dead band jump function.
2	Dead band continuous function (recommended).

7.3.4.5.2 Object 0x6343: Actuator conditioning dead band compensation A side

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

Valve_ActuatorConditioning_DeadbandCompensation							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6343	1	ASide	INT16	rw	Y	0…16384	0
0x6343	2	Unit	UINT8	ro	-	UINT8	0
0x6343	3	Prefix	INT8	ro	-	-4…0	0

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

7.3.4.5.3 Object 0x6344: Actuator conditioning dead band compensation B side

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

Valve_ActuatorConditioning_DeadbandCompensation							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6344	1	BSide	INT16	rw	Y	0…16384	0
0x6344	2	Unit	UINT8	ro	-	UINT8	0
0x6344	3	Prefix	INT8	ro	-	-4…0	0

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

7.3.4.5.4 Object 0x6345: Actuator conditioning dead band compensation threshold

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

Valve_ActuatorConditioning_DeadbandCompensation							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6345	1	Threshold	INT16	rw	Y	0…16383	0
0x6345	2	Unit	UINT8	ro	-	UINT8	0
0x6345	3	Prefix	INT8	ro	-	-4…0	0

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

7.3.4.6 Jump function (dead band compensation type 1)

If the input signals keep within the limits of the threshold value (set with parameter <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 <ASide> (0x6343) or <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 <Type> (0x6342) set to 1 (dead band jump function).

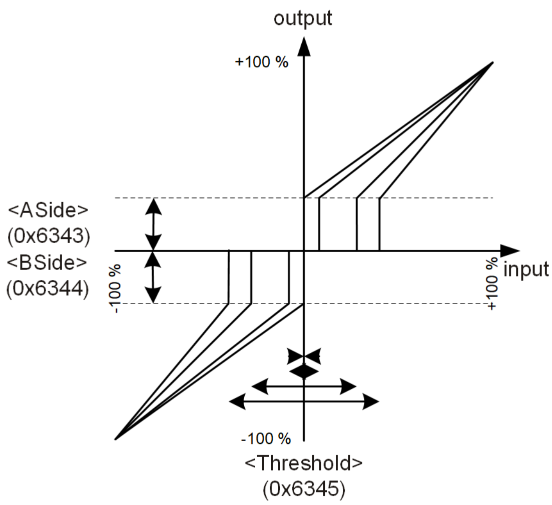


Figure 70: Dead band compensation type 1

7.3.4.7 Continuous function (dead band compensation type 2)

If the input signals keep within the range of the threshold value (set with parameter <Threshold>, 0x6345), the output values will be interpolated between zero and the values defined by the parameters <ASide> (0x6343) or <BSide> (0x6344), respectively. At the threshold position the output signal is increased or decreased by the values set for the <ASide> (0x6343) or <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 <Type> (0x6342) set to 2 (dead band continuous function).

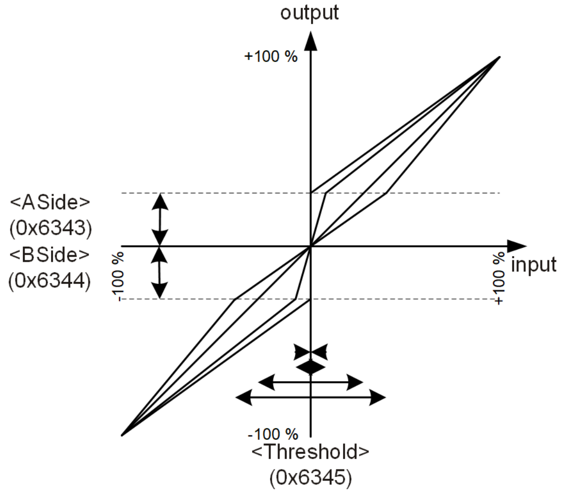


Figure 71: Dead band compensation type 2

7.3.4.8 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.

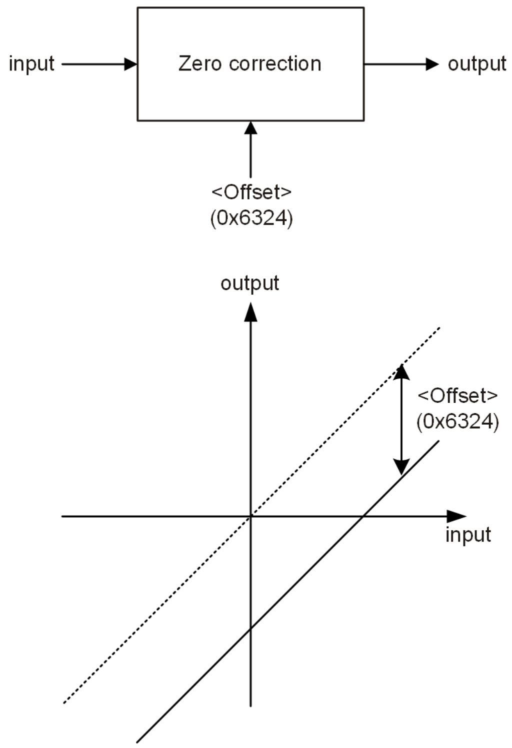


Figure 72: Zero correction

7.3.4.8.1 Object 0x6324: Actuator conditioning zero correction offset

Valve_ActuatorConditioning_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	-	-4...0	0

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

7.3.5 Pilot spool position actuator conditioning

The pilot spool position actuator conditioning only includes a zero correction.

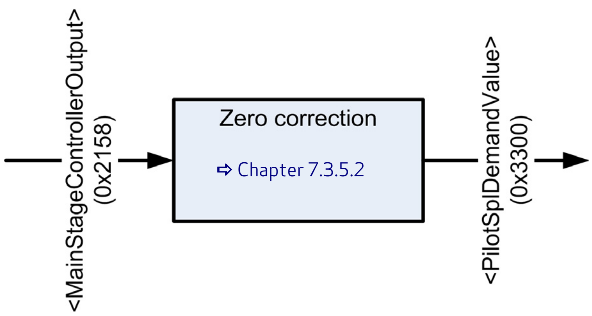


Figure 73: Pilot spool position actuator conditioning

➪ Chapter "7.3.8.3 Object 0x2158: Main stage controller output", page 179

7.3.5.1 Object 0x3300: Pilot spool position demand value

This object shall provide the output of the pilot spool position actuator conditioning.

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x3300	1	PltSplDemandValue	INT16	ro	-	INT16	None

7.3.5.2 Zero correction

The zero correction enables shifting of the input signal up and down by any desired offset. The <OffsetForPilotInDualStageMode> (0x242E) is added to the input signal according to the following figure.

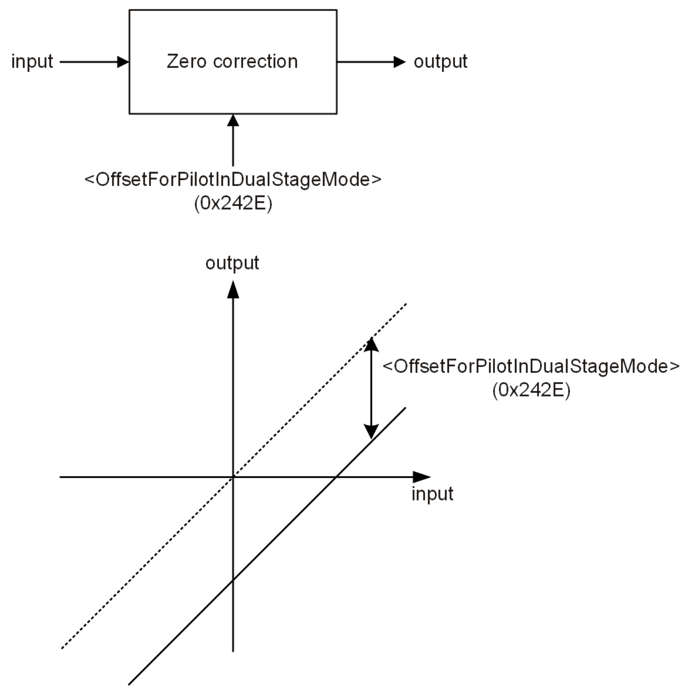


Figure 74: Zero correction

7.3.5.2.1 Object 0x242E: Offset for pilot spool position in dual stage mode

This parameter allows an offset to the pilot valve spool position <PilotSplDemandValue> (0x3300). This offset may help for example in case of a temperature drift or another drift.

Valve_ActuatorConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x242E	0	OffsetForPilotInDualStageMode	INT16	rw	Y	INT16	0

7.3.6 Spool position / pilot spool position controller

For a single stage servo valve this spool position controller controls the spool position. The parameter <SplControlDeviation> (0x6350) holds the spool position control deviation.

For a dual stage servo valve, this spool controller is used to control the pilots spool position. The main stage is controlled by the main stage spool position controller. The parameter <SplControlDeviation> (0x6350) holds the main stage spool position control deviation. The parameter <PilotControlDeviation> (0x3302) holds the pilot stage spool control deviation.

7.3.6.1 Object 0x6350: Control deviation

The control deviation is the difference between the setpoint and the actual value. In case of a dual stage servo valve this parameter holds the control deviation of the main stage spool position. In case of the single stage servo valve this parameter holds the control deviation of the pilot 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	-	−4…0	0

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

7.3.6.2 Object 0x3302: Pilot control deviation

The pilot control deviation is the difference between the setpoint 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).

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

7.3.6.3 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	-

7.3.6.4 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	REAL32	rw	Y	0.0…2.0	1.0

7.3.7 Main stage spool position actual value path

If a dual stage servo valve is used, the main stage spool position actual value is scaled and mapped to the signal <MainStagePositionActualValue> (0x215B) and the signal <SpoolActualValue> (0x6301).

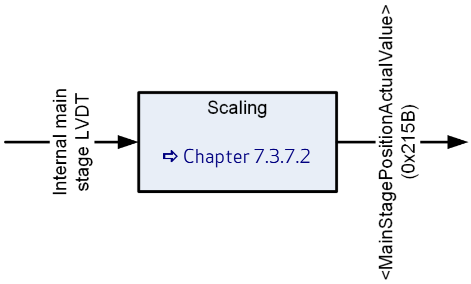


Figure 75: Main stage spool position actual value path

⇨ Chapter "7.3.3.1 Object 0x6301: Spool position actual value", page 164

7.3.7.1 Object 0x215B: Main stage position actual value

This parameter holds the main stage spool position actual value.

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x215B	1	MainStagePositionActualValue	INT16	ro	-	INT16	None

7.3.7.2 Object 0x3237: Customer scaling of main stage spool position sensor

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

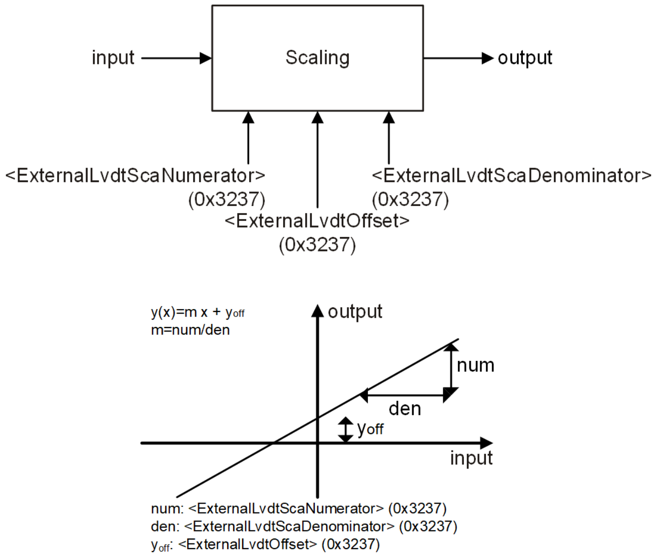


Figure 76: Customer scaling of main stage spool position sensor

ExternalLVDT							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	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

7.3.8 Main stage spool position controller

The main stage spool position controller will be only used for dual stage servo valves. The main stage spool position control deviation and the main stage spool position controller output can be accessed by corresponding output parameters.

➪ Chapter "7.3.6.1 Object 0x6350: Control deviation", page 176

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	REAL32	rw	Y	0.0…2.0	1.0

7.3.8.2 Object 0x2171: Main stage controller integral part output

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

ValveMainStageControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2171	0	MainStageIntegralPart	INT32	ro	-	INT32	None

7.3.8.3 Object 0x2158: Main stage 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 p setpoint conditioning

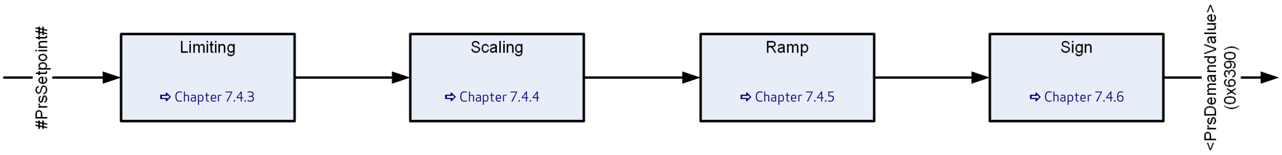


Figure 77: Pressure setpoint conditioning

!	The internal signal #PrsSetpoint# is used to link the pressure setpoint to the pressure setpoint conditioning. ➪ Chapter "6.1.4 p setpoint path", page 67

7.4.1 Object 0x6390: Setpoint conditioning demand value

The demand value indicated by this parameter is generated from the #PrsSetpoint# by means of the functions in the setpoint conditioning and forwarded to the pressure controller.

ValvePressureControl_SetpointConditioning							
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	-	−4…0	0

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

7.4.2 Object 0x6391: Setpoint conditioning reference value

The reference value corresponds to 100 % of the p input signal. This means that a 100 % p 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 read by the fieldbus master to scale the setpoint. The reference value is a read only object.

ValvePressureControl_SetpointConditioning							
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	-	−4…0	0

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

7.4.3 Limiting 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 "5.1.3.2 Object 0x6041: Status word", page 57

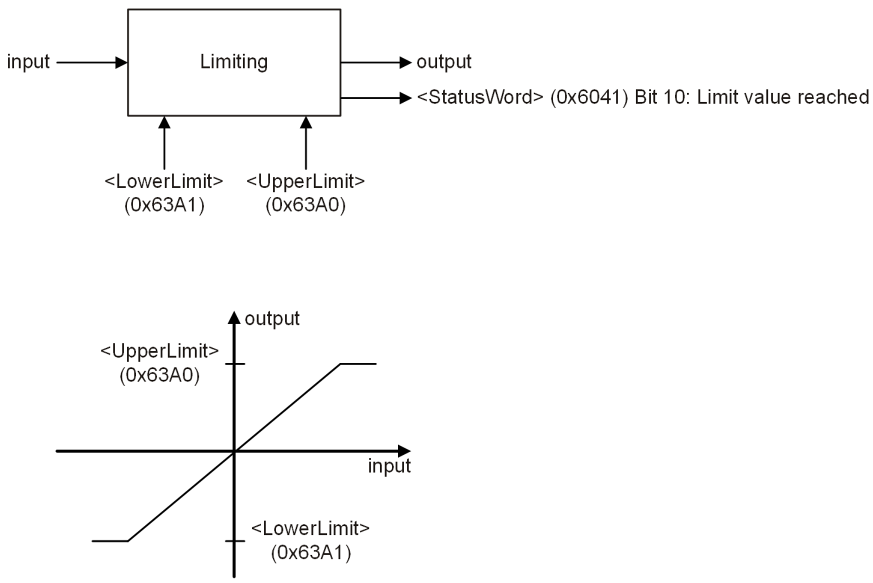


Figure 78: Limiting function

❗	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: Setpoint conditioning upper setpoint limit

ValvePressureControl_SetpointConditioning_SetpointLimit							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63A0	1	UpperLimit	INT16	rw	Y	‹LowerLimit› (0x63A1)…32767	16384
0x63A0	2	Unit	UINT8	ro	-	UINT8	0
0x63A0	3	Prefix	INT8	ro	-	-4…0	0

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

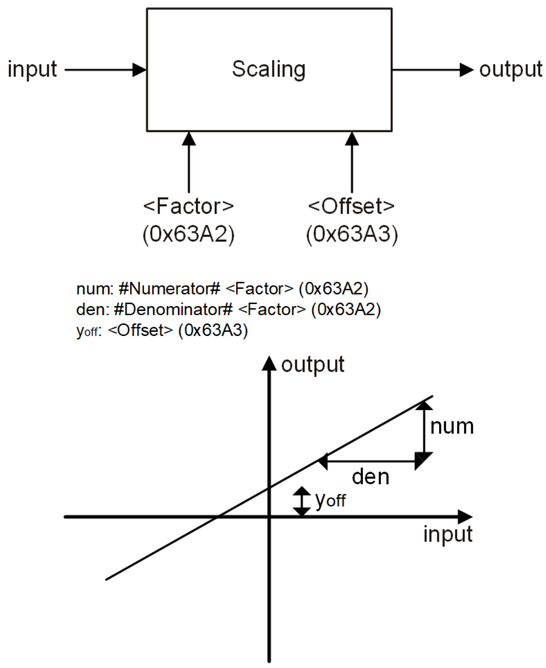
7.4.3.2 Object 0x63A1: Setpoint conditioning lower setpoint limit

ValvePressureControl_SetpointConditioning_SetpointLimit							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x63A1	1	LowerLimit	INT16	rw	Y	-32767…‹UpperLimit› (0x63A0)	-16384
0x63A1	2	Unit	UINT8	ro	-	UINT8	0
0x63A1	3	Prefix	INT8	ro	-	-4…0	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.



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

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

Figure 79: Scaling function

7.4.4.1 Object 0x63A2: Setpoint conditioning scaling factor

This parameter is a slope factor by which the input is multiplied. It is defined by two signed integer values, the numerator (represented by the upper 16 bits of the parameter) and the denominator (represented by the lower 16 bits of the parameter).

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

Value description

Table 83: Data structure of the slope factor

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

7.4.4.2 Object 0x63A3: Setpoint conditioning scaling offset

This parameter is the offset of the function.

ValvePressureControl_SetpointConditioning_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	-	-4...0	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-, two- or four-quadrant ramp function or to deactivate it.

If the signal slope of the input signal is limited by configured slew rate (ramp running), the <StatusWord> (0x6041) bit 9 is set.

If the running ramp is stopped by the bit 15 (ramp stop) of the #ControlWord# signal, the <StatusWord> (0x6041) bit 15 signalizes that the output value of the ramp signal is hold (ramp frozen).

➪ Chapter "5.1.3.2 Object 0x6041: Status word", page 57
➪ Chapter "5.1.5 Object 0x604F: Device local", page 59
➪ Chapter "5.1.3.1 Object 0x6040: Device control word", page 56
➪ Chapter "5.1.6 Object 0x4040: Local control word", page 59

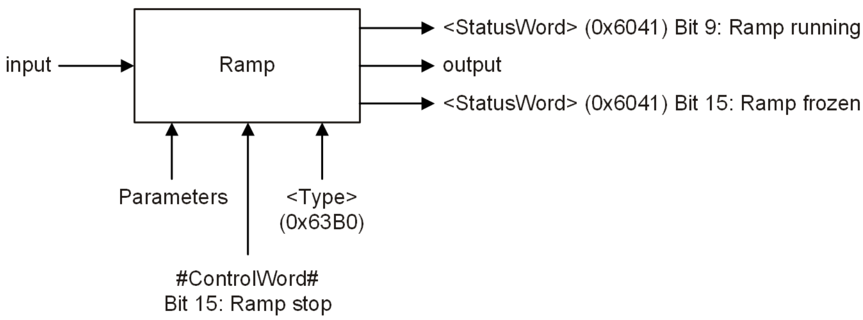


Figure 80: Ramp function

7.4.5.1 Object 0x63B0: Setpoint conditioning ramp type

This parameter defines the progression of the ramp.

ValvePressureControl_SetpointConditioning_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

Table 84: Possible values of parameter <Type> (0x63B0)

<Type>	Description
0	No ramp
1	One-quadrant ramp
2	Two-quadrant ramp
3	Four-quadrant ramp

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.

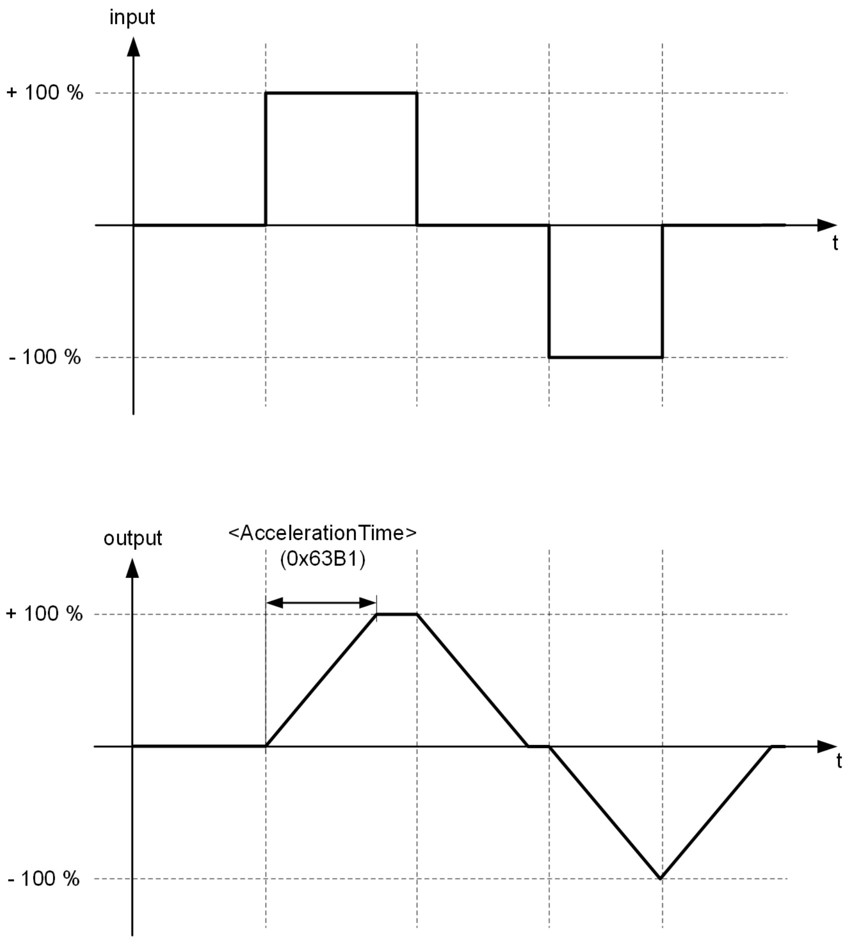


Figure 81: Ramp type 1

7.4.5.2.1 Object 0x63B1: Setpoint conditioning ramp 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_SetpointConditioning_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	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 <DecelerationTime> (0x63B4).

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

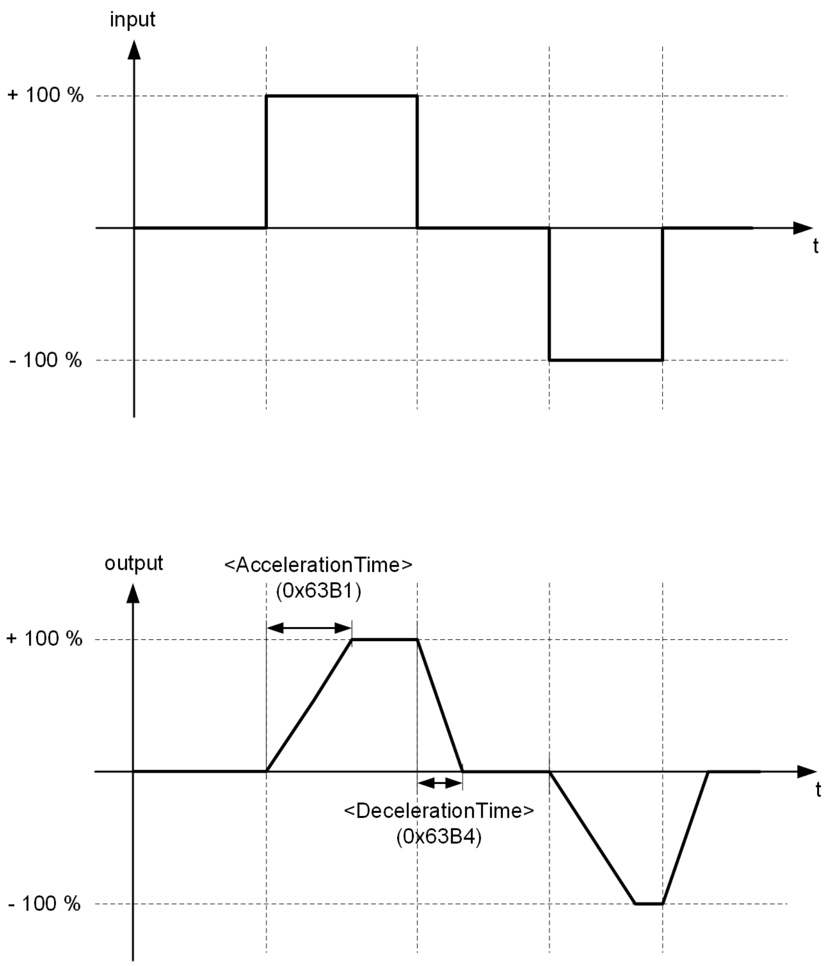


Figure 82: Ramp type 2

7.4.5.3.1 Object 0x63B1: Setpoint conditioning ramp acceleration time

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

➪ Chapter "7.4.5.2.1 Object 0x63B1: Setpoint conditioning ramp acceleration time", page 184

7.4.5.3.2 Object 0x63B4: Setpoint conditioning 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_SetpointConditioning_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	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.

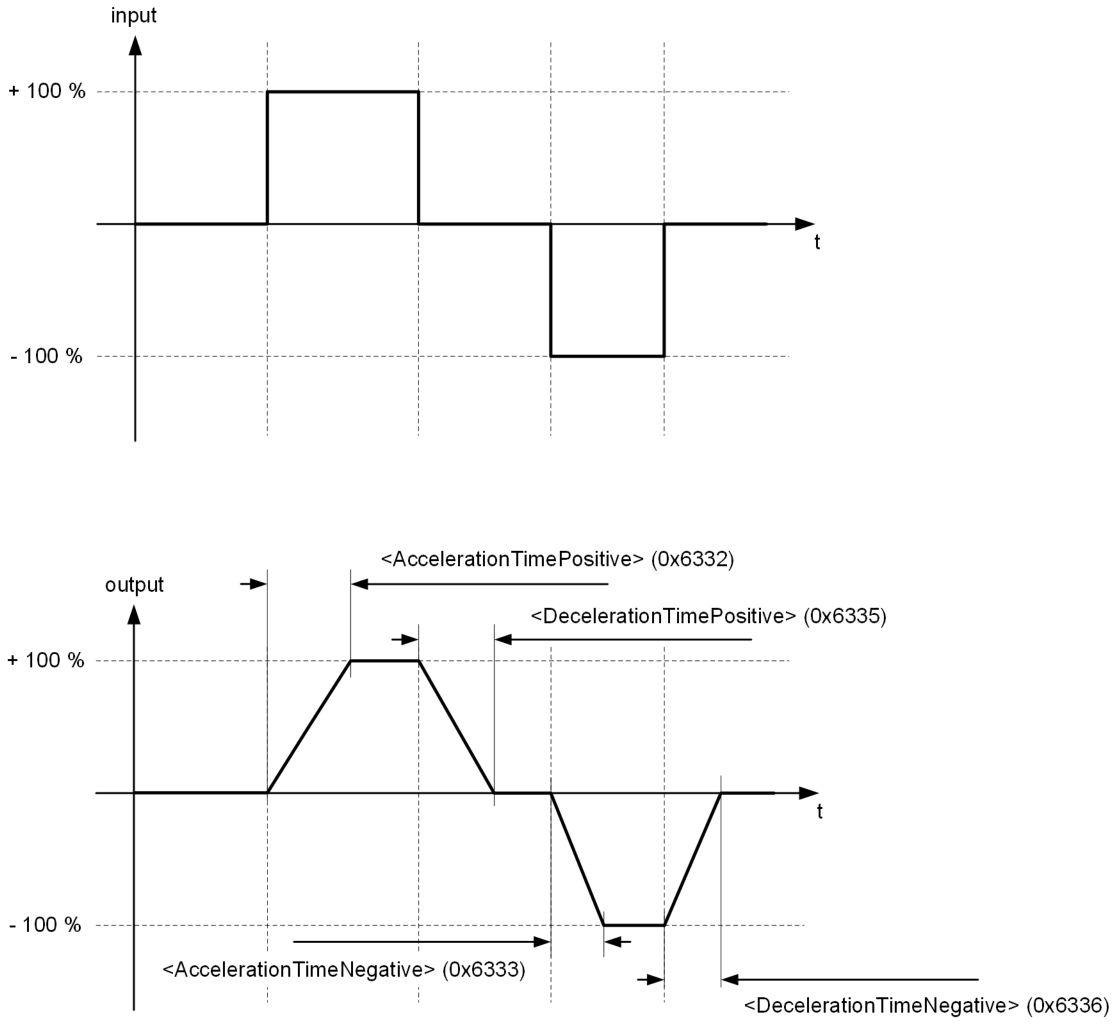


Figure 83: Ramp type 3

7.4.5.4.1 Object 0x63B2: Setpoint conditioning ramp 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_SetpointConditioning_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	Prefix	INT8	rw	Y	-4…0	-3

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

7.4.5.4.2 Object 0x63B3: Setpoint conditioning ramp 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_SetpointConditioning_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	Prefix	INT8	rw	Y	-4…0	-3

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

7.4.5.4.3 Object 0x63B5: Setpoint conditioning ramp 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_SetpointConditioning_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	Prefix	INT8	rw	Y	-4…0	-3

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

7.4.5.4.4 Object 0x63B6: Setpoint conditioning ramp 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_SetpointConditioning_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	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 (<ControlMode> (0x6043) is set to 5), the following structure calculates the polarity of the pressure demand value. Negative pressure setpoint means that the pressure setpoint is effective at port B.

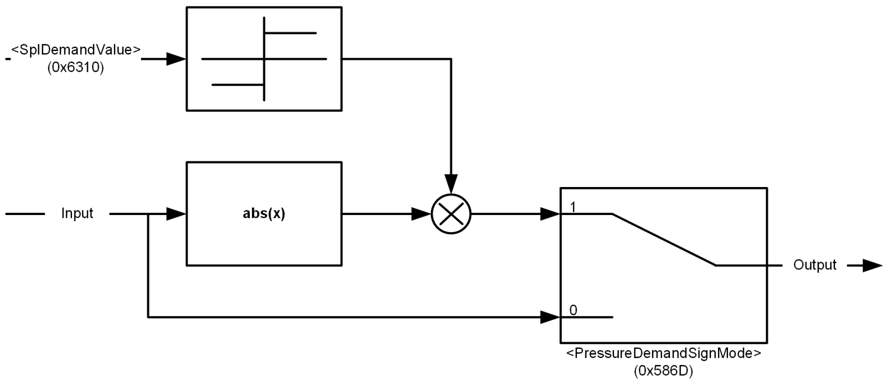


Figure 84: Pressure demand signal sign

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.

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

Table 85: Possible values of parameter <PressureDemandSignMode> (0x586D)

<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.

7.5 Pressure controller

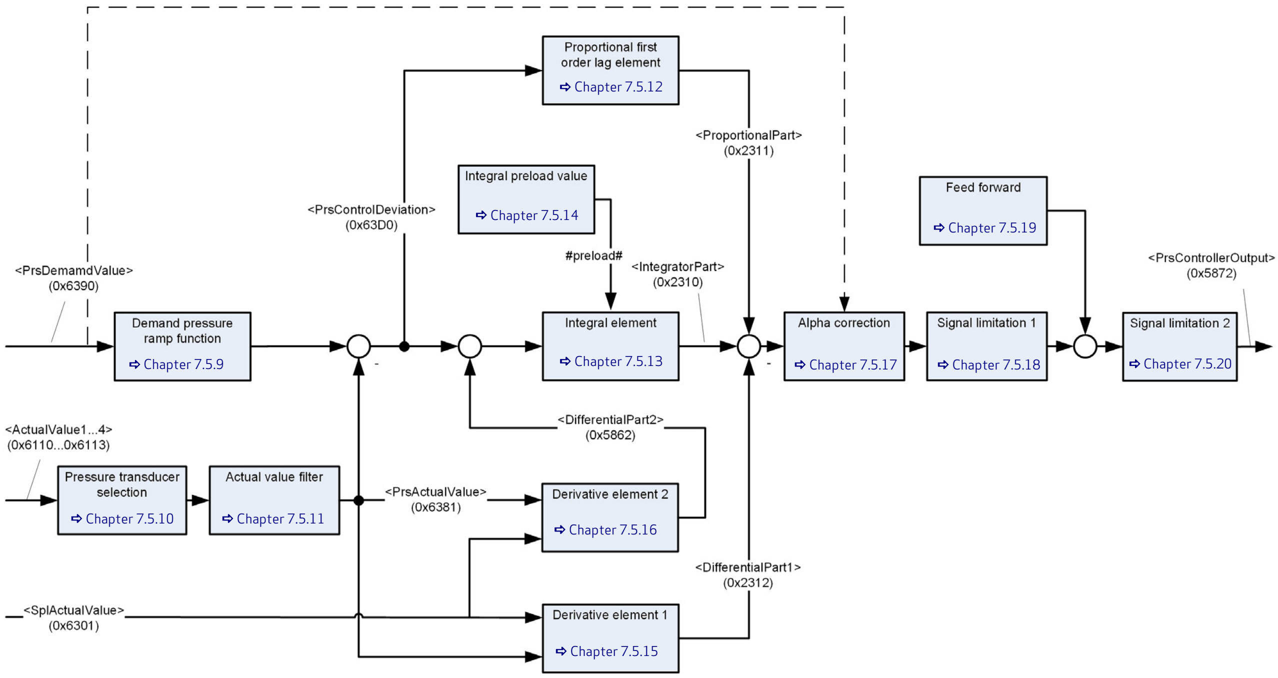


Figure 85: Pressure controller

➪ Chapter "7.4.1 Object 0x6390: Setpoint conditioning demand value", page 179
➪ Chapter "7.5.2 Object 0x63D0: Pressure control deviation", page 190
➪ Chapter "7.5.5 Object 0x2312: Differential part 1", page 190
➪ Chapter "7.5.6 Object 0x5862: Differential part 2", page 190
➪ Chapter "7.5.4 Object 0x2310: Integrator part", page 190
➪ Chapter "7.5.3 Object 0x2311: Proportional part", page 190
➪ Chapter "7.5.7 Object 0x5872: Pressure controller output", page 191

Description of the feedback signal <ActualValue1…4> (0x6110…0x6113):
➪ Chapter "6.2 Actual value transducer interface", page 72

7.5.1 Object 0x6381: Pressure actual value

The filter output <PrsActualValue> (0x6381) represents the pressure actual value. This value is used as feedback input for 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	-	-4…0	0

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

7.5.2 Object 0x63D0: Pressure control deviation

This parameter holds the deviation between the ramped pressure setpoint 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	-	−4…0	0

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

7.5.3 Object 0x2311: Proportional part

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	ProportionalPart	REAL32	ro	-	REAL32	None

7.5.4 Object 0x2310: Integrator part

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	IntegratorPart	REAL32	ro	-	REAL32	None

7.5.5 Object 0x2312: Differential part 1

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	DifferentialPart1	REAL32	ro	-	REAL32	None

7.5.6 Object 0x5862: Differential part 2

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	DifferentialPart2	REAL32	ro	-	REAL32	None

7.5.7 Object 0x5872: Pressure 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
0x5872	0	PrsControllerOutput	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.

Table 86: Pressure controller objects contained in a 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

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	UINT8	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.

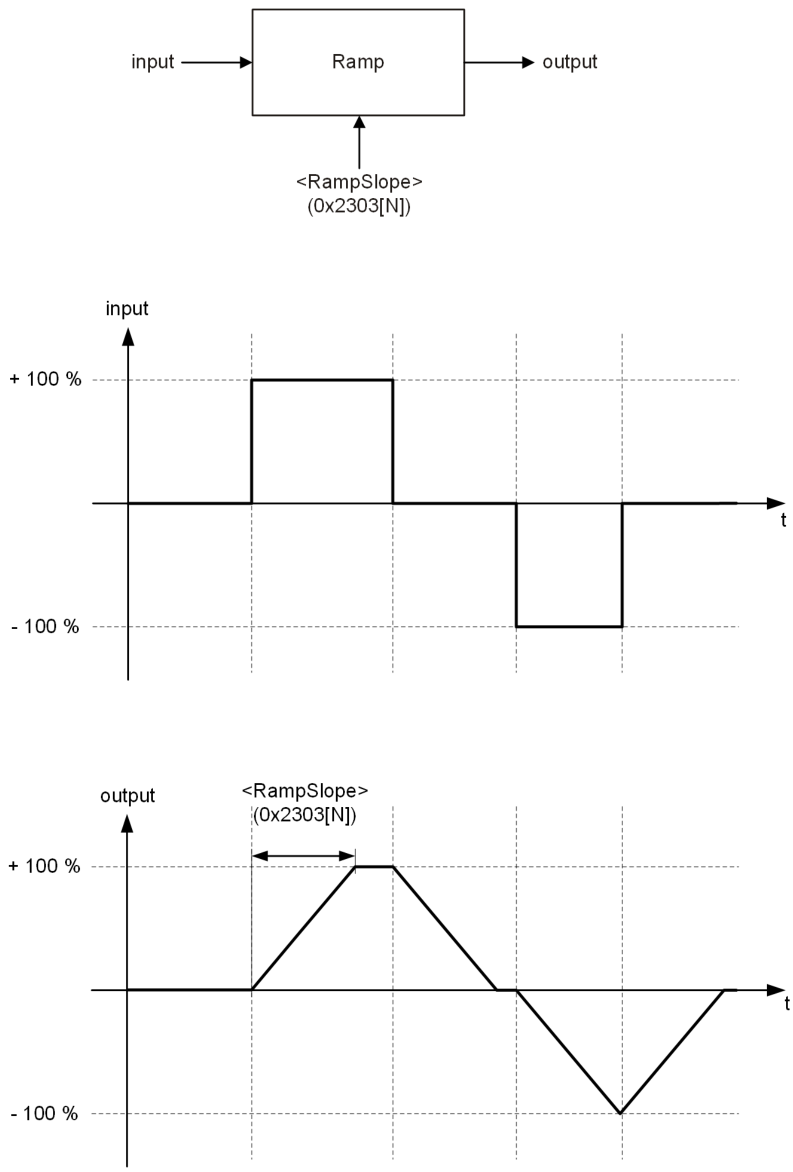


Figure 86: Demand pressure ramp function

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, used as feedback signal for the pressure controller, will be routed through the four possible transducer interfaces. Three different actual value pressure calculation, depending on the pressure transducer selection and the pressure chamber mode switch <PressureChamberModeSwitch> (0x233F) are possible:

- Normal pressure control
  Only one pressure signal from the transducer interface <ActiveTransducerInterfaceAreaA> (0x230D[N]) is fed to the pressure controller. The second interface, <ActiveTransducerInterfaceAreaB> (0x230F[N]) is set to zero.
- Differential pressure control with alpha correction
  Two pressure signals from the transducer interfaces used for the alpha correction and fed to the pressure controller. 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.
- Differential pressure control with flow sign depending pressure routing
  Depending on the sign of the demand flow signal, either the signal from the transducer interface <ActiveTransducerInterfaceAreaA> (0x230D[N]) or the signal from the <ActiveTransducerInterfaceAreaB> (0x230F[N]) negated is used as an input signal for the pressure controller. This mode is activated by setting the parameter <PressureChamberModeSwitch> (0x233F) to 1.

One from <ActiveTransducerInterfaceAreaA> (0x230D[N]) representing the pressure in the servo valve port A and one from the transducer interface <ActiveTransducerInterfaceAreaB> (0x230F[N]) representing the pressure in the servo valve port B.

➪ Chapter "7.6.1 Object 0x6710: Setpoint conditioning demand value", page 209
➪ Chapter "7.5.1 Object 0x6381: Pressure actual value", page 189

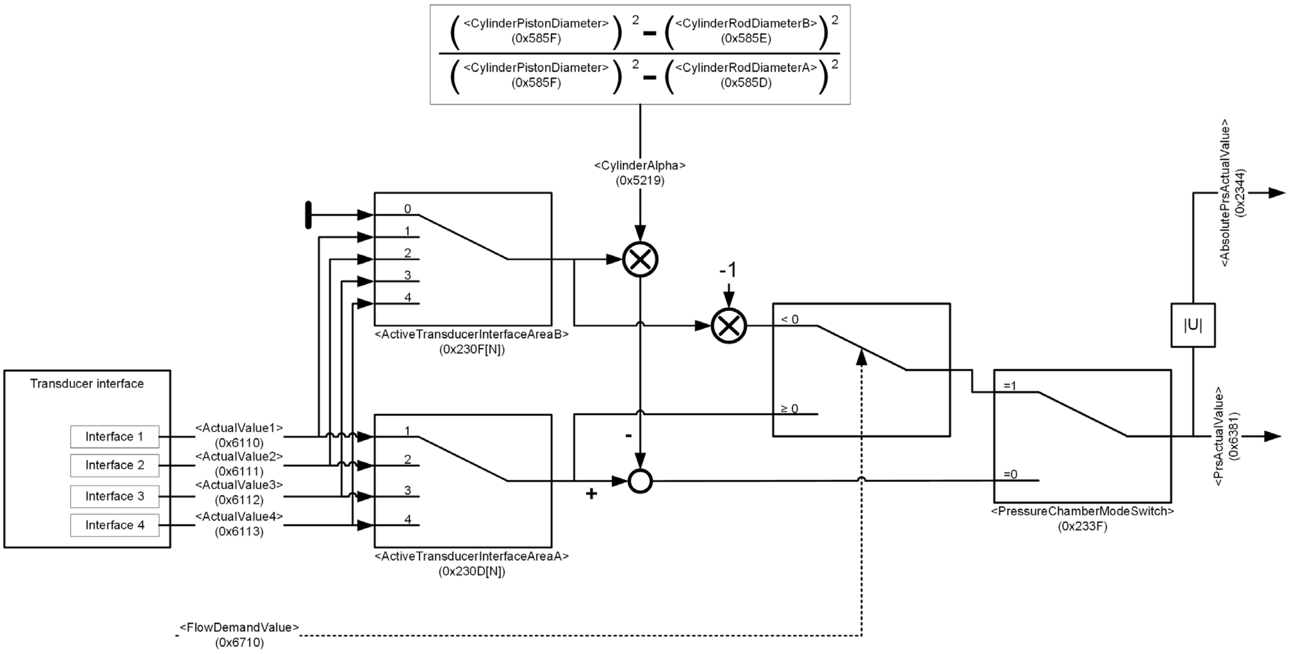


Figure 87: Pressure transducer selection

❗	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	Persistence	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	REAL32	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	REAL32	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	REAL32	rw	Y	0.0…<CylinderPistonDiameter> (0x585F)	0.0

7.5.10.6 Object 0x5219: Cylinder alpha

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x5219	0	CylinderAlpha	REAL32	ro	-	REAL32	None

7.5.10.7 Object 0x233F: Pressure chamber mode switch

With this parameter the calculation mode for the pressure controller actual value can be switched.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x233F	0	PressureChamberModeSwitch	UINT8	rw	Y	0…1	0

Value description

Table 87: Possible values of parameter <PressureChamberModeSwitch> (0x233F)

<PressureChamberModeSwitch>	Description
0	Differential pressure control with alpha correction
1	Differential pressure control with flow sign depending pressure routing

7.5.10.8 Object 0x2344: Absolute pressure actual value

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2344	0	AbsolutePrsActualValue	INT16	ro	-	INT16	None

7.5.11 Actual pressure 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).

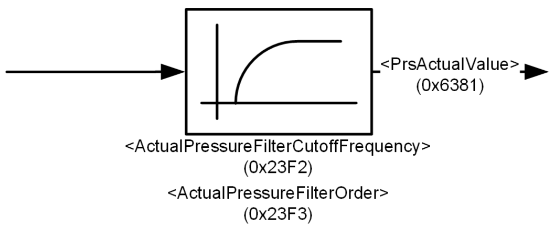


Figure 88: Actual value filter

7.5.11.1 Object 0x23F0: Actual pressure value filter coeff B

This parameter specifies the coefficient B of the Butterworth filter.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x23F0	1	PressureValueFilterBCoeff	REAL32	rw	Y	REAL32	0.0

7.5.11.2 Object 0x23F1: Actual pressure value filter coeff A

This parameter specifies the coefficient A of the Butterworth filter.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x23F1	1	PressureValueFilterACoeff	REAL32	rw	Y	REAL32	0.0

7.5.11.3 Object 0x23F2: Actual pressure value 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	REAL32	rw	Y	0.0…5000.0	0.0

7.5.11.4 Object 0x23F3: Actual pressure value filter order

This parameter sets the order of the Butterworth filter. Setting this parameter to 0 deactivates the filter.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x23F3	0	ActualPressureFilterOrder	UINT8	rw	Y	0…3	0

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.

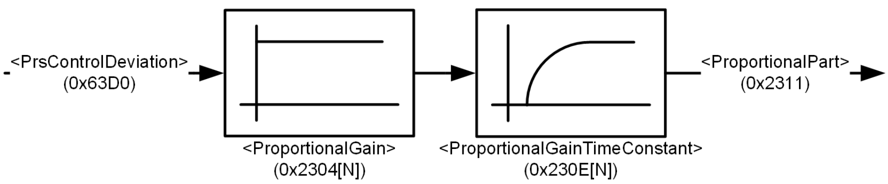


Figure 89: Proportional first order lag element (PPT1)

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	RELA32	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	RELA32	rw	Y	0.0…+inf	0.0

7.5.13 Integrator element (I)

To ensure a bumpless transfer between spool position and pressure control, the pressure integrator can be set to a defined preload (#Preload#) value before switching to the pressure control.

➪ Chapter "7.5.14 Integrator preload value", page 201

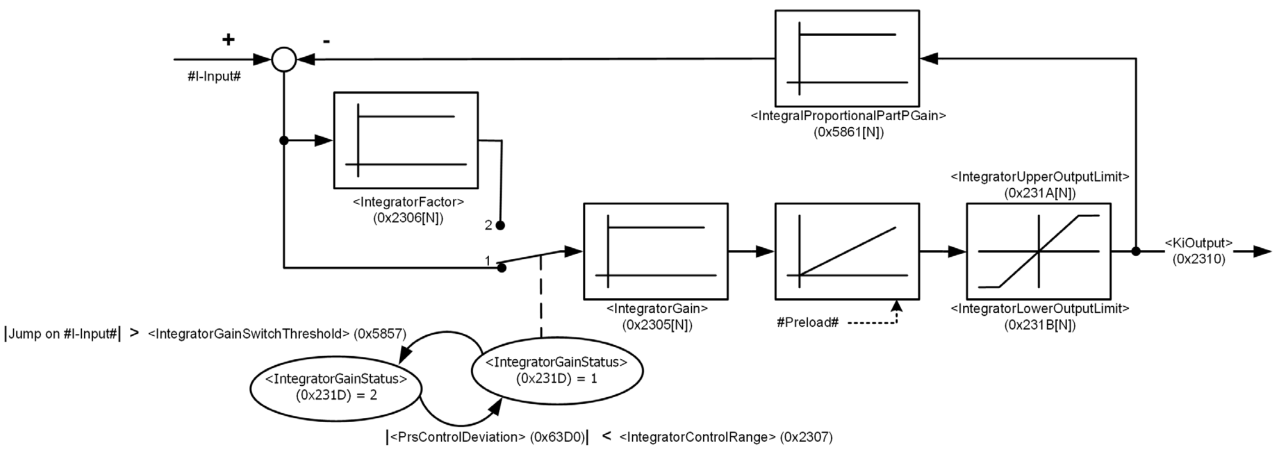


Figure 90: Integrator element (I)

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	RELA32	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	RELA32	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…32767	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	ro	-	1…2	None

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 ?s -> 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	REAL32	rw	Y	REAL32	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. Former value means the value of the <IntegratorPreloadMode> (0x586B) before setting the value to 2 (usually to default value 1). This function is comparable to a push-button.

➪ Chapter "7.5.13 Integrator element (I)", page 198

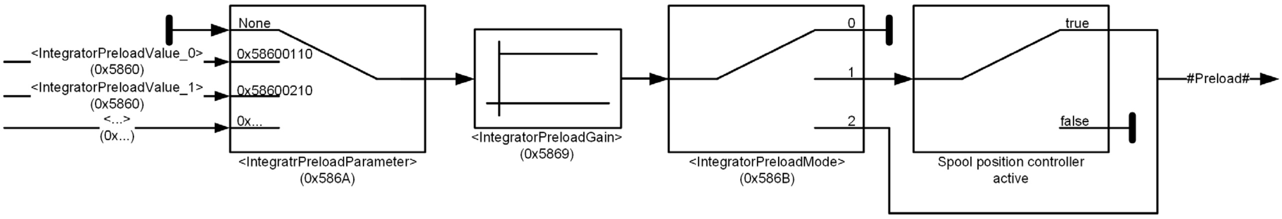


Figure 91: Integrator preload value

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

Table 88: Possible values of parameter <IntegratorPreloadMode> (0x586B)

<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).

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	REAL32	rw	Y	REAL32	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	IntegratorPreloadMappingParameter	UINT32	rw	Y	UINT32	0x63800110

Value description

<IntegratorPreloadMappingParameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length: 0x10 or 0x20
Default	0x63	0x80	0x01	0x10

This reference 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 or 0x20.

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	IntegratorPreloadValue1	INT16	ro	-	INT16	None
0x5860	2	IntegratorPreloadValue2	INT16	ro	-	INT16	None

Value description

Table 89: Behavior of preload output

<Parameter>	Behavior of preload output
<IntegratorPreloadValue1>	<SplDemandValue> (0x6310) minus <ProportionalPart> (0x2311)
<IntegratorPreloadValue2>	<SplDemandValue> (0x6310) minus <ProportionalPart> (0x2311) minus <FeedForwardOffset> (0x5870[N])

7.5.15 Derivative element 1 (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.

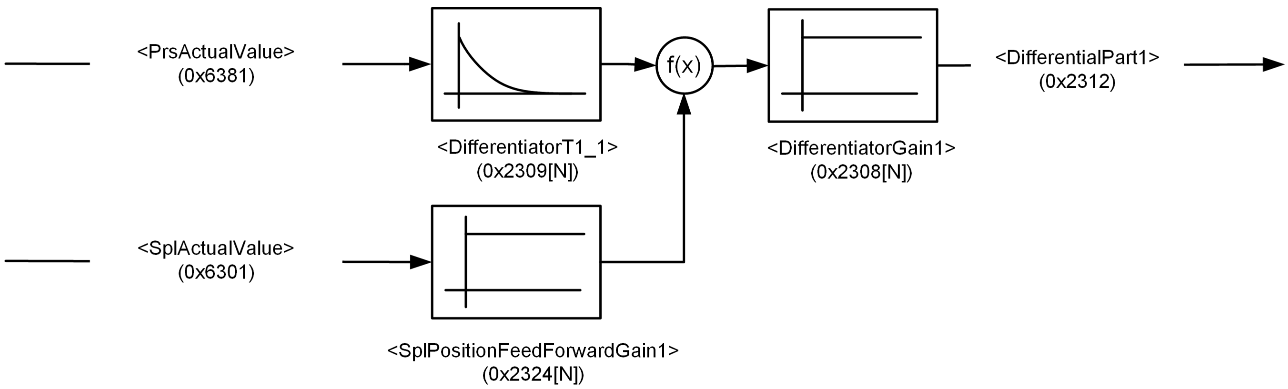


Figure 92: Proportional derivative element 1 (PD)

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	DifferentiatorGain1	REAL32	rw	Y	REAL32	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_1	REAL32	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	SplPositionFeedForwardGain1	REAL32	rw	Y	0.0…+inf	0.0

7.5.16 Feedback derivative element 2 (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.

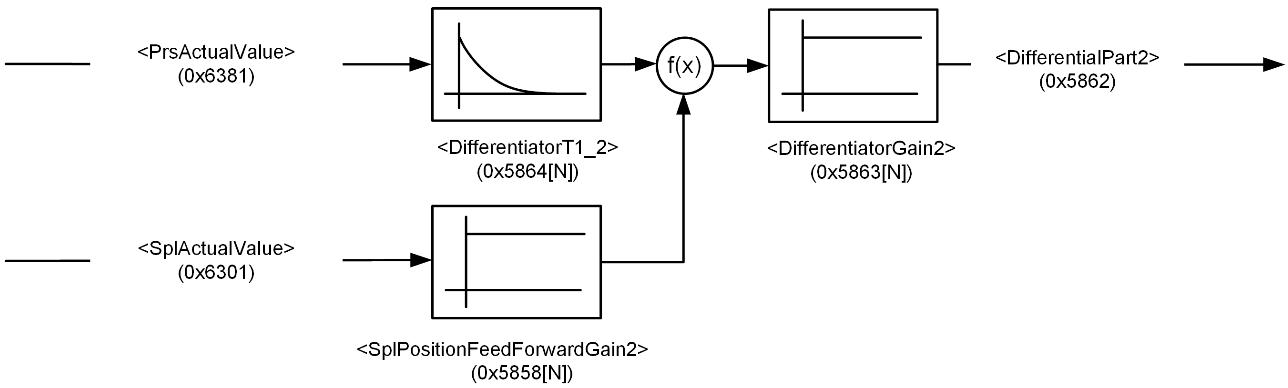


Figure 93: Proportional derivative element 2 (PD)

7.5.16.1 Object 0x5863[N]: Differentiator gain

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	REAL32	rw	Y	REAL32	0.0

7.5.16.2 Object 0x5864[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
0x5864	N=1…16	DifferentiatorT1_2	REAL32	rw	Y	0.0…+inf	0.0

7.5.16.3 Object 0x5858[N]: Spool position feed forward gain

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	SplPositionFeedForwardGain2	REAL32	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 <PrsDemandValue> (0x6390) is lower than zero.

➪ Chapter "7.4.1 Object 0x6390: Setpoint conditioning demand value", page 179
➪ Chapter "7.5.10.3 Object 0x585F[N]: Cylinder piston diameter", page 195
➪ Chapter "7.5.10.4 Object 0x585D: Cylinder rod diameter A", page 195
➪ Chapter "7.5.10.6 Object 0x5219: Cylinder alpha", page 196

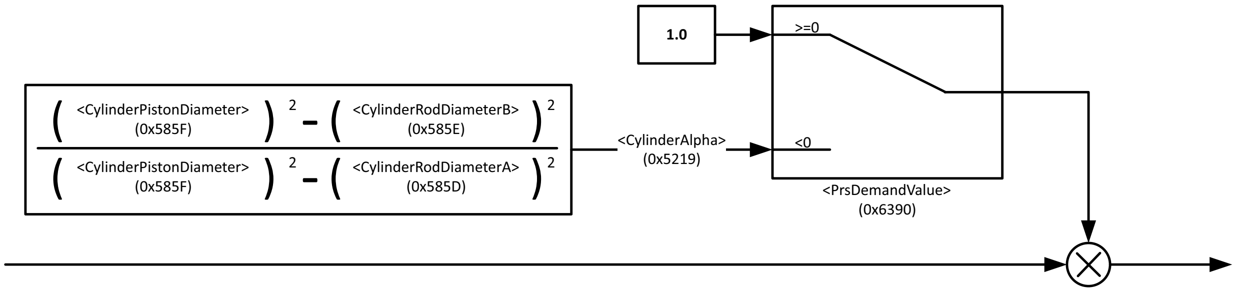


Figure 94: Alpha correction

7.5.18 Signal limitation 1

Signal limitation after the alpha correction.

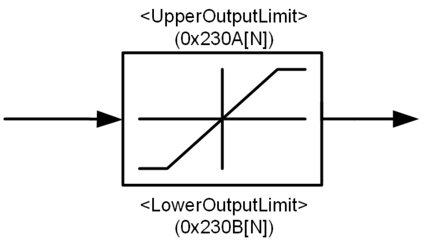


Figure 95: Signal limitation 1

7.5.18.1 Object 0x230A[N]: Upper output limit

This parameter contains the upper limit.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x230A	N=1…16	UpperOutputLimit	INT16	rw	Y	<LowerOutputLimit> (0x230B[N])…32767	16384

7.5.18.2 Object 0x230B[N]: Lower output limit

This parameter contains the lower limit.

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[N])	−16384

7.5.19 Feed forward

For some applications a feed forward control is advantageous. With the feed forward function a selectable signal, e.g., the pressure setpoint, can be forwarded to the output of the controller. The signal can be scaled and an offset can be added.

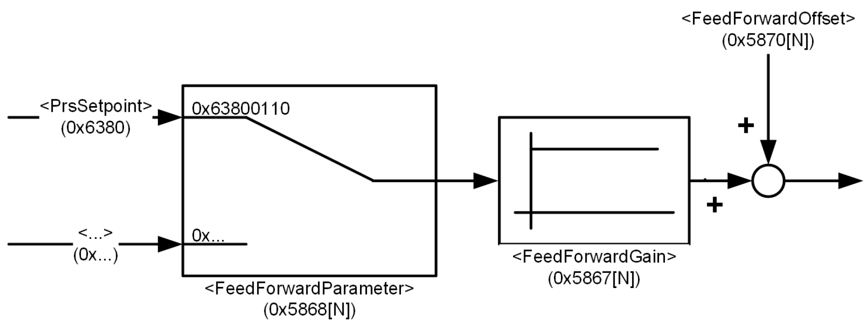


Figure 96: Feed forward

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	REAL32	rw	Y	REAL32	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 <PrsSetpoint> (0x6380) is mapped.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5868	N=1…16	FeedForwardMappingParameter	UINT32	rw	Y	UINT32	0x63800110

Value description

<FeedForwardMappingParameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length: 0x10 or 0x20
Default	0x63	0x80	0x01	0x10

This reference 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 or 0x20.

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.

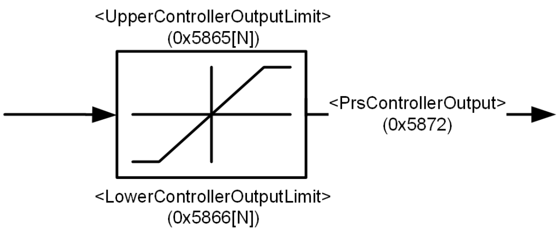


Figure 97: Signal limitation 2

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	<LowerControllerOutputLimit> (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… <UpperControllerOutputLimit> (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:

Table 90: Parameters used in a linear plant model

Parameter name	Description
V_{qu}	Linear gain between setpoint and flow (depends on working point)
D_v	Servo valve damping (depends on setpoint amplitude)
w_v	Servo valve natural frequency in [rad/s]
$C_H = \frac{V}{E_{Oil}}$	Hydraulic capacity C_H [10^{-6} l/bar] with: V Oil in the pipes volume [m^3] E_{Oil} Compressibility module $-1.8*10^{-9}$ [Pa]
K_P	Pressure controller proportional gain (calculation depends on C_H)
K_I	Pressure controller integrator gain (calculation depends on C_H)
K_D	Pressure controller differential gain (calculation depends on C_H)
T_1	Pressure controller differential time constant (calculation depends on C_H)

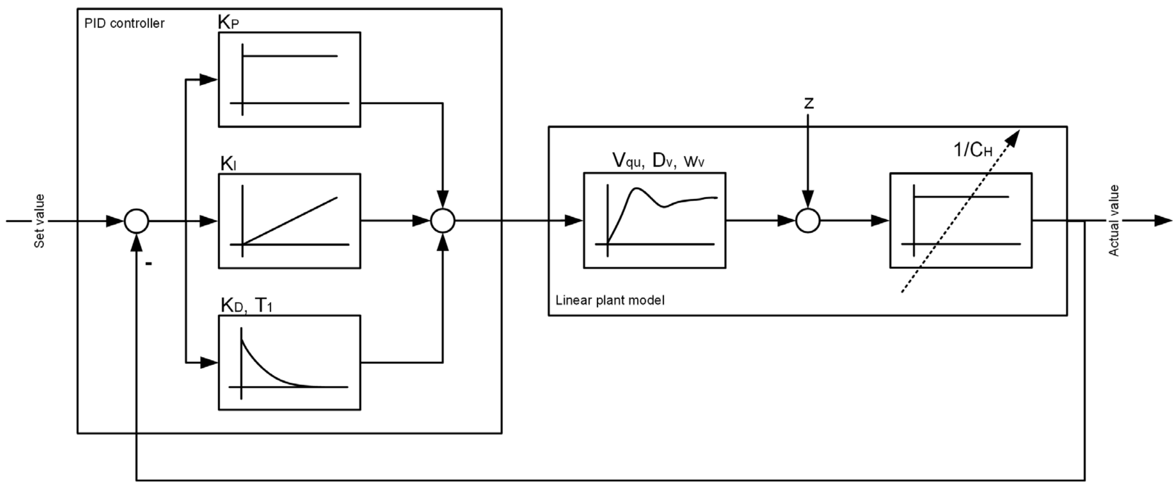


Figure 98: Parameterization of the pressure controller

An applicable way to find controller parameters for a stable system behavior is to use the 'worst case' pressure setpoint for controller tuning. The worst case pressure setpoint 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> (0x320C) is defined as $\frac{V}{E_{Oil}}$. 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	REAL32	rw	Y	0.0…+inf	0.0

7.5.21.2 Object 0x231C: System 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	PressureReference	INT16	rw	N	INT16	400
0x231C	2	Unit	UINT8	ro	-	UINT8	0
0x231C	3	Prefix	INT8	ro	-	-4…0	0

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

7.6 Flow setpoint conditioning

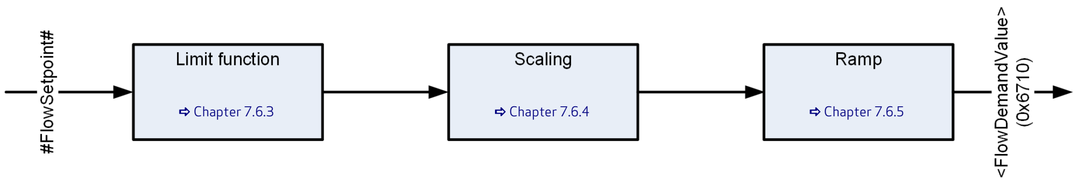


Figure 99: Flow setpoint conditioning

!	The internal signal #FlowSetpoint# is used to link the flow setpoint to the flow setpoint conditioning. ➪ Chapter "7.6 Flow setpoint conditioning", page 209

7.6.1 Object 0x6710: Setpoint conditioning demand value

The demand value indicated by this parameter is generated from the #FlowSetpoint# by means of the functions in the setpoint conditioning and forwarded to the flow controller.

ValveFlowControl_SetpointConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6710	1	FlowDemandValue	INT16	rw	N	INT16	None
0x6710	2	Unit	UINT8	ro	-	UINT8	0
0x6710	3	Prefix	INT8	ro	-	-4…0	0

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

7.6.2 Object 0x6711: Setpoint conditioning reference value

The reference value corresponds to 100 % of the flow input signal. This means that 100 % flow input signal is equal to 16384 increments and −100 % input signal is equal to −16384 increments. This parameter depends on the controller hardware. It can be read by the fieldbus master to scale the setpoint. The reference value is a read only object.

ValveFlowControl_SetpointConditioning							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6711	1	FlowReferenceValue	INT16	rw	N	INT16	16384
0x6711	2	Unit	UINT8	ro	-	UINT8	0
0x6711	3	Prefix	INT8	ro	-	−4…0	0

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

7.6.3 Limiting function

This function limits the value range of the flow 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 "5.1.3.2 Object 0x6041: Status word", page 57

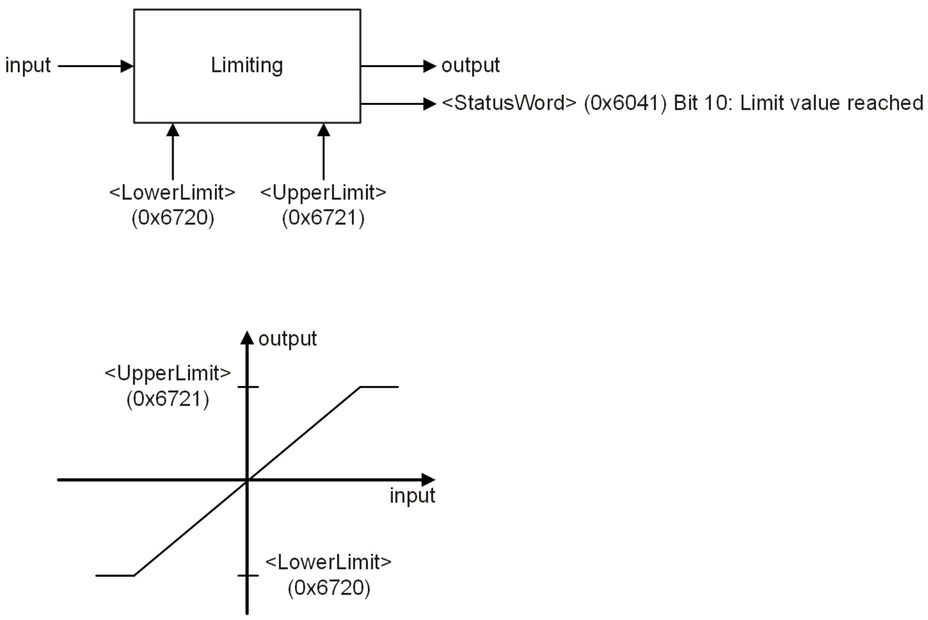


Figure 100: Limiting function

!	The ‹UpperLimit› (0x6721) must be greater than the ‹LowerLimit› (0x6720). If the ‹LowerLimit› (0x6720) is greater than the ‹UpperLimit› (0x6721), the ‹UpperLimit› (0x6721) will be set to the value of the ‹LowerLimit› (0x6720).

7.6.3.1 Object 0x6720: Setpoint conditioning upper setpoint limit

ValveFlowControl_SetpointConditioning_SetpointLimit							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6720	1	UpperLimit	INT16	rw	Y	‹LowerLimit› (0x6720)…32767	16384
0x6720	2	Unit	UINT8	ro	-	UINT8	0
0x6720	3	Prefix	INT8	ro	-	-4…0	0

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

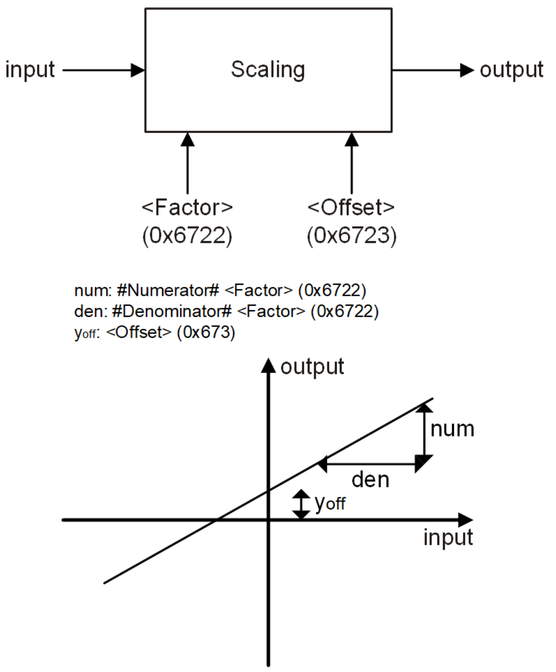
7.6.3.2 Object 0x6721: Setpoint conditioning lower setpoint limit

ValveFlowControl_SetpointConditioning_SetpointLimit							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6721	1	LowerLimit	INT16	rw	Y	-32767…‹UpperLimit› (0x6721)	-16384
0x6721	2	Unit	UINT8	ro	-	UINT8	0
0x6721	3	Prefix	INT8	ro	-	-4…0	0

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

7.6.4 Flow setpoint scaling

This function is used to scale the flow setpoint, e.g., to influence the input signal's value range. The output signal is calculated by multiplication of the input signal with a scaling factor and a subsequent addition of an offset according to the following figure.



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

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

Figure 101: Scaling function

7.6.4.1 Object 0x6722: Setpoint conditioning scaling 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).

ValveFlowControl_SetpointConditioning_Scaling							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6722	0	Factor	UINT32	rw	Y	UINT32	0x00010001

Value description

Table 91: Data structure of the slope factor

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

7.6.4.2 Object 0x6723: Setpoint conditioning scaling offset

This parameter is the offset of the linear output function.

ValveFlowControl_SetpointConditioning_Scaling							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6723	1	Offset	INT16	rw	Y	INT16	0
0x6723	2	Unit	UINT8	ro	-	UINT8	0
0x6723	3	Prefix	INT8	ro	-	-4…0	0

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

7.6.5 Ramp

The ramp function limits the slew rate of the input signal. The <Type> (0x6730) parameter is used to activate a one-, two- or four-quadrant ramp function or to deactivate it.

If the signal slope of the input signal is limited by configured slew rate (ramp running), the <StatusWord> (0x6041) bit 9 is set.

If the running ramp is stopped by the bit 15 (ramp stop) of the #ControlWord# signal, then the <StatusWord> (0x6041) bit 15 signalize that the output value of the ramp signal is hold (ramp frozen).

➪ Chapter "5.1.3.2 Object 0x6041: Status word", page 57
➪ Chapter "5.1.5 Object 0x604F: Device local", page 59
➪ Chapter "5.1.3.1 Object 0x6040: Device control word", page 56
➪ Chapter "5.1.6 Object 0x4040: Local control word", page 59

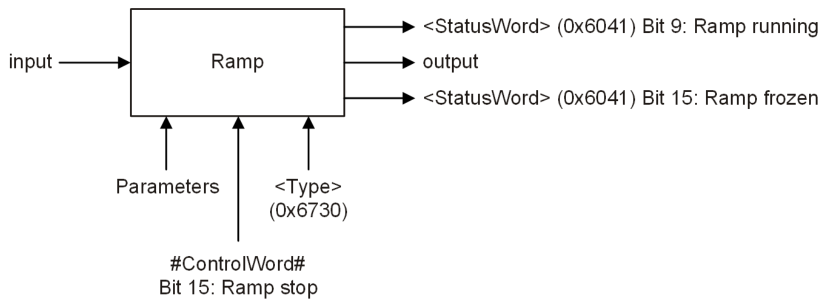


Figure 102: Ramp function

7.6.5.1 Object 0x6730: Setpoint conditioning ramp type

This parameter defines the progression of the ramp.

ValveFlowControl_SetpointConditioning_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6730	0	Type	INT8	rw	Y	0…3	0

Value description

Table 92: Possible values of parameter <Type> (0x6730)

<Type>	Description
0	No ramp
1	One-quadrant ramp
2	Two-quadrant ramp
3	Four-quadrant ramp

7.6.5.2 One-quadrant ramp (ramp type 1)

This function limits the input signal's rate of change to the defined <AccelerationTime> (0x6731).

This ramp type is active, if the parameter <Type> (0x6730) is set to 1.

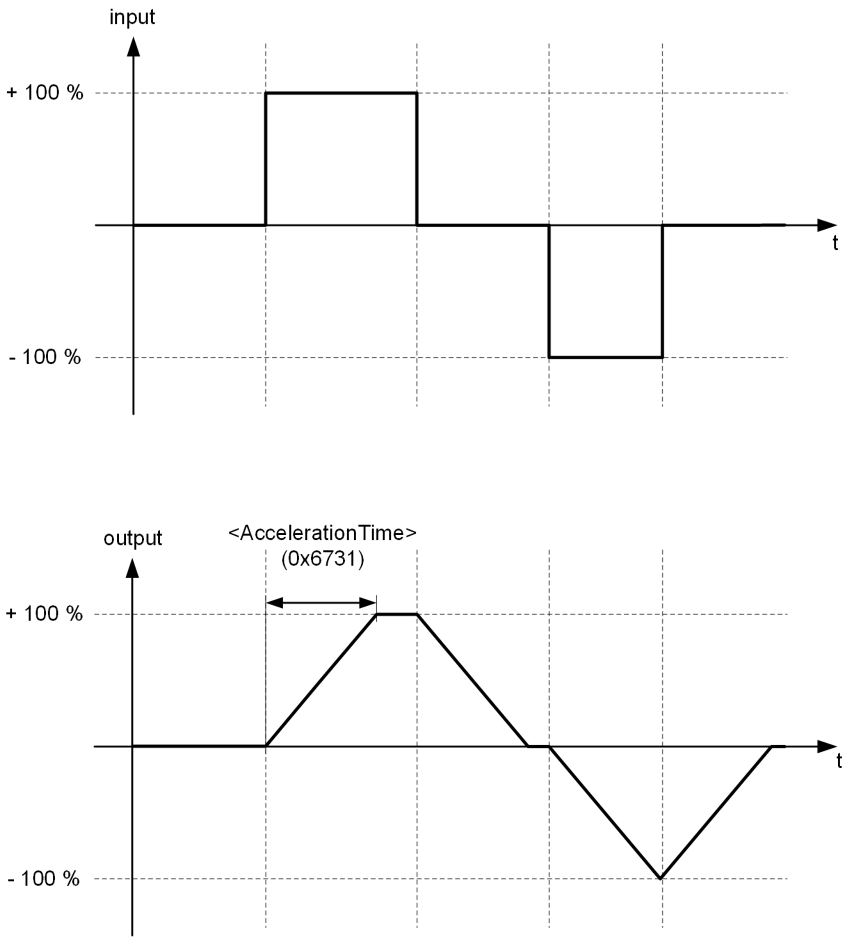


Figure 103: Ramp type 1

7.6.5.2.1 Object 0x6731: Setpoint conditioning ramp 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.

ValveFlowControl_SetpointConditioning_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6731	1	AccelerationTime	UINT16	rw	Y	UINT16	0
0x6731	2	Unit	UINT8	ro	-	UINT8	3
0x6731	3	Prefix	INT8	rw	Y	-4…0	-3

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

7.6.5.3 Two-quadrant ramp (ramp type 2)

This function limits the input signal's rate of change to the defined <AccelerationTime> (0x6731) and <DecelerationTime> (0x6734).

This ramp type is active, if the parameter <Type> (0x6730) is set to 2.

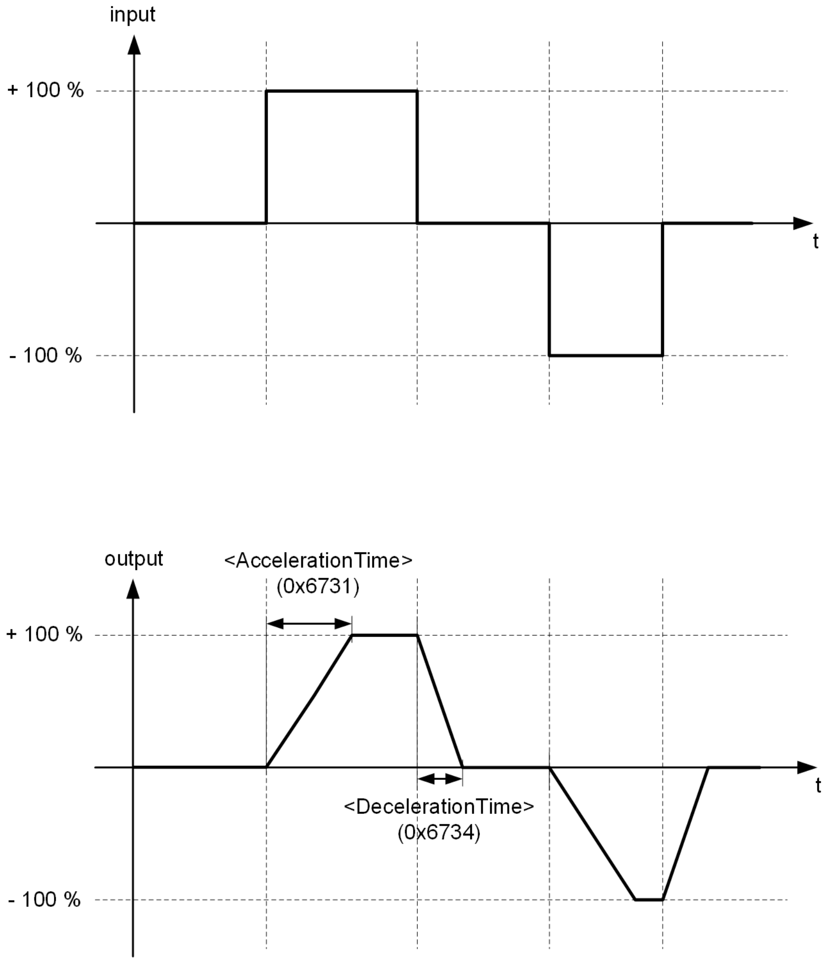


Figure 104: Ramp type 2

7.6.5.3.1 Object 0x6731: Setpoint conditioning ramp acceleration time

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

➪ Chapter "7.6.5.2.1 Object 0x6731: Setpoint conditioning ramp acceleration time", page 214

7.6.5.3.2 Object 0x6734: Setpoint conditioning ramp 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.

ValveFlowControl_SetpointConditioning_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6734	1	DecelerationTime	UINT16	rw	Y	UINT16	0
0x6734	2	Unit	UINT8	ro	-	UINT8	3
0x6734	3	Prefix	INT8	rw	Y	-4…0	-3

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

7.6.5.4 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> (0x6730) is set to 3.

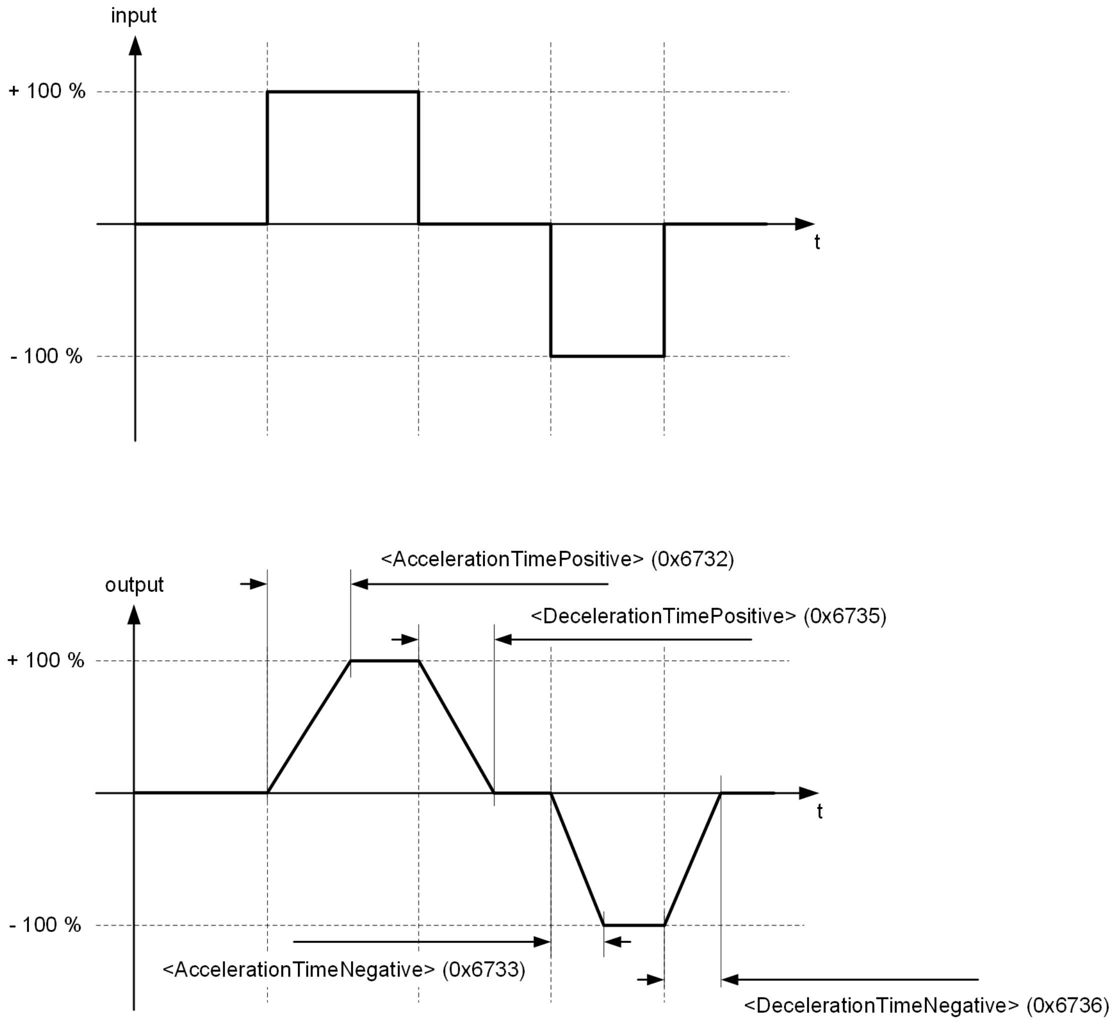


Figure 105: Ramp type 3

7.6.5.4.1 Object 0x6732: Setpoint conditioning ramp 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.

ValveFlowControl_SetpointConditioning_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6732	1	AccelerationTimePositive	UINT16	rw	Y	UINT16	0
0x6732	2	Unit	UINT8	ro	-	UINT8	3
0x6732	3	Prefix	INT8	rw	Y	-4...0	-3

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

7.6.5.4.2 Object 0x6733: Setpoint conditioning ramp 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.

ValveFlowControl_SetpointConditioning_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6733	1	AccelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x6733	2	Unit	UINT8	ro	-	UINT8	3
0x6733	3	Prefix	INT8	rw	Y	-4...0	-3

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

7.6.5.4.3 Object 0x6735: Setpoint conditioning ramp 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.

ValveFlowControl_SetpointConditioning_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6735	1	DecelerationTimePositive	UINT16	rw	Y	UINT16	0
0x6735	2	Unit	UINT8	ro	-	UINT8	3
0x6735	3	Prefix	INT8	rw	Y	-4...0	-3

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

7.6.5.4.4 Object 0x6736: Setpoint conditioning ramp 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.

ValveFlowControl_SetpointConditioning_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6736	1	DecelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x6736	2	Unit	UINT8	ro	-	UINT8	3
0x6736	3	Prefix	INT8	rw	Y	-4…0	-3

7.7 Flow controller

The flow control mode is used to control the valve flow independent from the pressure drop over the valve. To achieve this, changes of the supply and load pressures are compensated by the valve ("load compensation"). The demand value in this control mode is not a spool position, but a flow.

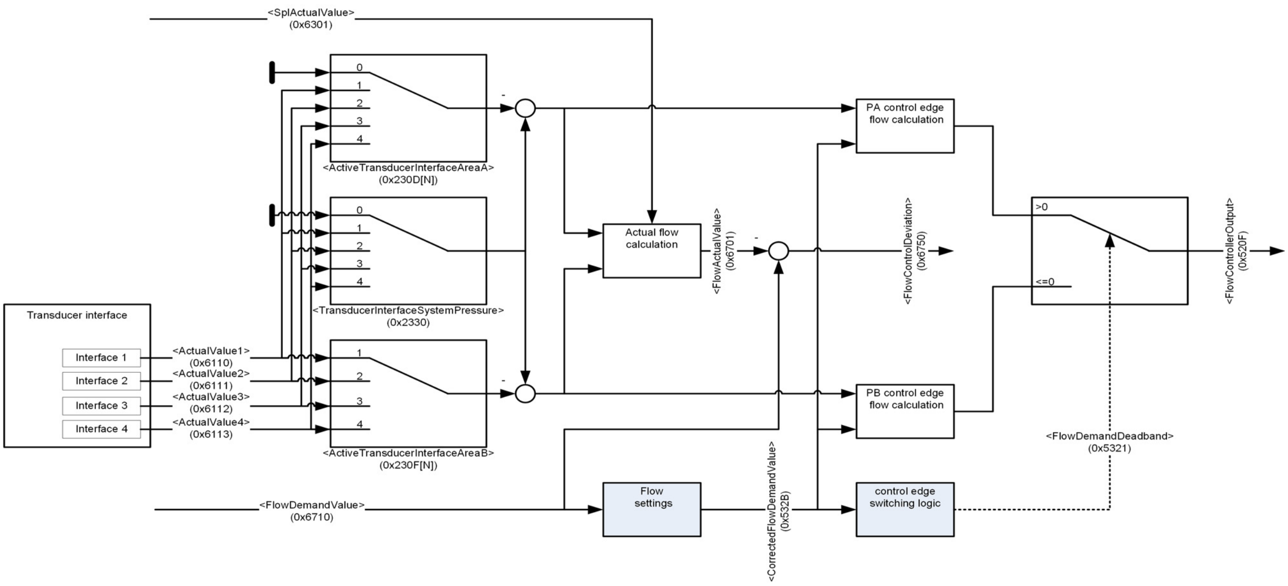


Figure 106: Flow controller

➪ Chapter "7.3.3.1 Object 0x6301: Spool position actual value", page 164
➪ Chapter "7.6.1 Object 0x6710: Setpoint conditioning demand value", page 209
➪ Chapter "7.5.10.1 Object 0x230D[N]: Active transducer interface area A", page 194
➪ Chapter "7.5.10.2 Object 0x230F[N]: Active transducer interface area B", page 195

7.7.1 Object 0x2330: Transducer interface system pressure

With this parameter, the source for the actual pressure on port P is selected. In this way another source or a constant value can be selected instead of the internal pressure sensor.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2330	0	TransducerInterfaceSystemPressure	UINT8	rw	Y	0…4	0

7.7.2 Object 0x532B: Corrected flow demand value

This parameter holds the output of the flow normalization calculation.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x532B	0	CorrectedFlowDemandValue	REAL32	ro	-	REAL32	None

7.7.3 Object 0x5321: Flow demand deadband

This parameter corresponds to the output of the switching calculation. The changeover between the control edges PA and PB is based on the flow direction. If this parameter is less than 0, the calculated flow of control edge A is taken as output of the flow controller, otherwise the calculated flow of control edge B.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5321	0	FlowDemandDeadband	REAL32	ro	-	REAL32	0.0

7.7.4 Object 0x520F: Flow controller output

This parameter holds the controller output of the flow controller.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x520F	0	FlowControllerOutput	INT16	ro	-	INT16	None

7.7.5 Object 0x6701: Flow actual value

This parameter holds the calculated actual flow.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6701	1	FlowActualValue	INT16	ro	-	INT16	None
0x6701	2	Prefix	INT8	ro	-	-4…0	0

7.7.6 Object 0x6750: Flow control deviation

This parameter holds the difference between the flow demand and the calculated actual flow.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x6750	1	FlowControlDeviation	INT16	ro	-	INT16	None
0x6750	2	Unit	UINT8	ro	-	UINT8	0
0x6750	3	Prefix	INT8	ro	-	-4…0	0

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

7.7.7 Flow settings

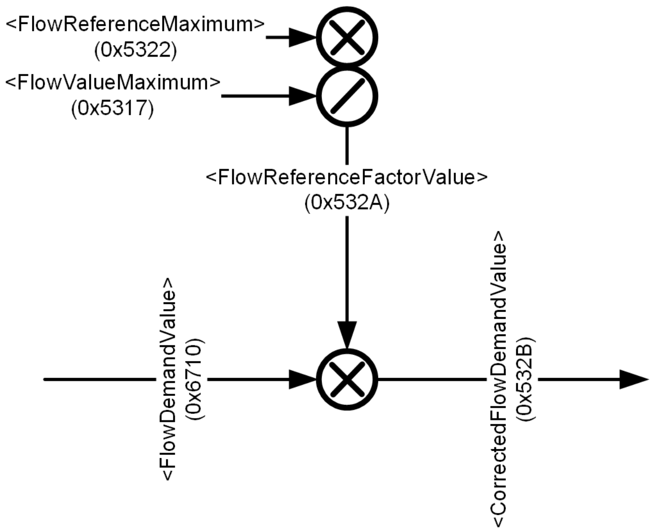


Figure 107: Flow settings

7.7.7.1 Object 0x5322: Flow reference maximum

With this parameter a further adjustment of the desired flow in litres per minute is possible, which is used for the calculation of the load compensation.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5322	0	FlowReferenceMaximum	REAL32	rw	Y	REAL32	180.0

7.7.7.2 Object 0x5317: Flow value maximum

This parameter determines the maximum permissible volume flow in litres per minute for the purchased servo valve.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5317	0	FlowValueMaximum	REAL32	ro	-	REAL32	180.0

7.7.7.3 Object 0x532A: Flow reference factor value

This parameter is the calculated factor from <FlowReferenceMaximum> (0x5322) divided by <FlowValueMaximum> (0x5317).

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x532A	0	FlowReferenceFactorValue	REAL32	ro	-	REAL32	None

7.7.8 Control edge switching logic

This switch between the control edge PA and PB can be manipulated by configuring the threshold <FlowDemandThresholdFactor> (0x531F) of the pressure compensation calculation logic. This prevents a controller output signal at a very small flow demand value. The control edge switching logic ensures that the correct pressure values are used for the load compensation when changing the flow direction. In addition, a threshold is used to avoid control edge switching for very small signals (signal noise). The default value is 1 %, which means that for flow demand values between −1 % and +1 % the flow demand is set to zero.

7.7.8.1 Object 0x531F: Flow demand threshold factor

This parameter specifies the switching hysteresis factor. This factor is multiplied by the increment for 100 % flow (16384). This results in the switching hysteresis in increments <FlowDemandThreshold> (05320).

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x531F	0	FlowDemandThresholdFactor	REAL32	ro	-	0.0…0.5	0.01

7.7.8.2 Object 0x5320: Flow demand threshold

This parameter holds the switching hysteresis in increments. If this parameter is less than 0, the load compensation calculation is performed via control edge PA, otherwise via PB.

ValveFlowControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x5320	0	FlowDemandThreshold	REAL32	ro	-	REAL32	0.0

7.8 Spool position (Q or flow) / pressure (p) switchover

The following structure is used to switch between spool position control (Q or flow) and pressure control. This block is effective in the control mode p/Q (5) and p/flow (14). 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: Device control mode", page 146

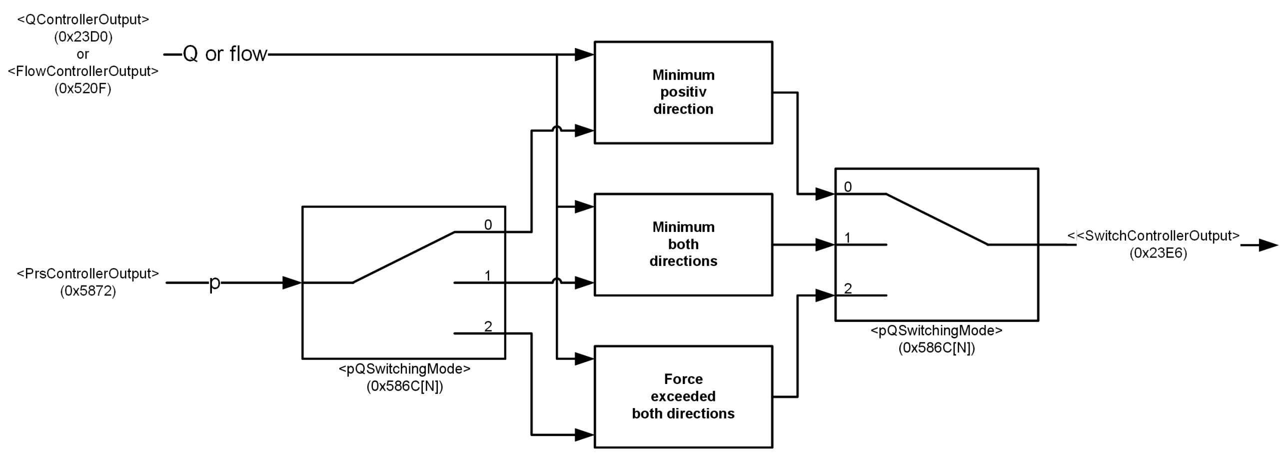


Figure 108: Spool position (Q or flow) / pressure (p) switchover

7.8.1 Object 0x586C[N]: p/Q or p/flow switching mode

This parameter defines the switching mode.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x586C	N=1…16	pQSwitchingMode	UINT8	rw	Y	0…2	0

Value description

Table 93: Possible values of parameter <pQSwitchingMode> (0x586C)

<pQSwitchingMode>	Description
0	Minimum criterion in positive direction.
1	Minimum criterion in both directions.
2	Force exceeded in both directions.

7.8.1.1 Object 0x23E6: Switch controller output

This parameter contains the output signal of the p/Q or p/flow switchover function.

ValvePressureControl							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x23E6	N=1…16	SwitchControllerOutput	INT16	ro	-	INT16	None

7.8.2 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 (Q or flow) and pressure (p) control.

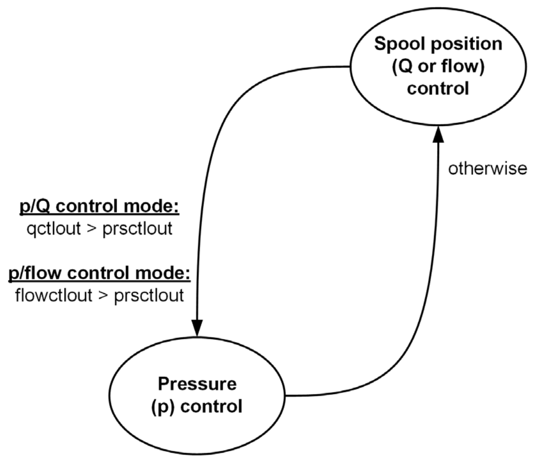


Figure 109: State machine used to switch between spool position (Q or flow) control and pressure (p) control

Value	Parameter
qctlout equals qdem	<QDemandValue> (0x6310)
flwctlout	<FlowControllerOutput> (0x520F)
prsctlout	<PrsControllerOutput> (0x5872)

➪ Chapter "7.2.1 Object 0x6310: Demand value", page 154
➪ Chapter "7.7.4 Object 0x520F: Flow controller output", page 219
➪ Chapter "7.5.7 Object 0x5872: Pressure controller output", page 191

7.8.3 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 (Q or flow) control and pressure (p) control.

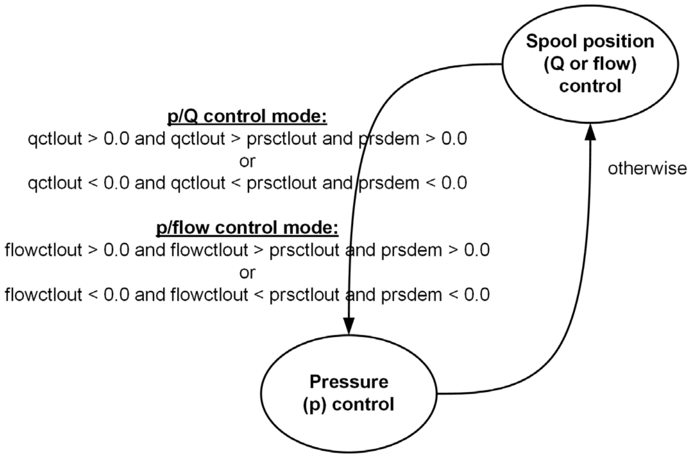


Figure 110: State machine used to switch between spool position (Q or flow) control and pressure (p) control

Value	Parameter
qctlout	<QDemandValue> (0x6310)
flwctlout	<FlowControllerOutput> (0x520F)
prsdem	<PrsDemandValue> (0x6390)
prsctlout	<PrsControllerOutput> (0x5872)

➪ Chapter "7.2.1 Object 0x6310: Demand value", page 154
➪ Chapter "7.7.4 Object 0x520F: Flow controller output", page 219
➪ Chapter "7.4.1 Object 0x6390: Setpoint conditioning demand value", page 179
➪ Chapter "7.5.7 Object 0x5872: Pressure controller output", page 191

7.8.4 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 (Q or flow) control and pressure (p) control.

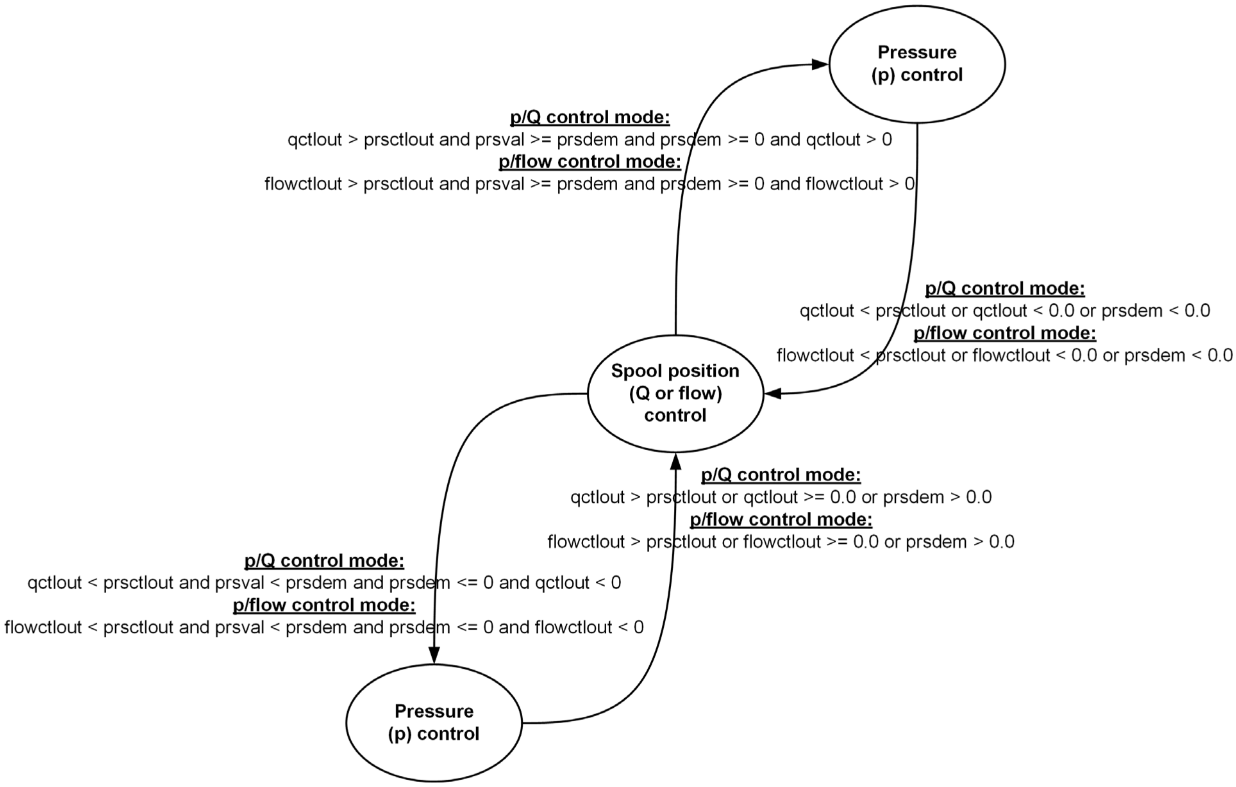


Figure 111: State machine used to switch between spool position (Q or flow) control and pressure (p) control

Value	Parameter
qctlout	‹QDemandValue› (0x6310)
prsdem	‹PrsDemandValue› (0x6390)
prsval	‹PrsActualValue› (0x6381)
prsctlout	‹PrsControllerOutput› (0x5872)

➪ Chapter "7.2.1 Object 0x6310: Demand value", page 154
➪ Chapter "7.4.1 Object 0x6390: Setpoint conditioning demand value", page 179
➪ Chapter "7.5.1 Object 0x6381: Pressure actual value", page 189
➪ Chapter "7.5.7 Object 0x5872: Pressure controller output", page 191

7.9 Monitoring

This chapter describes the monitoring functionalities listed below:

7.9.1 Power supply monitoring

The 24 V supply voltage is permanently monitored for over- and undervoltage. A fault is generated if the parameter is outside the range between 16.5 V and 38 V. The normal operating supply voltage range is between 18 V and 36 V.

7.9.1.1 Object 0x2804: Power supply voltage

This parameter holds the actual value of the power supply voltage (in mV).

PowerSupply							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2804	0	PowerSupplyVoltage	UINT16	ro	-	UINT16	None

The power supply is monitored. The following fault reactions can be triggered.

If the power supply voltage is below 16.5 V for more than 5 ms, the undervoltage fault is triggered.

If the power supply voltage is above 38 V for more than 200 ms, the overvoltage fault is triggered.

Fault description

Table 94: Fault codes

<PowerSupplyVoltage>	**Indicated fault code**		**Fault description**
	Dec	**Hex**	
PowerSupplyVoltage < 16.5 V	5	0x05	Power supply voltage too low.
PowerSupplyVoltage > 38 V	6	0x06	Power supply voltage too high.

The power supply voltage should be in the range of 18…36 V to ensure proper operation.

7.9.2 Analog input cable break monitoring

Cable break monitoring is available for the analog input signals 0…4 and for the sensor power supply wires. The integrated pressure sensor module for the internal pressure transducer P, B, A only has a monitoring of the module power supply.

If the 0…10 mA or the ±10 mA or 0…10 V or −10 V…+10 V 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 and the sensor module is active. If the 4…20 mA analog input type is used, sensor cable break monitoring for the analog inputs 0…4 is available.

For the analog inputs 2…4 a sensor power supply cable break monitoring exists. A sensor load current < 0.1 mA is interpreted as cable break (typical threshold at 55 µA). Each sensor has its own fault detection (for no sensor load). A short circuit on one of the sensors leads to a common fault response for all sensors as there is a common self resetting fuse. The maximum current per sensor is 200 mA. A power supply short circuit is also detected.

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.

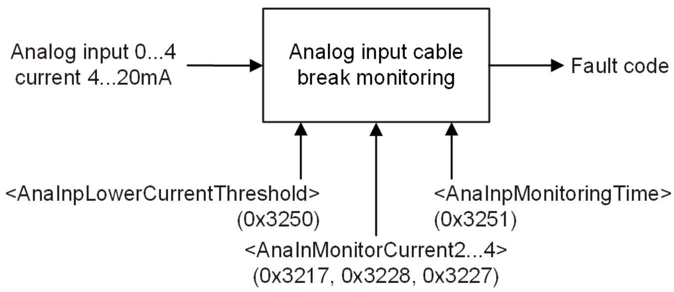
Table 95: Cable break monitoring features

	Power supply cable break monitoring	**Sensor cable break monitoring**		
	< 0.1 mA	**0…10 mA – 10…10 mA**	**4…20 mA, threshold adjustable with <AnaInputLowerCurrentThreshold> (0x3250)**	**0…10 V –10…10 V**
Internal pressure transducer (at port A)	No	No	No	No
Integrated pressure sensor module (at ports A, B, P)	Yes Fault code 120 (0x78)	No	No	No
Analog input 0	No	No	Yes Fault code 31 (0x1F)	No
Analog input 1	No	No	Yes Fault code 32 (0x20)	No
Analog input 2	Yes Fault code 28 (0x1C)	No	Yes Fault code 33 (0x21)	No
Analog input 3	Yes Fault code 29 (0x1D)	No	Yes Fault code 34 (0x22)	No
Analog input 4	Yes Fault code 30 (0x1E)	No	Yes Fault code 35 (0x23)	No

The following fault codes are generated in the case of a cable break. The fault reaction itself can be configured using the fault reaction logic.
➪ Chapter "8.1.3 Fault reaction", page 275

Table 96: Possible fault codes

Input	Indicated fault code		Fault description
	Dec.	**Hex.**	
Internal pressure transducer (at port A) or integrated pressure sensor module (at ports A, B, P)	120	0x78	Pressure sensor monitoring failed
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



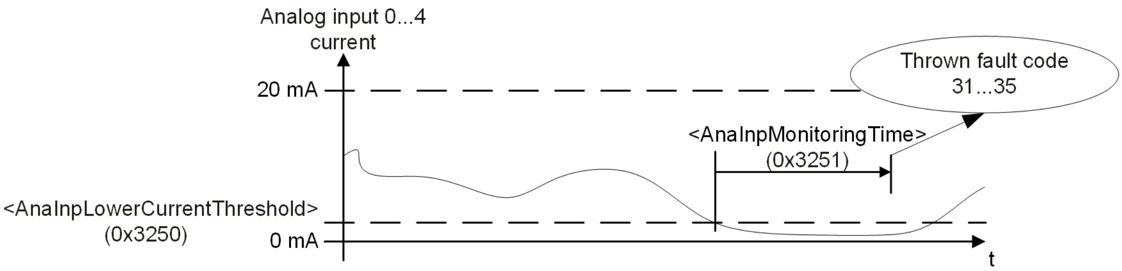


Figure 112: 4…20 mA analog input signal monitoring

If the current is below <AnaInpLowerCurrentThreshold> (0x3250) for <AnaInpMonitoringTime> (0x3251), cable break is detected.

7.9.2.1 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	AnaInpLowerCurrentThreshold	REAL32	rw	Y	2.2…20.0	3.0

7.9.2.2 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	AnalnpMonitoringTime	UINT16	rw	Y	0…60000	10

7.9.3 Electronic temperature monitoring

The temperature on the valve electronics is constantly measured and stored as a temperature histogram. If the temperature falls below or exceeds the maximum permissible temperature, a fault code is generated.

7.9.3.1 Object 0x2805: Electronic temperature

This parameter contains the temperature (in °C) of the servo valve electronics.

Temperature							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2805	0	PcbTemperature	INT16	ro	-	INT16	None

Fault description

Table 97: Fault codes

<PcbTemperature >	**Indicated fault code**		**Fault description**
	Dec	**Hex**	
PCB temperature < −40 °C	13	0x0D	Electronics temperature too low.
PCB temperature > 85 °C	14	0x0E	Electronics temperature too high.
PCB temperature > 105 °C	15	0x0F	Electronics temperature exceeded.

The PCB temperature should not exceed the range of −40…85 °C to ensure proper operation. The electronics temperature has a big impact on the electronics lifetime. The service life is increased if higher temperatures are avoided.

7.9.3.2 Object 0x2855: Electronic temperature histogram over operating time

This parameter counts the operating time that the device has been operated in different temperature ranges. The operating times for the individual temperature ranges are recorded in separate parameters. The operating time is measured in minutes.

Temperature							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2855	0	RangeMin	UINT32	ro	-	UINT32	0
0x2855	1	RangeLow	UINT32	ro	-	UINT32	0
0x2855	2	RangeMid	UINT32	ro	-	UINT32	0
0x2855	3	RangeHigh	UINT32	ro	-	UINT32	0
0x2855	4	RangeMax	UINT32	ro	-	UINT32	0

Value description

Table 98: Temperature ranges

<Parameter>	**Temperature range [°C]**
<RangeMin>	< 0
<RangeLow>	≥0 to <65
<RangeMid>	≥65 to <80
<RangeHigh>	≥80 to <95
<RangeMax>	≥95

7.9.4 Operating time counters

The duration of three operating states is recorded.

- Time since last power on
- Cumulative power on time since production of valve
- Cumulative active time since production of valve

7.9.4.1 Object 0x2827: Time since last power on

This parameter contains the power on time of the device for the current power cycle.

The time is measured in minutes.

Hardware_DiagnosticData							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2827	0	TimeSinceLastPowerOn	UINT32	ro	-	UINT32	0

7.9.4.2 Object 0x280D: Cumulative power on time since production

The parameter <CumulativePowerOnTime> (0x280D), sub-index 1 contains the cumulative power on time of the device since its production.

The parameter <CumulativeValveActiveTime> (0x280D), sub-index 2 contains the cumulative active time of the device since its production. Active Time means the device states are 'HOLD', 'FAULT HOLD' or 'ACTIVE'.

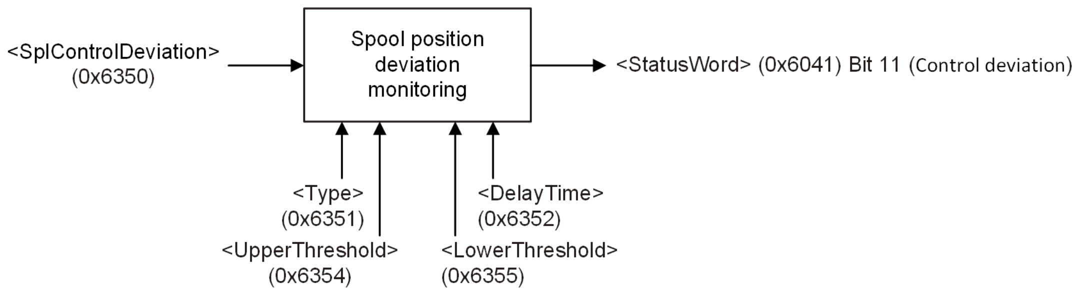
The times are measured in minutes.

Hardware_DiagnosticData							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x280D	1	CumulativePowerOnTime	UINT32	ro	-	UINT32	0
0x280D	2	CumulativeValveActiveTime	UINT32	ro	-	UINT32	0

7.9.5 Spool position control deviation monitoring

The spool position deviation monitoring can be activated for all controllers (<ControlMode> (0x6043) = 1, 2, 3, 4, 5, 13, 14). If the spool position exceeds the defined thresholds, this will be indicated with the bit 11 of the status Word <StatusWord> (0x6041). This monitoring can be activated using the object <Type> (0x6351).

➪ Chapter "7.9.5.1 Object 0x6351: Control monitoring type", page 232
➪ Chapter "7.1.1 Object 0x6043: Device control mode", page 146
➪ Chapter "5.1.3.2 Object 0x6041: Status word", page 57



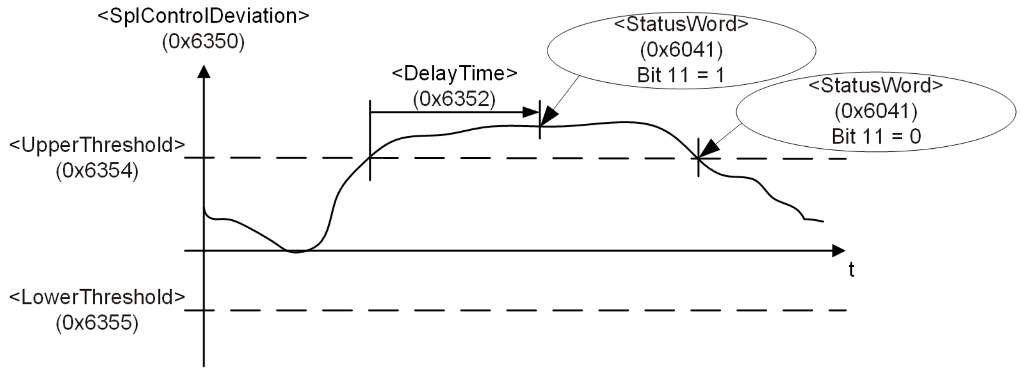


Figure 113: Spool position control deviation monitoring

➪ Chapter "7.3.6.1 Object 0x6350: Control deviation", page 176

7.9.5.1 Object 0x6351: Control monitoring type

The parameter <Type> (0x6351) is used to activate or deactivate the 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

Table 99: Possible values of parameter <Type> (0x6351)

<Type>	**Description**
0	Spool position control deviation monitoring off.
1	Spool position control deviation monitoring on.

7.9.5.2 Object 0x6352: Control monitoring 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	-	-4…0	-3

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

7.9.5.3 Object 0x6354: Control monitoring 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	-	-4…0	0

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

7.9.5.4 Object 0x6355: Control monitoring 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	-	-4…0	0

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

7.9.6 Pilot spool position control deviation monitoring

The pilot spool position deviation monitoring can be activated for all controllers (<ControlMode> (0x6043) = 1, 2, 3, 4, 5, 13, 14).

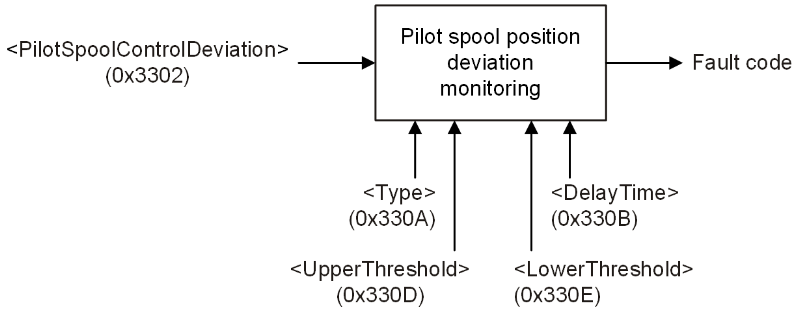
If the spool position exceeds the defined thresholds, this will generate the fault code 68 (0x44). The fault reaction itself can be configured using the fault reaction logic.

➪ Chapter "8.1.3 Fault reaction", page 275

Monitoring of the pilot stage is only useful for dual stage valves or for RKP-D pilot valves.

This monitoring can be activated using the object <Type> (0x330A).

➪ Chapter "7.9.5.1 Object 0x6351: Control monitoring type", page 232
➪ Chapter "7.1.1 Object 0x6043: Device control mode", page 146



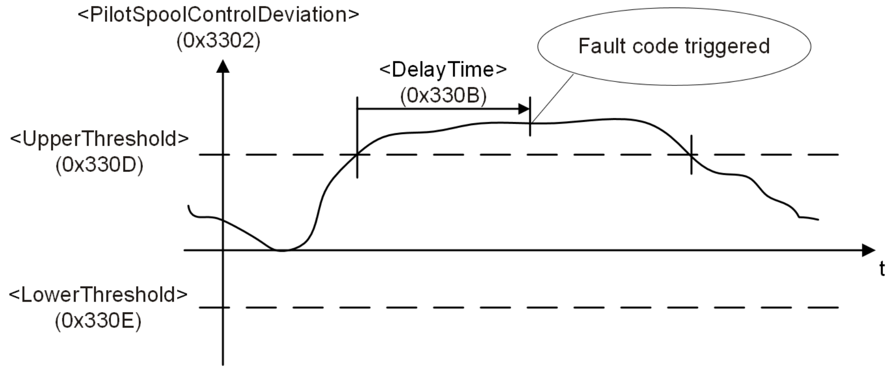


Figure 114: Pilot spool position control deviation monitoring

7.9.6.1 Object 0x3302: Pilot control deviation

➪ Chapter "7.3.6.2 Object 0x3302: Pilot control deviation", page 176

7.9.6.2 Object 0x330A: Pilot spool control monitoring type

The parameter <Type> (0x330A) is used to activate or deactivate the pilot spool position control deviation monitoring function.

ValvePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x330A	0	Type	INT8	rw	Y	0…1	0

Value description

Table 100: Possible values of parameter <Type> (0x330A)

<Type>	**Description**
0	Pilot spool position control deviation monitoring off.
1	Pilot spool position control deviation monitoring on.

7.9.6.3 Object 0x330B: Pilot spool control monitoring 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> (0x330A), <UpperThreshold> (0x330D), <LowerThreshold> (0x330E) or <DelayTime> (0x330B).

ValvePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x330B	1	DelayTime	UINT16	rw	Y	UINT16	30
0x330B	2	Unit	UINT8	ro	-	UINT8	3
0x330B	3	Prefix	INT8	ro	-	-4…0	-3

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

7.9.6.4 Object 0x330D: Pilot spool control monitoring upper threshold

This parameter defines the upper threshold of the tolerance band of the control deviation.

ValvePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x330D	1	UpperThreshold	INT16	rw	Y	INT16	512
0x330D	2	Unit	UINT8	ro	-	UINT8	3
0x330D	3	Prefix	INT8	ro	-	-4…0	-3

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

7.9.6.5 Object 0x330E: Pilot spool control monitoring 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
0x330E	1	LowerThreshold	INT16	rw	Y	INT16	-512
0x330E	2	Unit	UINT8	ro	-	UINT8	3
0x330E	3	Prefix	INT8	ro	-	-4…0	-3

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

7.9.7 Flow control deviation monitoring

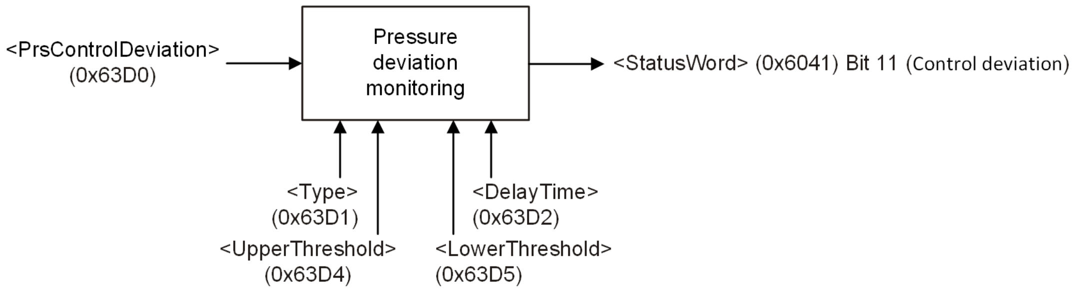
As the measured flow actual value isn't available yet, it is recommended to monitor the controller deviation of the spool position controller <SplControlDeviation> (0x6350).

➪ Chapter "7.3.6.1 Object 0x6350: Control deviation", page 176

7.9.8 Pressure control deviation monitoring

The pressure deviation monitoring can be activated with the controllers p, p/Q and p/Flow (<ControlMode> (0x6043) = 3, 4, 5, 13, 14). For the p/Q and p/Flow controllers, pressure deviation monitoring is only active if the pressure controller effective (indicated by bit 8 in the status Word <StatusWord> (0x6041). If the pressure exceeds the defined thresholds, bit 11 is set in the status Word <StatusWord> (0x6041). This monitoring can be activated using the object <Type> (0x63D1).

➪ Chapter "7.9.8.1 Object 0x63D1: Control monitoring type", page 237
➪ Chapter "7.1.1 Object 0x6043: Device control mode", page 146
➪ Chapter "5.1.3.2 Object 0x6041: Status word", page 57



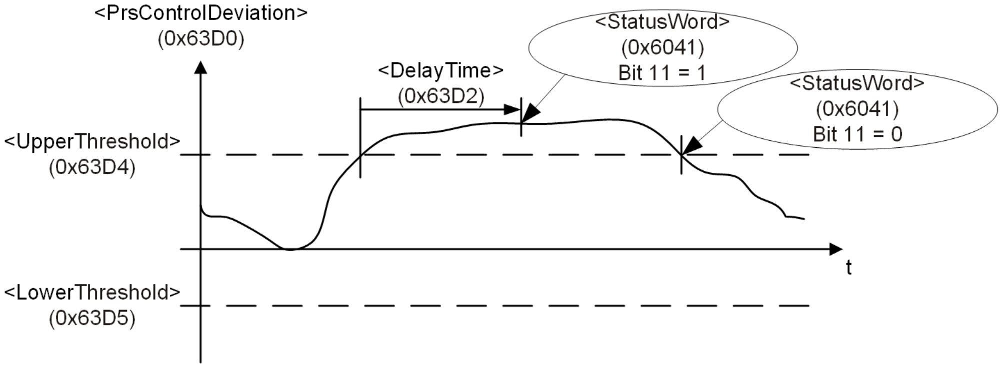


Figure 115: Pressure control deviation monitoring

➪ Chapter "7.5.2 Object 0x63D0: Pressure control deviation", page 190

7.9.8.1 Object 0x63D1: Control monitoring 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

Table 101: Possible values of parameter <Type> (0x63D1)

<Type>	Description
0	Pressure control deviation monitoring off.
1	Pressure control deviation monitoring on.

7.9.8.2 Object 0x63D2: Control monitoring 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	-	-4…0	-3

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

7.9.8.3 Object 0x63D4: Control monitoring 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	-	-4…0	0

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

7.9.8.4 Object 0x63D5: Control monitoring 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	-	-4...0	0

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

7.9.9 Main stage spool position sensor cable break monitoring

Monitoring is only active in case of device state greater 'INIT'. In case of a cable break, the fault code 20 (0x14) (Main stage spool position sensor cable break) is generated and the device state machine state changes to 'FAUL INIT'.

!	The servo valve must be serviced by Moog service technicians. The servo valve may behave unpredictably.

7.9.10 Spool / pilot spool position sensor cable break monitoring

Monitoring is only active in case of device state greater 'INIT'. In case of a cable break, the fault code 17 (0x11) (Pilot/single stage spool position sensor cable break) is indicated and the device state machine state changes to 'FAULT INIT'.

!	The servo valve must be serviced by Moog service technicians. The servo valve may behave unpredictably.

7.9.11 Current control deviation monitoring

The internal control deviation monitoring of the actuator current leads to fault code 66 (0x42) (Power stage current control monitoring) in case of a fault. This monitoring is always enabled.

7.9.12 Actuator over current monitoring

This function monitors the power stage of the valve actuator. The power stage is switched off by hardware if there is a permanent over current. Actuator over current leads to the fault code 16 (0x10) (Overcurrent of Power Stage) in case of a fault.

7.9.13 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 spool position actual value <SpoolActualValue> (0x6301) leaves the failsafe position window, the digital output switches to 0 V.

The failsafe spool position monitoring function is enabled by selecting a digital output as failsafe output by setting one of the parameters <DigitalOutputConfiguration0…2> (0x5E41) to 1 (failsafe spool position monitoring on). The servo valve monitors the failsafe position of the spool position <SpoolActualValue> (0x6301).

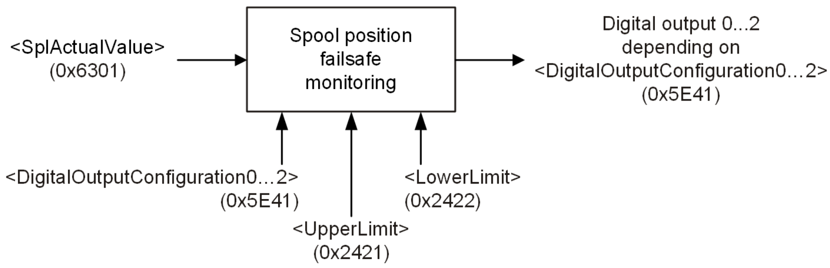
➪ Chapter "6.6.2 Object 0x5E41: Digital output configuration", page 104

⚠ 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!



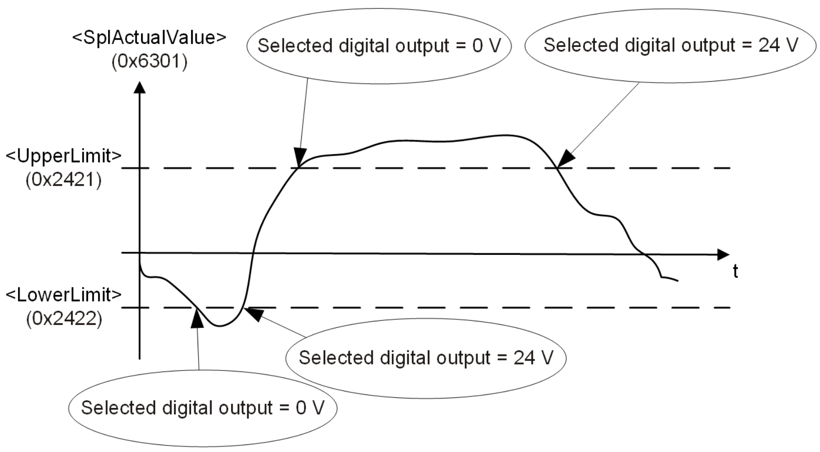


Figure 116: Failsafe monitoring

Electrical signal

Spool position <SpoolActualValue> (0x6301)	Electrical signal (negative logic)
Actual spool position value is within the failsafe monitoring window	Selected digital output 1 = 24 V
Actual spool position value is outside the failsafe monitoring window	Selected digital output 1 = 0 V

Failsafe spool position monitoring behavior depending on the device state machine 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 <DigitalOutputConfiguration0…2> (0x5E41) is set to 1 (failsafe spool position monitoring on).

7.9.13.1 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

7.9.13.2 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

7.9.13.3 Object 0x3307: Spring Position Minimum

This parameter is optionally set on request. It 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).

ValveFailSafeWindowMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3307	0	SpringPositionMinimum	INT16	ro	Y	INT16	None

7.9.13.4 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).

ValveFailSafeWindowMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3308	0	SpringPositionMaximum	INT16	ro	Y	INT16	None

7.9.14 Object 0x2862: Spool distance cumulative

This parameter counts the cumulative distance the spool moves during the whole operating time. It can be used as a measure of spool wear. The different sub-objects represent the same parameter in different units and data types. The sub-object 5 and 6 represent the raw value.

ConditionMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2862	1	Spool Distance Cumulative In Fullscale	REAL32	ro	Y	REAL32	None
0x2862	4	Spool Distance Cumulative In Millimeter	REAL32	ro	Y	REAL32	None
0x2862	5	Spool Distance Cumulative Least Significant Word	REAL32	ro	Y	REAL32	None
0x2862	6	Spool Distance Cumulative Most Significant Word	REAL32	ro	Y	REAL32	None

7.10 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 fieldbus. The variables calculated by the event handler can be accessed via the fieldbus. There are in total eight event handlers, which will be configured by eight associated strings processed by the event 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 at least every millisecond.

7.10.1 Event expressions

The event expressions are strings with maximal 192 characters and are built in the following manner:

- The parameters to be used can only be accessed through their short names. The short names of a parameter can be found in the chapter "Object dictionary".
  ⇨ Chapter "10 Object dictionary", page 290
- 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 operand

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

<table><tr><th>Symbol</th><th>Description</th></tr><tr><td>|</td><td>Bitwise OR</td></tr></table>

- Logical operator AND

Symbol	Description
&&	Logical AND

- Logical operator OR

<table><tr><th>Symbol</th><th>Description</th></tr><tr><td>||</td><td>Logical OR</td></tr></table>

- 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

7.10.1.1 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	EventExpression1	STRING(192)	rw	Y	STRING(192)	""

7.10.1.2 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	EventExpression2	STRING(192)	rw	Y	STRING(192)	""

7.10.1.3 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	EventExpression3	STRING(192)	rw	Y	STRING(192)	""

7.10.1.4 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	EventExpression4	STRING(192)	rw	Y	STRING(192)	""

7.10.1.5 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	EventExpression5	STRING(192)	rw	Y	STRING(192)	""

7.10.1.6 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	EventExpression6	STRING(192)	rw	Y	STRING(192)	""

7.10.1.7 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	EventExpression7	STRING(192)	rw	Y	STRING(192)	""

7.10.1.8 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	EventExpression8	STRING(192)	rw	Y	STRING(192)	""

7.10.1.9 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	EventEnable1	UINT8	rw	Y	0…1	0
0x2909	2	EventEnable2	UINT8	rw	Y	0…1	0
0x2909	3	EventEnable3	UINT8	rw	Y	0…1	0
0x2909	4	EventEnable4	UINT8	rw	Y	0…1	0
0x2909	5	EventEnable5	UINT8	rw	Y	0…1	0
0x2909	6	EventEnable6	UINT8	rw	Y	0…1	0
0x2909	7	EventEnable7	UINT8	rw	Y	0…1	0
0x2909	8	EventEnable8	UINT8	rw	Y	0…1	0

7.10.2 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[…], varf[…], dumu[…], dums[…], and dumf[…] are reserved for these tasks.

⇨ Chapter "6.8 Free to use parameters", page 140

Example 1:

varu32[1]=varu32[1]+1;qset=varu32[1]*5

- Calculations are processed every main task cycle (no condition).
- Increase varu32[1] by one.
- Calculates the Q setpoint as five times the varu32[1] variable.

Example 2:

splval>10000?qset=0:(splval<1000?qset=11000)

- Two separate events in one expression using a semicolon as separator.
- Q setpoint is only changed if the Q 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):
qset<1000?ctlmod=2;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?qset=0:(splval<1000?qset=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.
  - Q setpoint is only changed if the Q 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.

7.11 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.

7.11.1 Data logger state machine

The data logger is controlled by a state machine. The states and the transitions are explained in this chapter.

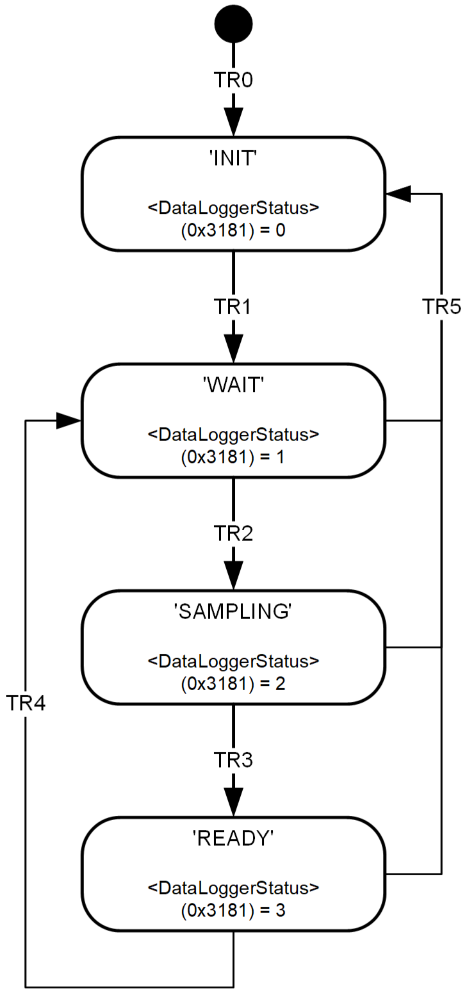


Figure 117: Data logger state machine

Table 102: States of the data logger state machine

State	<DataLoggerTriggerType> (0x3188)		
	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 103: Transitions 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) <TriggerMapppingParameter> (0x3189)

7.11.1.1 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	INT8	rw	Y	0…1	0

Value description

Table 104: Possible values of parameter <Control> (0x3180)

<Control>	Description
0	Disable data logger.
1	Enable data logger.

7.11.1.2 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

Table 105: Possible values of parameter <Status> (0x3181)

<Status>	Description
0	'INIT'
1	'WAIT'
2	'SAMPLING'
3	'READY'

7.11.2 Channel settings

Four channels can be used within the data logger. All readable parameters are available as input for the channels.

7.11.2.1 Object 0x3185: Channel mapping parameter

The parameters <ChannelMappingParameter1…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	ChannelMappingParameter1	UINT32	rw	Y	UINT32	0x63100110
0x3185	2	ChannelMappingParameter2	UINT32	rw	Y	UINT32	0x63010110
0x3185	3	ChannelMappingParameter3	UINT32	rw	Y	UINT32	0x63900110
0x3185	4	ChannelMappingParameter4	UINT32	rw	Y	UINT32	0x63810110

Value description

<ChannelMappingParameter1…4>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length 0x10 or 0x20
Example	0x63	0x10	0x01	0x10

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

The example value is 0x63100110, which refers to the <SplDemandValue> (0x6310), sub-index 0x01 with a length of 16 bit (16=0x10).

7.11.2.2 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	EnableChannel1	UINT8	rw	Y	0…1	0
0x3184	2	EnableChannel2	UINT8	rw	Y	0…1	0
0x3184	3	EnableChannel3	UINT8	rw	Y	0…1	0
0x3184	4	EnableChannel4	UINT8	rw	Y	0…1	0

Value description

Table 106: Possible values of parameter <EnableChannel> (0x3184)

<EnableChannel1…4>	**Description**
0	Channel disabled.
1	Channel enabled.

7.11.3 Sample frequency

The maximum sample frequency is limited to 20000 samples per seconds (20k Hz). 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).

7.11.3.1 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

Table 107: Possible values of parameter <Divider> (0x3182)

<Divider>	Description
1	20000 samples per second (20 kHz).
2…65534	20000/2…20000/65534 samples per second.
65535	20000/65535 = 0.3052 samples per second.

7.11.4 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).

7.11.4.1 Object 0x3189: Trigger parameter

The <TriggerMappingParameter> (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	TriggerMappingParameter	UINT32	rw	Y	UINT32	0x63100110

Value description

<TriggerMappingParameter>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length 0x10 or 0x20
Default	0x63	0x10	0x01	0x10

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

The default value is 0x63100110, which refers to the <SplDemandValue> (0x6310), sub-index 0x01 with a length of 16 bit (16=0x10).

7.11.4.2 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

Table 108: Possible values of parameter <TriggerType> (0x3188)

<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).

7.11.4.3 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

7.11.4.4 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

Table 109: Possible values of parameter <TriggerCoupling> (0x318A)

<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).

7.11.4.5 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

Table 110: Possible values of parameter <TriggerSlope> (0x318B)

<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).

7.11.4.6 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

Table 111: Possible values of parameter <TriggerPosition> (0x318D)

<TriggerPosition>	**Description**
> 0	Post trigger.
= 0	No delay.
< 0	Pre trigger.

7.11.5 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.

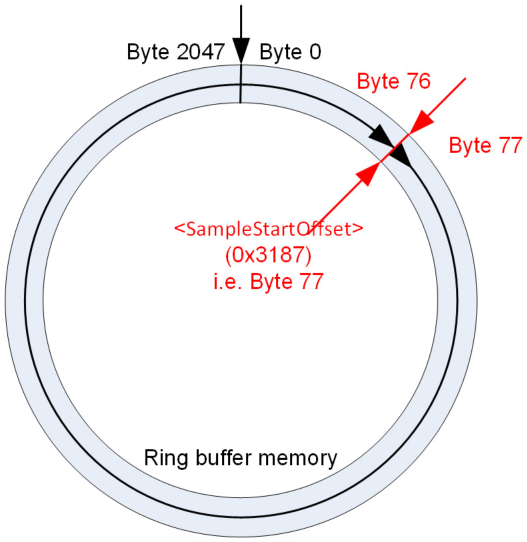


Figure 118: Data memory organization

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.

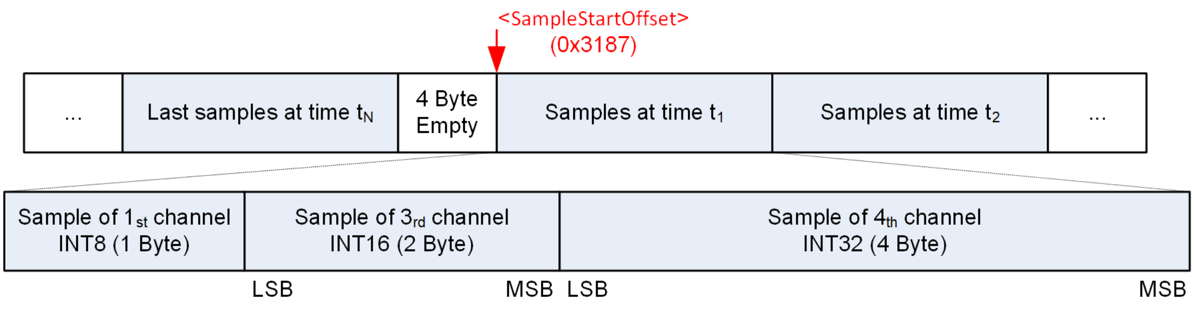


Figure 119: Data memory - mixed channel data

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.

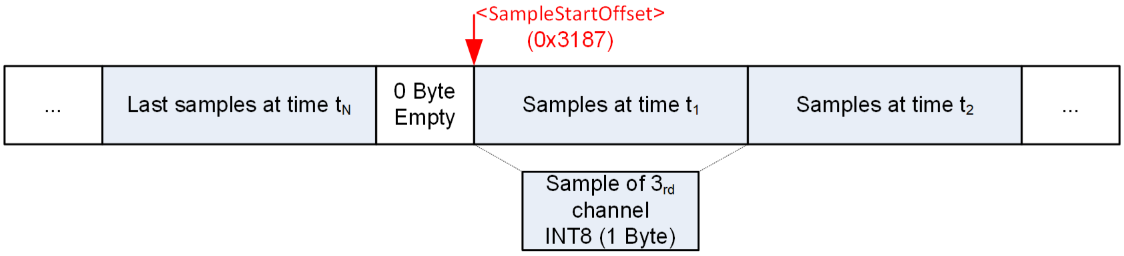


Figure 120: Data memory - one channel with INT8 parameter

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.

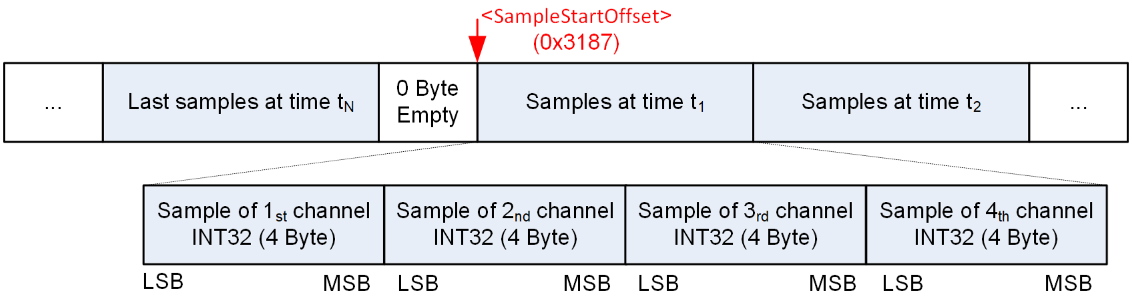


Figure 121: Data memory - four channels with INT32 parameters

7.11.5.1 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	-	DOMAIN	...

7.11.5.2 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

7.11.5.3 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

7.12 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 for the servo valve to optimize the servo valves behavior. Different shapes, amplitude and offset can be configured.
  ➪ Chapter "7.12.3.1 Object 0x3101: Output", page 259
- 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 "7.12.3.3 Object 0x3102: Square", page 260

7.12.1 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.

7.12.1.1 Rectangular output signal (type 1)

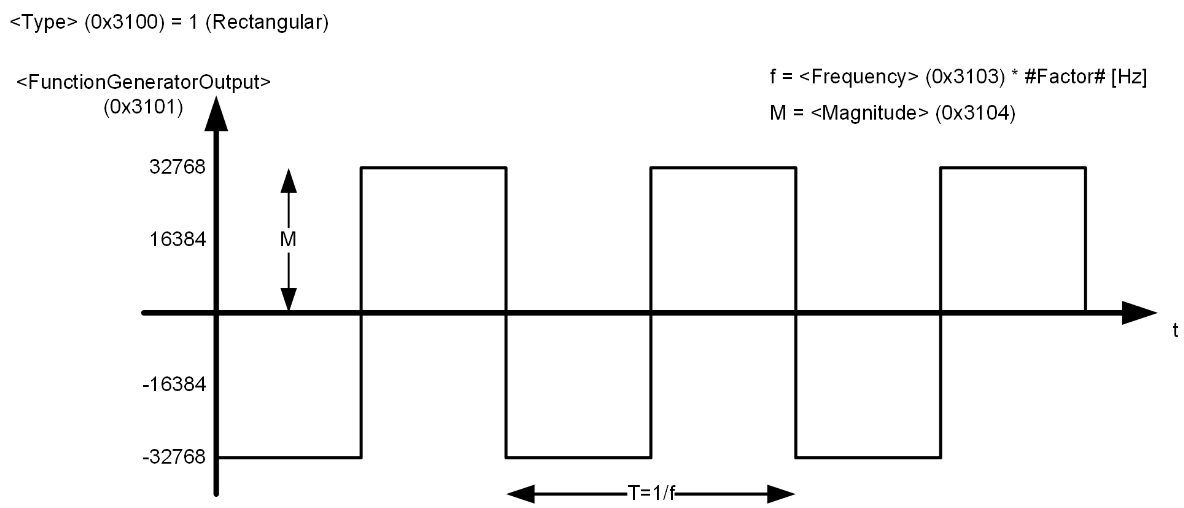


Figure 122: Rectangular output signal (type 1)

7.12.1.2 Triangle output signal (type 2)

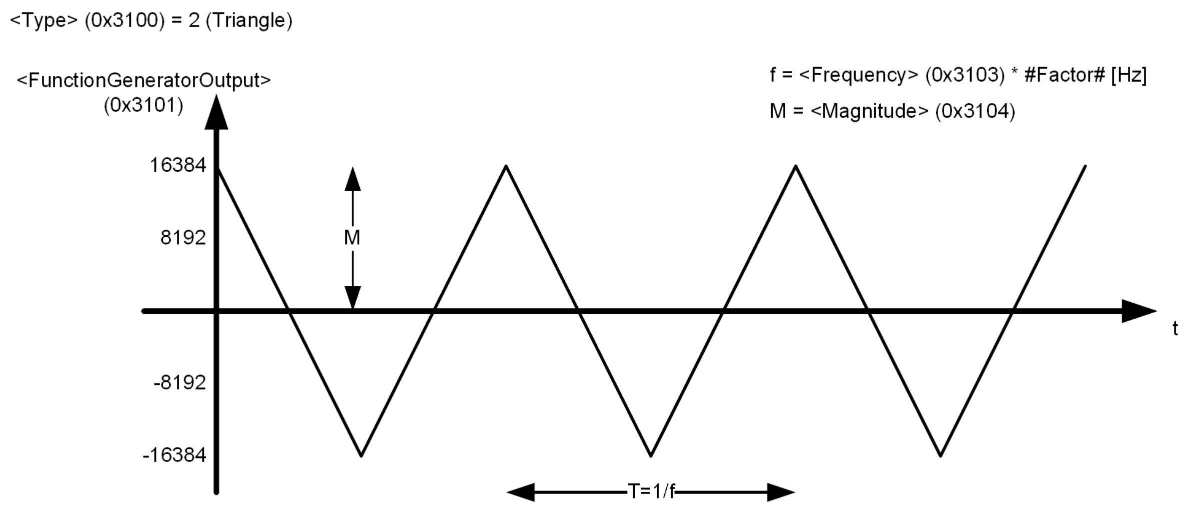


Figure 123: Triangle output signal (type 2)

7.12.1.3 Sawtooth signal (type 3)

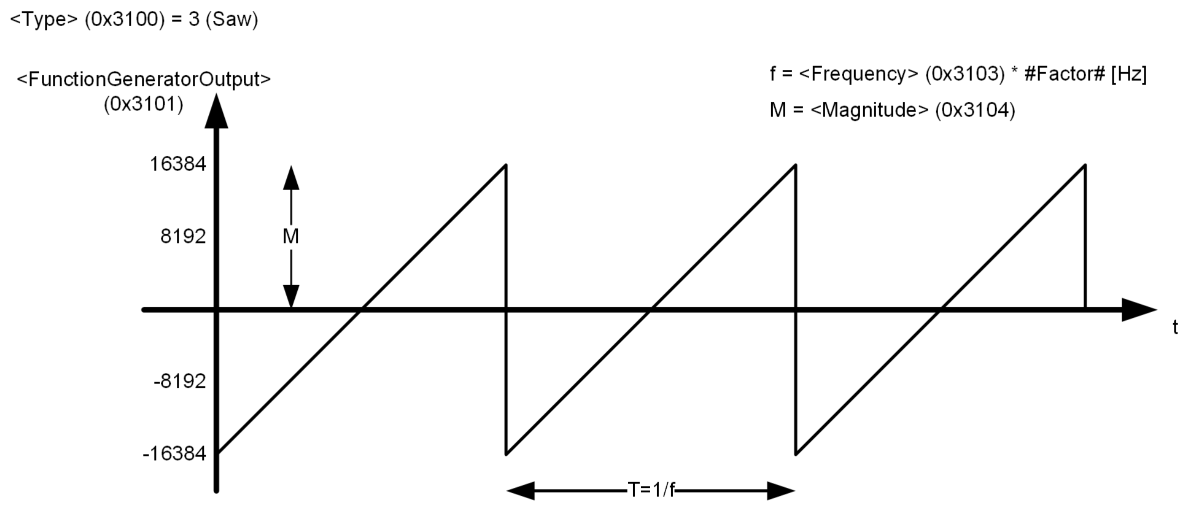


Figure 124: Saw signal (type 3)

❗	The output signal can be inverted by setting the parameter ‹Sign› (0x3107) to −1. ⇨ Chapter "7.12.1.9 Object 0x3107: Sign", page 258

7.12.1.4 Trapezoid signal (type 4)

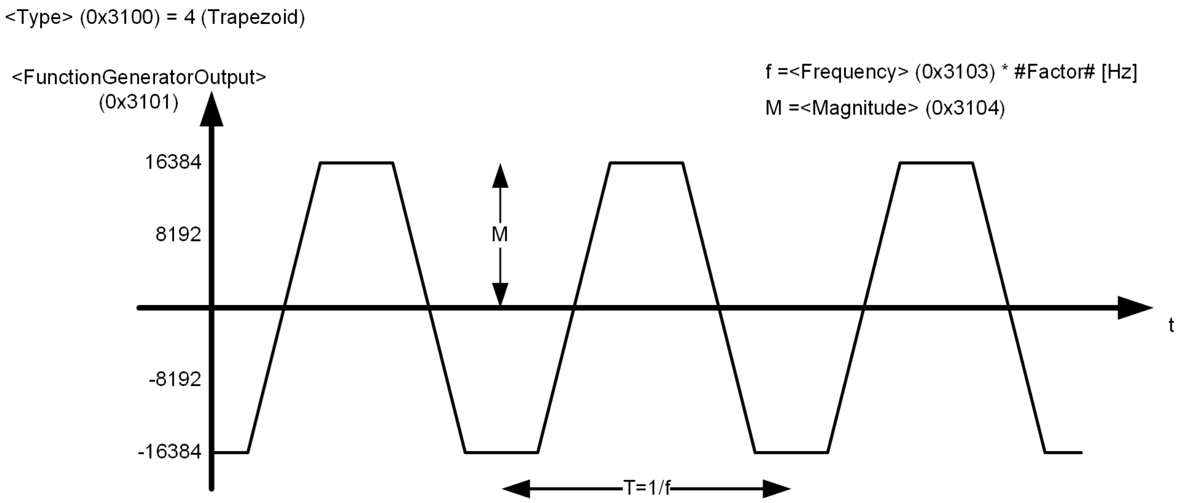


Figure 125: Trapezoid signal (type 4)

7.12.1.5 Sine signal (type 5)

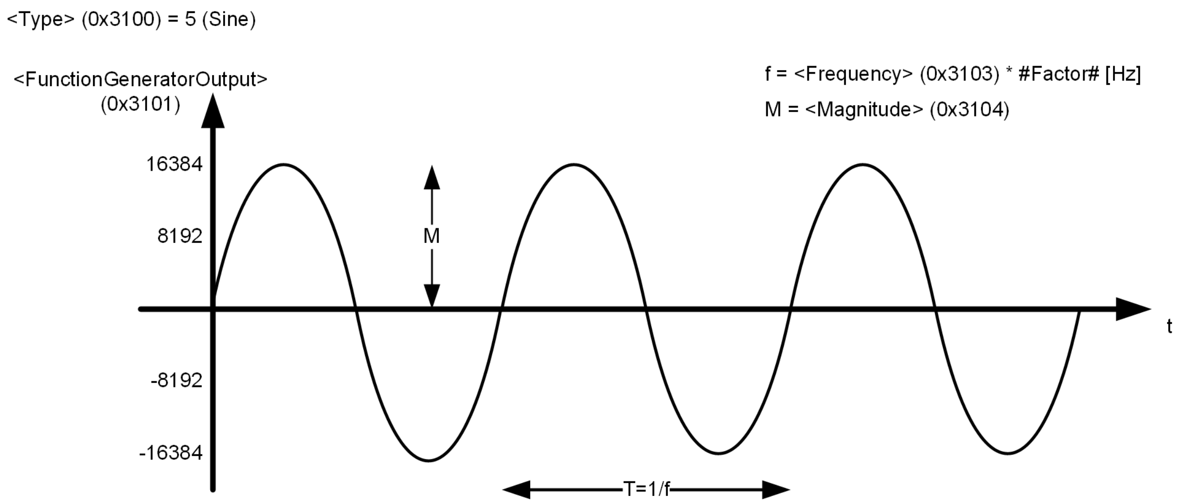


Figure 126: Sine signal (type 5)

7.12.1.6 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	Type	INT8	rw	Y	0…5	0

Value description

Table 112: Possible values of parameter <Type> (0x3100)

<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.

7.12.1.7 Object 0x3104: Magnitude

This parameter is the magnitude of the function generator output signal in increments. To configure e.g. a setpoint of ±100 %, 16384 has to be set as magnitude.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3104	0	Magnitude	INT16	rw	N	0…32767	0

7.12.1.8 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	Offset	INT16	rw	N	INT16	0

7.12.1.9 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	Sign	INT8	rw	Y	−1…1	1

7.12.2 Function generator output signal frequency

The function generator output frequency is defined as:

$$f = \langle\text{Frequency}\rangle\ (0x3103) \cdot \#\text{Factor}\#$$

The #Factor# is dependent on the frequency prefix parameter <FrequencyPrefix> (0x3108). The Frequency unit is Hertz [Hz] or [1/s].

7.12.2.1 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	Frequency	UINT16	rw	Y	1…20000	10

7.12.2.2 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	FrequencyPrefix	INT8	rw	Y	-4…0	0

Value description

Table 113: Possible values of parameter <FrequencyPrefix> (0x3108)

<FrequencyPrefix>	Resulting #Factor#
0	1.0
-1	0.1
-2	0.01
-3	0.001
-4	0.0001

7.12.3 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 mapping: <QSetpointMappingParameter> (0x3320)
  ➪ Chapter "6.1.3.2 Object 0x3320: Q setpoint source selection parameter", page 66
- Pressure setpoint mapping: <PrsSetpointMappingParameter> (0x3310)
  ➪ Chapter "6.1.4.2 Object 0x3310: Pressure setpoint source selection parameter", page 68

7.12.3.1 Object 0x3101: Output

This is the function generator output signal.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3101	0	FunctionGeneratorOutput	INT16	ro	-	INT16	None

7.12.3.2 Object 0x3109: Output 32 bit

This is the function generator output signal as 32 bit value.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3109	0	FunctionGeneratorOutput32	INT32	ro	-	INT32	None

7.12.3.3 Object 0x3102: Square

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. The signal can be used as trigger signal.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3102	0	FunctionGeneratorSquare	INT16	ro	-	INT16	None

Value description

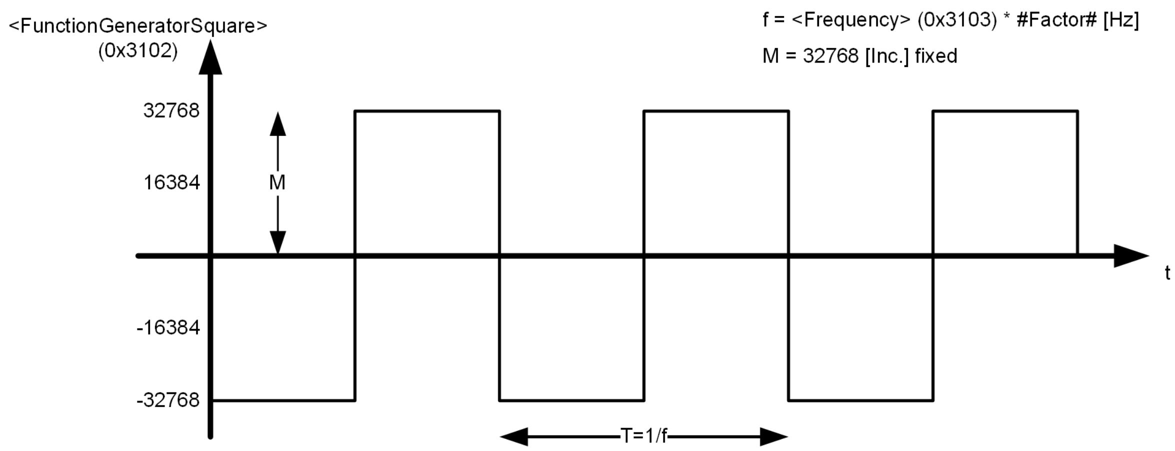


Figure 127: Trigger signal

8 Diagnostics

8.1 Fault reaction

If a malfunction occurs and the fault reaction is enabled, the servo valve software indicates a fault. The corresponding fault is set in the fault state <FaultStatus> (0x2831) and <FaultRetainStatus> (0x2834) parameters.

If no fault reaction for this fault is defined within the parameter <FaultReaction> (0x2830) no further fault reaction is done.

If a specific fault occurs and the response is switched on for that fault, the fault is reported in the parameter <ErrorRegister> (0x1001).

Then the fault code and the error code are saved in an array <StandardErrorField> (0x1003)#1…16 which holds the last sixteen faults occurred.

If the configured error response of the reported error requires a change of the device state, the corresponding device state machine transition will be forced.

8.1.1 Fault reaction flow chart

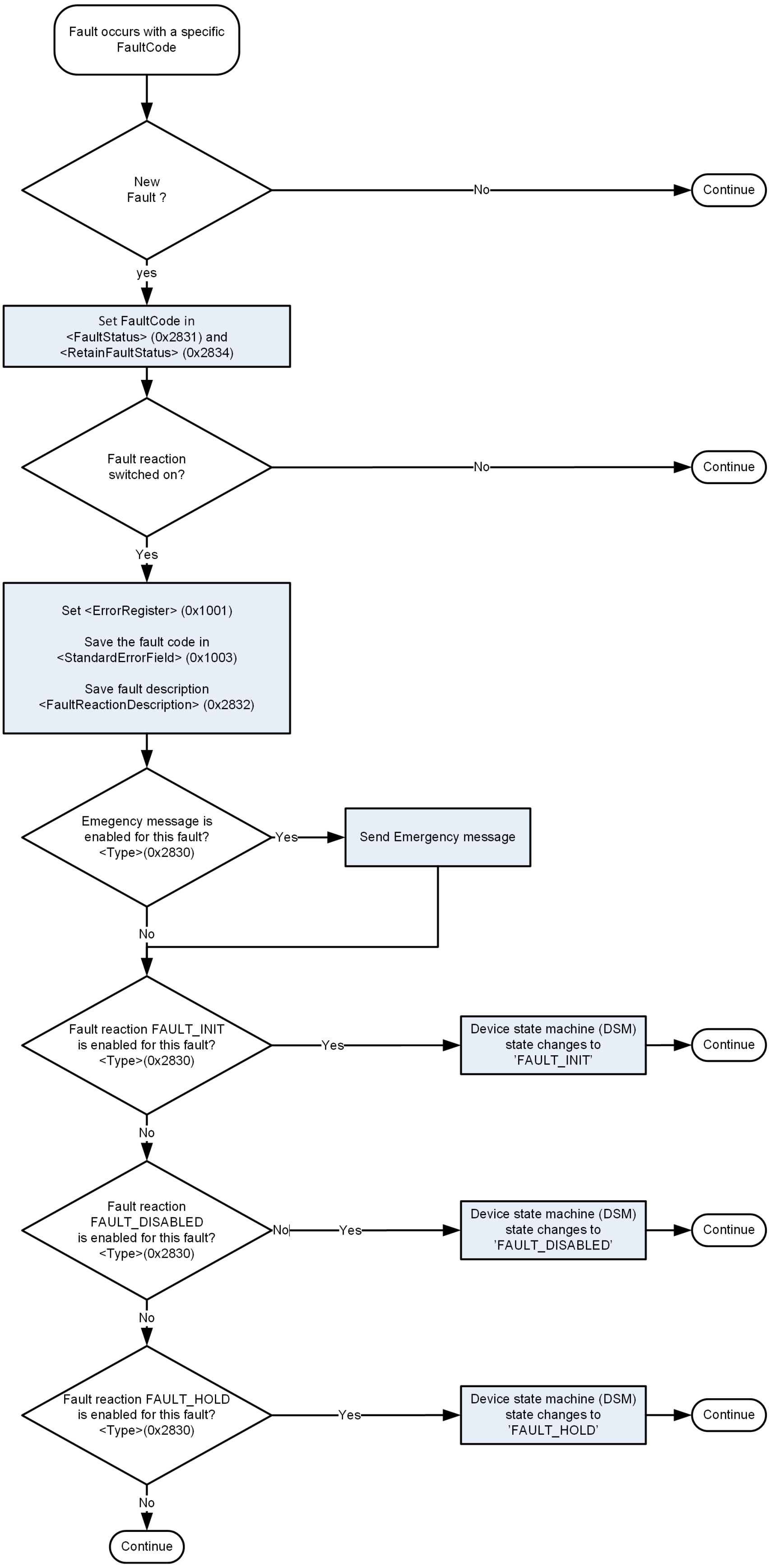


Figure 128: Fault reaction flow chart

8.1.2 Possible fault codes

The possible fault codes and their default settings are shown in the following table. The default settings can differ depending on the valve model or hardware variant.

Table 114: Possible fault codes (part 1 of 12)

Fault code		Fault description	<FaultReaction> (0x2830)						
Dec	Hex		Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0…7)
0	0x00	**Reserved**	1	0	0	0	0	0	0x08
1	0x01	**Error microprocessor core** Microcontroller error which led to the termination of the software program. => Send servo valve back to factory service.	2	0	0	1	1	1	0x3F
2	0x02	**Reserved**	3	0	0	0	0	0	0x08
3	0x03	**Reserved**	4	0	0	0	0	0	0x08
4	0x04	**Reserved**	5	0	0	0	0	0	0x08
5	0x05	**Power supply voltage too low** Power supply voltage exceeds lower limit of 18 V. => Check power supply.	6	0	1	0	1	1	0x57
6	0x06	**Power supply voltage too high** Power supply voltage exceeds upper limit of 32 V. => Check power supply.	7	0	1	0	1	1	0x57
7	0x07	**Internal supply voltage too low** Internal power supply for the processor is too low. => Send servo valve back to factory service.	8	0	1	0	1	1	0x5F
8	0x08	**Internal supply voltage too high** Internal power supply for the processor is too high. => Send servo valve back to factory service.	9	0	1	0	1	1	0x5F
9	0x09	**Reserved**	10	0	0	0	0	0	0x08
10	0x0A	**Reserved**	11	0	0	0	0	0	0x08
11	0x0B	**Actuator current too low**	12	0	1	0	1	1	0x5F
12	0x0C	**Reserved**	13	0	0	0	0	0	0x08
13	0x0D	**Electronics temperature too low (< −40 °C)** Temperature of the electronic is less than −40 °C. => Check environmental temperature. => Consider additional protection cover or heating. Respect temperature limit (−40 °C to +85 °C).	14	0	1	0	1	1	0x57
14	0x0E	**Electronics temperature too high (> 85?°C)** Temperature of the electronic is greater than 85 °C. => Check environmental temperature. => Consider additional protection cover or cooling. Respect temperature limit (−40 °C to +85 °C).	15	0	0	0	1	1	0x11
15	0x0F	**Electronics temperature exceeded (> 105 °C)** Temperature of the electronic is greater than 105 °C. Check environmental temperature. => 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.	16	0	1	0	1	1	0x57

Table 114: Possible fault codes (part 2 of 12)

Fault code		Fault description	<FaultReaction> (0x2830)						
Dec	Hex		Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0…7)
16	0x10	**Overcurrent of power stage** The current controller has detected an error in the current measurement. => Send valve back to factory service.	17	0	1	0	1	1	0x5F
17	0x11	**Pilot/single stage LVDT cable break** Cable break on the internal LVDT detected. => Send servo 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.	18	0	1	0	1	1	0x5F
18	0x12	**Reserved**	19	0	0	0	0	0	0x08
19	0x13	**Pilot/single stage LVDT circuit failure** Internal LVDT sensor conditioning on electronic board is defective. => Send valve back to factory for service.	20	0	1	0	1	1	0x5F
20	0x14	**Main stage LVDT cable break** Cable break on the external LVDT detected. => Check cable connection between pilot and main stage (for example, loosen connector). => If the error still occurs after restarting the servo valve, send the servo valve back to factory for service.	21	0	0	0	1	0	0x10
21	0x15	**Reserved**	22	0	0	0	0	0	0x08
22	0x16	**Reserved**	23	0	0	0	0	0	0x08
23	0x17	**Reserved**	24	0	0	0	0	0	0x08
24	0x18	**Reserved**	25	0	0	0	0	0	0x08
25	0x19	**Reserved**	26	0	0	0	0	0	0x08
26	0x1A	**Reserved**	27	0	0	0	0	0	0x08
27	0x1B	**Reserved**	28	0	0	0	0	0	0x08
28	0x1C	**Analog input 2 supply cable break/short circuit** Supply voltage of analog input 2 is monitored. => Check cable and cable connection. => Check input configuration and sensor adjustments.	29	0	0	0	1	0	0x10
29	0x1D	**Analog input 3 supply cable break/short circuit** Supply voltage of analog input 3 is monitored. => Check cable and cable connection. => Check input configuration and sensor adjustments.	30	0	0	0	1	0	0x10
30	0x1E	**Analog input 4 supply cable break/short circuit** Supply voltage of analog input 4 is monitored. => Check cable and cable connection. Check input configuration and sensor adjustments.	31	0	0	0	1	0	0x10
31	0x1F	**Analog input 0 current too low (4…20 mA)/ADC overflow (voltage)** Measured current is below the adjusted limit (anamonlow). => Check cable and cable connection. => Check input configuration and sensor adjustments. => Check parameter setting (anamonlow).	32	0	0	0	1	0	0x10
32	0x20	**Analog input 1 current too low (4…20 mA)/ADC overflow (voltage)** Measured current is below the adjusted limit (anamonlow). => Check cable and cable connection. => Check input configuration and sensor adjustments. => Check parameter setting (anamonlow).	33	0	0	0	1	0	0x10

Table 114: Possible fault codes (part 3 of 12)

Fault code		Fault description	<FaultReaction> (0x2830)						
Dec	Hex		Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0…7)
33	0x21	**Analog input 2 current too low (4…20 mA)/ADC overflow (voltage)** Measured current is below the adjusted limit (anamonlow). => Check cable and cable connection. => Check input configuration and sensor adjustments. => Check parameter setting (anamonlow).	34	0	0	0	1	0	0x10
34	0x22	**Analog input 3 current too low (4…20 mA)/ADC overflow (voltage)** Measured current is below the adjusted limit (anamonlow). => Check cable and cable connection. => Check input configuration and sensor adjustments. => Check parameter setting (anamonlow).	35	0	0	0	1	0	0x10
35	0x23	**Analog input 4 current too low (4…20 mA)/ADC overflow (voltage)** Measured current is below the adjusted limit (anamonlow). => Check cable and cable connection. => Check input configuration and sensor adjustments. => Check parameter setting (anamonlow).	36	0	0	0	1	0	0x10
36	0x24	**Reserved**	37	0	0	0	0	0	0x08
37	0x25	**Reserved**	38	0	0	0	0	0	0x08
38	0x26	**Reserved**	39	0	0	0	0	0	0x08
39	0x27	**Reserved**	40	0	0	0	0	0	0x08
40	0x28	**Reserved**	41	0	0	0	0	0	0x08
41	0x29	**Reserved**	42	0	0	0	0	0	0x08
42	0x2A	**Reserved**	43	0	0	0	0	0	0x08
43	0x2B	**Reserved**	44	0	0	0	0	0	0x08
44	0x2C	**Reserved**	45	0	0	0	0	0	0x08
45	0x2D	**Power stage** A fault has been detected in the actuator output stage. => Send servo valve back to factory for service.	46	0	0	1	1	1	0x5F
46	0x2E	**Reserved**	47	0	0	0	0	0	0x08
47	0x2F	**Reserved**	48	0	0	0	0	0	0x08
48	0x30	**Internal nonvolatile memory** An internal error during EEPROM initialization / database was detected. => Restart the servo valve. => If the error still occurs after restarting the valve, send the valve back to factory for service.	49	0	1	0	1	1	0x3F
49	0x31	**Reserved**	50	0	0	0	0	0	0x08
50	0x32	**Software coding** An internal software error (deadlock, illegal code operation) caused an restart of the servo valve. => Send servo valve back to factory for service.	51	0	0	1	1	1	0x37
51	0x33	**Software reset (watchdog)** Software reset (watchdog) occurred.	52	0	0	1	1	1	0x37

Table 114: Possible fault codes (part 4 of 12)

Fault code		Fault description	<FaultReaction> (0x2830)						
Dec	Hex		Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0…7)
52	0x34	**Interrupt time exceeded** 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. => Reduce transducer interface and data logger usage => Change the control mode. => If the error occurs often or on a daily basis, even after re-evaluation of all tasks, send the servo valve back to factory for service.	53	0	1	0	1	1	0x57
53	0x35	**Task time exceeded** The internal task time has exceeded. The number of tasks and the time to calculate them exceeds the general processor limitation. => Check the task time. => Reduce the number of tasks (for example data logger, event handler). => Reduce transducer interface usage. => Change the control mode. => If the error occurs often or on a daily basis, even after re-evaluation of all tasks, send the servo valve back to factory for service.	54	0	1	0	1	0	0x56
54	0x36	**Parameter initialization error** Internal error during initialization of RAM parameter settings occurred. => Restart the servo valve. => If the error still occurs after restarting the servo valve, send the valve back to factory for service.	55	0	0	1	1	1	0x37
55	0x37	**Extended EEPROM data load failed** Internal error during initialization of extended parameter settings occurred. => Restart the servo valve. => If the error still occurs after restarting the servo valve, send the servo valve back to factory for service.	56	0	0	1	1	1	0x37
56	0x38	**Customer EEPROM data load failed** Internal error during initialization of customer parameter settings occurred. => Restart the servo valve. => If the error still occurs after restarting the servo valve, send the servo valve back to factory for service.	57	0	0	1	1	1	0x37
57	0x39	**Restore data memory corrupted** The servo valve firmware could not initialize / read the parameters from the non volatile memory. => Restart the servo valve. => If the error still occurs after restarting the servo valve, restore the parameters to factory settings. => Send the servo valve back to factory for service.	58	0	0	1	1	1	0x3F
58	0x3A	**Factory EEPROM data load failed** Internal error during initialization of factory parameter settings occurred. => Restart the servo valve. => If the error still occurs after restarting the servo valve, send the servo valve back to factory for service.	59	0	0	1	1	1	0x3F

Table 114: Possible fault codes (part 5 of 12)

Fault code Dec	Fault code Hex	Fault description	<FaultReaction> (0x2830) Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0…7)
59	0x3B	**Calibration data memory corrupted** The servo valve firmware could not initialize / read out the servo valve database. => Restart the servo valve. => If the error still occurs after restarting the servo valve, send the servo valve back to factory for service.	60	0	0	1	1	1	0x3F
60	0x3C	**Diagnosis EEPROM data load failed** Internal error during initialization of diagnose parameter settings occurred. => Restart the servo valve. => If the error still occurs after restarting the servo valve, send the servo valve back to factory for service.	61	0	0	0	1	1	0x11
61	0x3D	**Reserved**	62	0	0	0	0	0	0x08
62	0x3E	**Reserved**	63	0	0	0	0	0	0x08
63	0x3F	**Reserved**	64	0	0	0	0	0	0x08
64	0x40	**Reserved**	65	0	0	0	0	0	0x08
65	0x41	**Pressure control monitoring** A pressure control monitoring fault is detected. The current pressure deviation exceeds the adjusted limits. => 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 "7.9.8 Pressure control deviation monitoring", page 236	66	0	0	0	1	0	0x11
66	0x42	**Power stage current control monitoring** A current control monitoring fault is detected. Most likely due to a faulty hardware. => Restart the servo valve. => If the error still occurs after restarting the servo valve, send the servo valve back to factory for service.	67	0	0	0	1	0	0x11
67	0x43	**Spool position control monitoring** 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). => Check adjusted deviation limits. => Flush servo valve with clean oil at low pressure. => Switch to open loop operation and move spool end stop to end stop. => Shake the servo valve at low pressure with sweeping sine and/or square signal. For Pilot operated servo valves: => Check pressure levels. For Pilot operated servo 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 servo valves, the main stage is monitored. ⇨ Chapter "7.9.5 Spool position control deviation monitoring", page 231	68	0	1	0	1	1	0x57

Table 114: Possible fault codes (part 6 of 12)

Fault code Dec	Fault code Hex	Fault description	<FaultReaction> (0x2830) Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0…7)
68	0x44	**Pilot spool control monitoring** A pilot spool position control monitoring fault is detected (useful for dual stage valves or pump pilot valves). The actual pilot spool position control deviation exceeds the adjusted limits (for example, due to particles or pollution). => Check adjusted deviation limits. => Flush servo valve with clean oil at low pressure. => Switch to open loop operation and move spool end stop to end stop. => Shake the servo valve at low pressure with sweeping sine and/or square signal. In order to make a customer configured fault reaction working, the position monitoring must be switched on. ⇨ Chapter "7.9.5 Spool position control deviation monitoring", page 231	69	0	0	0	1	0	0x10
69	0x45	**Eventhandler exception** A general event handler fault is detected. => Check the event handler expression for illegal operations. => Clear all event handler expressions => Restart the servo valve. => If the error still occurs after restarting the servo valve, send the servo valve back to factory for service.	70	0	0	0	1	0	0x10
70	0x46	**Local CAN general fault**	71	0	0	0	1	2	0x11
71	0x47	**Reserved**	72	0	0	0	0	0	0x08
72	0x48	**Reserved**	73	0	0	0	0	0	0x08
73	0x49	**Reserved**	74	0	0	0	0	0	0x08
74	0x4A	**Local CAN RPD01 time out** A timeout on local CAN receive PDO 1 occurred. => Check cable and cable connection. => Check local CAN communication interface settings. => Check local CAN RPDO timeout settings.	75	0	0	0	1	0	0x10
75	0x4B	**Local CAN RPD02 time out** A timeout on local CAN receive PDO 2 occurred. => Check cable and cable connection. => Check local CAN communication interface settings. => Check local CAN RPDO timeout settings.	76	0	0	0	1	0	0x10
76	0x4C	**Local CAN RPD03 time out** A timeout on local CAN receive PDO 3 occurred. => Check cable and cable connection. => Check local CAN communication interface settings. => Check local CAN RPDO timeout settings.	77	0	0	0	1	0	0x10
77	0x4D	**Local CAN RPD04 time out** A timeout on local CAN receive PDO 4 occurred. => Check cable and cable connection. => Check local CAN communication interface settings. => Check local CAN RPDO timeout settings.	78	0	0	0	1	0	0x10
78	0x4E	**Local CAN RPD01 data** A problem with the parametrization on local CAN receive PDO 1 was detected. => Check cable and cable connection. => Check local CAN communication interface settings.	79	0	0	0	1	0	0x10

Table 114: Possible fault codes (part 7 of 12)

Fault code Dec	Hex	Fault description	<FaultReaction> (0x2830) Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0…7)
79	0x4F	**Local CAN RPD02 data** A problem with the parametrization on local CAN receive PDO 2 was detected. => Check cable and cable connection. => Check local CAN communication interface settings	80	0	0	0	1	0	0x10
80	0x50	**Local CAN RPD03 data** A problem with the parametrization on local CAN receive PDO 3 was detected. => Check cable and cable connection. => Check local CAN communication interface settings	81	0	0	0	1	0	0x10
81	0x51	**Local CAN RPD04 data** A problem with the parametrization on local CAN receive PDO 4 was detected. => Check cable and cable connection. => Check local CAN communication interface settings	82	0	0	0	1	0	0x10
82	0x52	**Local CAN TPD01 time out** A timeout on local CAN transmit PDO 1 occurred. => Check cable and cable connection. => Check local CAN communication interface settings. => Check local CAN TPDO timeout settings.	83	0	0	0	1	0	0x10
83	0x53	**Local CAN TPD02 time out** A timeout on local CAN transmit PDO 2 occurred. => Check cable and cable connection. => Check local CAN communication interface settings. => Check local CAN TPDO timeout settings.	84	0	0	0	1	0	0x10
84	0x54	**Local CAN TPD03 time out** A timeout on local CAN transmit PDO 3 occurred. => Check cable and cable connection. => Check local CAN communication interface settings. => Check local CAN TPDO timeout settings.	85	0	0	0	1	0	0x10
85	0x55	**Local CAN TPD04 time out** A timeout on local CAN transmit PDO 4 occurred. => Check cable and cable connection. => Check local CAN communication interface settings. => Check local CAN TPDO timeout settings.	86	0	0	0	1	0	0x10
86	0x56	**Local CAN TPD01 data** A problem with the parametrization on local CAN transmit PDO 1 was detected. => Check cable and cable connection. => Check local CAN communication interface settings.	87	0	0	0	1	0	0x10
87	0x57	**Local CAN TPD02 data** A problem with the parametrization on local CAN transmit PDO 2 was detected. => Check cable and cable connection. => Check local CAN communication interface settings.	88	0	0	0	1	0	0x10
88	0x58	**Local CAN TPD03 data** A problem with the parametrization on local CAN transmit PDO 3 was detected. => Check cable and cable connection. => Check local CAN communication interface settings.	89	0	0	0	1	0	0x10

Table 114: Possible fault codes (part 8 of 12)

Fault code		Fault description	<FaultReaction> (0x2830)						
Dec	Hex		Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0...7)
89	0x59	**Local CAN TPD04 data** A problem with the parametrization on local CAN transmit PDO 4 was detected. => Check cable and cable connection. => Check local CAN communication interface settings.	90	0	0	0	1	0	0x10
90	0x5A	**CAN general fault** A general problem in the CAN fieldbus initialization or communication was detected. A problem in the Network State Machine occurred. ➪ Chapter "2.13 Process data object (PDO)", page 23 => 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 289 => If the error still occurs after restarting the servo valve, send the servo valve back to factory for service.	91	0	0	0	1	0	0x10
91	0x5B	**CAN overrun (data frame lost)** The CAN interface or the driver runs out of buffer space, or the bus load is so high that the device cannot handle the traffic, an overload condition is displayed to the application. => Check CAN communication interface settings.	92	0	0	0	1	0	0x10
92	0x5C	**Reserved**	93	0	0	0	0	0	0x08
93	0x5D	**Reserved**	94	0	0	0	0	0	0x08
94	0x5E	**CAN RPD01 time out** A timeout on CAN receive PDO 1 occurred. => Check cable and cable connection. => Check CAN communication interface settings. => Check CAN TPDO timeout settings.	95	0	1	0	1	1	0x57
95	0x5F	**CAN RPD02 time out** A timeout on CAN receive PDO 2 occurred. => Check cable and cable connection. => Check CAN communication interface settings. => Check CAN TPDO timeout settings.	96	0	1	0	1	1	0x57
96	0x60	**CAN RPD03 time out** A timeout on CAN receive PDO 3 occurred. => Check cable and cable connection. => Check CAN communication interface settings. => Check CAN TPDO timeout settings.	97	0	1	0	1	1	0x57
97	0x61	**CAN RPD04 time out** A timeout on CAN receive PDO 4 occurred. => Check cable and cable connection. => Check CAN communication interface settings. => Check CAN TPDO timeout settings.	98	0	1	0	1	1	0x57
98	0x62	**CAN RPD01 data** A problem with the parametrization on CAN receive PDO 1 was detected. ➪ Chapter "2.13 Process data object (PDO)", page 23 => Check cable and cable connection. => Check CAN communication interface settings.	99	0	0	0	1	0	0x10

Table 114: Possible fault codes (part 9 of 12)

Fault code		Fault description	<FaultReaction> (0x2830)						
Dec	Hex		Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0…7)
99	0x63	**CAN RPD02 data** A problem with the parametrization on CAN receive PDO 2 was detected. ➪ Chapter "2.13 Process data object (PDO)", page 23 => Check cable and cable connection. => Check CAN communication interface settings.	100	0	0	0	1	0	0x10
100	0x64	**CAN RPD03 data** A problem with the parametrization on CAN receive PDO 3 was detected. ➪ Chapter "2.13 Process data object (PDO)", page 23 => Check cable and cable connection. => Check CAN communication interface settings.	101	0	0	0	1	0	0x10
101	0x65	**CAN RPD04 data** A problem with the parametrization on CAN receive PDO 4 was detected. ➪ Chapter "2.13 Process data object (PDO)", page 23 => Check cable and cable connection. => Check CAN communication interface settings.	102	0	0	0	1	0	0x10
102	0x66	**CAN TPD01 time out** A timeout on CAN transmit PDO 1 occurred. ➪ Chapter "2.15 Transmit process data object (TPDO)", page 30 => Check cable and cable connection. => Check CAN communication interface settings. => Check CAN TPDO timeout settings.	103	0	0	0	1	0	0x10
103	0x67	**CAN TPD02 time out** A timeout on CAN transmit PDO 2 occurred. ➪ Chapter "2.15 Transmit process data object (TPDO)", page 30 => Check cable and cable connection. => Check CAN communication interface settings. => Check CAN TPDO timeout settings.	104	0	0	0	1	0	0x10
104	0x68	**CAN TPD03 time out** A timeout on CAN transmit PDO 3 occurred. ➪ Chapter "2.15 Transmit process data object (TPDO)", page 30 => Check cable and cable connection. => Check CAN communication interface settings. => Check CAN TPDO timeout settings.	105	0	0	0	1	0	0x10
105	0x69	**CAN TPD04 time out** A timeout on CAN transmit PDO 4 occurred. ➪ Chapter "2.15 Transmit process data object (TPDO)", page 30 => Check cable and cable connection. => Check CAN communication interface settings. => Check CAN TPDO timeout settings.	106	0	0	0	1	0	0x10
106	0x6A	**CAN TPD01 data** A problem with the parametrization on CAN transmit PDO 1 was detected. ➪ Chapter "2.15 Transmit process data object (TPDO)", page 30 => Check cable and cable connection. => Check CAN communication interface settings.	107	0	0	0	1	0	0x10
107	0x6B	**CAN TPD02 data** A problem with the parametrization on CAN transmit PDO 2 was detected. ➪ Chapter "2.15 Transmit process data object (TPDO)", page 30 => Check cable and cable connection. => Check CAN communication interface settings.	108	0	0	0	1	0	0x10

Table 114: Possible fault codes (part 10 of 12)

Fault code Dec	Hex	Fault description	<FaultReaction> (0x2830) Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0…7)
108	0x6C	**CAN TPD03 data** A problem with the parametrization on CAN transmit PDO 3 was detected. ⇨ Chapter "2.15 Transmit process data object (TPDO)", page 30 => Check cable and cable connection. => Check CAN communication interface settings.	109	0	0	0	1	0	0x10
109	0x6D	**CAN TPD04 data** A problem with the parametrization on CAN transmit PDO 4 was detected. ⇨ Chapter "2.15 Transmit process data object (TPDO)", page 30 => Check cable and cable connection. => Check CAN communication interface settings.	110	0	0	0	1	0	0x10
110	0x6E	**CAN life guard error or heartbeat error** 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 guard time and life time factor. ⇨ Chapter "2.12 Network management (NMT) node guarding", page 22 ⇨ Chapter "2.11 Network management (NMT) heartbeat", page 21 => Check cable and cable connection. => Check CAN communication interface settings. => Check guarding settings. => Check NMT state machine. => Check guard request from PLC (master).	111	0	1	0	1	1	0x57
111	0x6F	**CAN SYNC producer time out** A synchronization problem on CAN occurred. The synchronization signal could not be received within timeout. ⇨ Chapter "2.18 Synchronization (SYNC)", page 38 => Check cable and cable connection. => Check CAN communication interface settings. => Check corresponding receive PDO and Transmission type.	112	0	0	0	1	0	0x10
112	0x70	**CAN SYNC consumer time out** A synchronization problem on CAN occurred. ⇨ Chapter "2.18 Synchronization (SYNC)", page 38 ⇨ Chapter "2.13 Process data object (PDO)", page 23 ⇨ Chapter "2.15 Transmit process data object (TPDO)", page 30 => Check cable and cable connection. => Check CAN communication interface settings. => Check corresponding receive PDO and transmission type.	113	0	0	0	1	0	0x10
113	0x71	**EtherCAT communication fault (not implemented)**	114	0	0	0	1	0	0x10
114	0x72	**EtherCAT RPDO time out** A timeout on EtherCAT receive PDO occurred. => Check cable and cable connection. => Check EtherCAT communication interface settings.	115	0	1	0	0	1	0x57
115	0x73	**EtherCAT RPDO data** A problem with the parametrization on EtherCAT receive PDO was detected. => Check cable and cable connection. => Check EtherCAT communication interface settings.	116	0	0	0	1	0	0x10
116	0x74	**EtherCAT TPDO time out (not implemented)** A timeout on EtherCAT transmit PDO occurred. => Check cable and cable connection. => Check EtherCAT communication interface settings.	117	0	0	0	0	0	0x08

Table 114: Possible fault codes (part 11 of 12)

Fault code Dec	Hex	Fault description	<FaultReaction> (0x2830) Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0...7)
117	0x75	**EtherCAT TPDO data** A problem with the parametrization on EtherCAT transmit PDO was detected. => Check cable and cable connection. => Check EtherCAT communication interface settings.	118	0	0	0	1	0	0x10
118	0x76	**Reserved**	119	1	0	0	0	0	0x08
119	0x77	**Reserved**	120	0	0	0	0	0	0x08
120	0x78	**Pressure sensor monitoring failure** A problem with the internal pressure sensor / pressure sensor module was detected. => Send servo valve back to factory service.	121	0	0	0	1	0	0x10
121	0x79	**Function initialization failure** An internal error during startup occurred. => Send servo valve back to factory service.	122	0	0	1	1	1	0x37
122	0x7A	**Utility averaging failure** Internal mapping or calculation fault during average calculation occurred. Averaging must be switched on and parameter must be mapped. => Check parameter settings. => Restart average calculation.	123	0	0	0	1	1	0x11
123	0x7B	**CiA 408 state machine interaction force fault** Relation between state machines. The transitions C5 and C8 in the communication state machine ('Pre-Operational' -> 'Stopped' and 'Operational' -> 'Stopped') or the transitions C12, C13 and C14 in the communication state machine ('Operational' -> 'Reset Communication', 'Stopped' -> 'Reset Communication' and 'Pre-operational' -> 'Reset Communication') forced the transition TR8 in the device state machine ('DEVICE_MODE_ACTIVE' -> 'FAULT_REACTION'). => To avoid this fault, shut down the servo valve application state machine before shutting down the network state machine.	124	0	1	0	1	1	0x57
124	0x7C	**Enable input fault** A problem with the safety function of the enable input is detected. Fault is indicated if the two internal redundant signals of the enable input have different levels. The fault logic is only active if safety function is enabled. => Send servo valve back to factory service.	125	0	1	0	1	1	0x57
125	0x7D	**Pressure sensor module offset correction fault** This fault can only occur if a pressure sensor module is mounted on the valve. This fault is indicated if there is a problem with the zero correction of pressure sensor module. => Make sure all pressure ports are depressurized and start the offset calibration again.	126	0	1	0	1	1	0x57
126	0x7E	**Fieldbus synchronization fault** Synchronization is not between limits. => Make sure the jitter of the cyclic messages on the fieldbus are within the defined limits.	127	0	0	0	1	1	0x11
127	0x7F	**Fieldbus initialization fault** An error occurred during the fieldbus initialization. => Send servo valve back to factory service.	128	0	1	0	1	1	0x57
128	0x80	**Reserved**	129	0	0	0	0	0	0x08

Table 114: Possible fault codes (part 12 of 12)

Fault code		Fault description	<FaultReaction> (0x2830)						
Dec	Hex		Sub-Index	HOLD (Bit 7)	DISABLED (Bit 6)	INIT (Bit 5)	EMCY (Bit 4)	On/Off (Bit 0)	Default (Bits 0…7)
129	0x81	**I2C general error** A servo valve internal bus error has occurred. The internal communication is broken or disturbed. => Send servo valve back to factory service.	130	0	0	1	1	1	0x3F
130	0x82	**CAN_rpdo0_less_data_than_mapped** This message indicates that more bytes are mapped in the RPDO (see Object 0x1600, RPDO mapping) than received from the bus. => Reduce the number of bytes mapped in the RPDO or additional parameters should be mapped in the TPDO mapping of the PDO producer node.	131	0	0	0	1	1	0x11
131	0x83	**CAN_rpdo1_less_data_than_mapped** This message indicates that more bytes are mapped in the RPDO (see Object 0x1601, RPDO mapping) than received from the bus. => Reduce the number of bytes mapped in the RPDO or additional parameters should be mapped in the TPDO mapping of the PDO producer node.	132	0	0	0	1	1	0x11
132	0x84	**CAN_rpdo2_less_data_than_mapped** This message indicates that more bytes are mapped in the RPDO (see Object 0x1602, RPDO mapping) than received from the bus. => Reduce the number of bytes mapped in the RPDO or additional parameters should be mapped in the TPDO mapping of the PDO producer node.	133	0	0	0	1	1	0x11
133	0x85	**CAN_rpdo3_less_data_than_mapped** This message indicates that more bytes are mapped in the RPDO (see Object 0x1602, RPDO mapping) than received from the bus. => Reduce the number of bytes mapped in the RPDO or additional parameters should be mapped in the TPDO mapping of the PDO producer node.	134	0	0	0	1	1	0x11
134	0x86	**Local_CAN_rpdo0_less_data_than_mapped** This message indicates that more bytes are mapped in the RPDO (see Object 0x5600, RPDO mapping) than received from the bus. => Reduce the number of bytes mapped in the RPDO or additional parameters should be mapped in the TPDO mapping of the PDO producer node.	135	0	0	0	1	1	0x11
135	0x87	**CAN_rpdo1_less_data_than_mapped** This message indicates that more bytes are mapped in the RPDO (see Object 0x5601, RPDO mapping) than received from the bus. => Reduce the number of bytes mapped in the RPDO or additional parameters should be mapped in the TPDO mapping of the PDO producer node.	136	0	0	0	1	1	0x11
136	0x88	**CAN_rpdo2_less_data_than_mapped** This message indicates that more bytes are mapped in the RPDO (see Object 0x5602, RPDO mapping) than received from the bus. => Reduce the number of bytes mapped in the RPDO or additional parameters should be mapped in the TPDO mapping of the PDO producer node.	137	0	0	0	1	1	0x11
137	0x89	**CAN_rpdo3_less_data_than_mapped** This message indicates that more bytes are mapped in the RPDO (see Object 0x5602, RPDO mapping) than received from the bus. => Reduce the number of bytes mapped in the RPDO or additional parameters should be mapped in the TPDO mapping of the PDO producer node.	138	0	0	0	1	1	0x11
…	…	**Reserved**	…	…	…	…	…	…	…
159	0xA0	**Reserved**	160	…	…	…	…	…	…

8.1.3 Fault reaction

The fault reaction parameter <FaultReaction> (0x2830) can be used to configure the fault behavior for each fault code.

8.1.3.1 Object 0x2830: Fault reaction

The fault reaction for each fault can be configured by selecting different fault reactions.

!	To get the sub-index of the fault reaction <FaultReaction> (0x2830), the fault code must be incremented by one. <FaultReaction> (0x2830)#[FaultCode+1]

FaultReaction							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2830	1…130	FaultReaction	INT8	rw	Y	⇨ Table 114, page 263	

Value description

Table 115: Fault reaction settings

<FaultReaction> Bit	**Description**
Bit 0 (0x0001) ON/OFF	Switches on/off the fault reaction configuration. This flag must be activated for a configured fault response to be executed.
Bit 1…2	Reserved
Bit 3 (0x0008) 'CONFIG'	This flag indicates that this fault reaction cannot be changed by the user. This flag is read only.
Bit 4 (0x0010) 'EMERGENCY'	If this flag is set, occurring faults initiate the corresponding Emergency Message on the fieldbus.
Bit 5 (0x0020) 'FAULT_INIT'	If this flag is set, occurring faults force the device state machine to switch to the 'FAULT_INIT' state. The power stage of the device is switched off, while almost all device functions are stopped. The device must be re-enabled with the control word or the enable signal in order to return into normal operation.
Bit 6 (0x0040) 'FAULT_DISABLED'	If this flag is set, occurring faults force the device state machine to switch to the 'FAULT_DISABLED' state. 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.
Bit 7 (0x0080) 'FAULT_HOLD'	If this flag is set, occurring faults force the device state machine to switch to the 'FAULT_HOLD' state. If a malfunction for the monitored fault is detected, the device enters the 'FAULT_HOLD' state. The spool position is controlled to the hold setpoint. The device must be re-enabled with the control word or the enable signal in order to return into normal operation.
Bit 8…15	Reserved for internal use

8.1.4 Error codes depending on fault codes

The following table combines the Moog specific fault codes with the error codes sent with an emergency message.

Details about the error codes used for monitoring: ⇨ Chapter "7.9 Monitoring", page 226

Table 116: Possible error codes depending on fault codes (part 1 of 3)

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
20, 21, 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
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
124	5301	Enable input fault
45	5410	Power driver
46	5510	RAM
47	5520	EPROM
48	5530	EEPROM
120	5801	Pressure sensor monitoring failed
125	5802	Pressure sensor module offset correction fault
51	6010	Software reset (watchdog)
50	6101	Software coding
52	6102	Interrupt time exceeded
53	6103	Task time exceeded
49	6104	Out of memory
121	6105	Function initialization failure
122	6106	Utility averaging failure
69	6201	Event handler
55	6311	Node identifier data
56	6312	User data

Table 116: Possible error codes depending on fault codes (part 2 of 3)

Fault code	Error code defined in Device Profile Fluid Power	Error description
57	6313	Restore data
58	6314	Factory data
59	6315	Calibration data
60	6316	Diagnosis data
54	6320	Parameter error
129	7000	I2C general error
119	7002	DIP switch failure
90, 127, 128	8100	Fieldbus communication
70	8101	Local CAN communication
123	8102	CiA 408 state machine interaction force fault
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	8231	RPD01 time out
95	8232	RPD02 time out
96	8233	RPD03 time out
97	8234	RPD04 time out
102	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, 126	823A	SYNC consumer time out
113	823B	EtherCAT communication fault
114	823C	EtherCAT RPDO time out
115	823D	EtherCAT RPDO data
116	823E	EtherCAT TPDO time out
117	823F	EtherCAT TPDO data
98	8241	RPD01 data
99	8242	RPD02 data
100	8243	RPD03 data
101	8244	RPD04 data
106	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

Table 116: Possible error codes depending on fault codes (part 3 of 3)

Fault code	Error code defined in Device Profile Fluid Power	Error description
85	8258	Local TPD04 time out
118	8260	Reserved
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, 68	8301	Spool 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

8.1.4.1 Object 0x604E: Last error code

This parameter shows the last occurred error code.

FaultReaction							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x604E	0	DeviceErrorCode	UINT16	ro	-	UINT16	None

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 (5 words with 32 bit, built with the sub-indexes 1…5 of the fault status) stands for a fault code. The corresponding fault code is equal to the bit position in the bit field.

8.1.5.1 Object 0x2831: Fault status

Actual reported faults in bit coded form.

FaultReaction							
Index	Sub-index	Parameter name	Data type	Access	Per-sistence	Value range	Default
0x2831	1	FaultStatus (fault code 0…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
0x2831	5	FaultStatus (fault code 129…160)	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	Per-sistence	Value range	Default
0x2834	1	FaultRetainStatus (fault code 0…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
0x2834	5	FaultRetainStatus (fault code 129…160)	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 280

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

Table 117: Possible values of parameter <ErrorRegister> (0x1001)

<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

If a specific error occurs, the corresponding bit is 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 indicated which trigger 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	UINT8	rw	N	UINT8	0
0x1003	1…16	StandardErrorField	UINT32	ro	-	UINT32	0

Value description

Table 118: Possible values of parameter <PreDefinedErrorField> (0x1003)

Parameter	**Description**
<NumberOfErrors>	Number of actual recorded errors.
<StandardErrorField>	Array of recorded errors.

<StandardErrorField>				
Byte	3	2	1	0
Description	Additional information		Error code ⇨ Chapter "8.1.4 Error codes depending on fault codes", page 276	
	Reserved	Fault code ⇨ Chapter "8.1.2 Possible fault codes", page 263		

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 occurrence time ‹CumulativePowerOnTime›. The format is:

"A/B @ M min STRING"

With:

A = displayed fault number
B = count of faults
M = fault occurrence time
STRING = description.

FaultReaction							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2832	0	FaultReactionDescription	STRING(128)	ro	-	STRING(128)	""

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

If enabled by parameter 0x2830, a configured fault 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 have disappeared. In this case the fault code 0x00 (Error reset or no error) will be sent.

➪ Chapter "2.19.1 Object 0x1014: EMCY protocol COB-ID configuration", page 40
➪ Chapter "8.1.4 Error codes depending on fault codes", page 276

The coding of the emergency message is as follows:

<table>
<tr><td>Byte</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td rowspan="2">Description</td><td colspan="5">Moog-specific error code</td><td rowspan="2">Error register</td><td colspan="2" rowspan="2">Emergency error code
➪ Chapter "8.1.4 Error codes depending on fault codes", page 276</td></tr>
<tr><td colspan="4">Cumulative power on time</td><td>Fault code
➪ Chapter "8.1.2 Possible fault codes", page 263</td></tr>
</table>

The displayed byte order of the transmitted emergency message is depending on the fieldbus 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 power on time since production 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	
	Fault occurrence timestamp <CumulativePowerOnTime> 0x280D				Fault code			
Content	0x00	0x00	0x03	0xE8	0x05	0x04	0x34	0x12
Result	0x000003E805043412							

8.1.10 Fault disappears

If all faults 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 fieldbus master. If the device state machine (DSM) is in the state 'FAULT_HOLD', 'FAULT_DISABLED' or 'FAULT_INIT', the DSM must be set to 'ACTIVE' again. This can be done by the #Control Word# or the enable signal (digital input 0).

➪ Chapter "8.1.11 Fault acknowledgment", page 283

8.1.11 Fault acknowledgment

Depending on the configured fault reaction, the servo valve sends out an emergency message and changes the device state machine to the corresponding fault state.

➪ Chapter "8.1.3 Fault reaction", page 275

In order to get out of the fault state the fault must be acknowledged. This can be achieved by
- Sending the #ControlWord# to the servo valve with the bit 3 (fault reset) is set.
- Toggling the enable signal (digital input 0).

➪ Chapter "5.1.4 DSM control word source selection / local mode", page 58
➪ Chapter "5.1 Device state machine (DSM)", page 50
➪ Chapter "6.6.2 Object 0x5E41: Digital output configuration", page 104

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	ErrorCode	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 production and valve is powered on.

ErrorHandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2823	1…5	ErrorTime	UINT32	ro	-	UINT32	0

8.2.3 Object 0x2824: Internal error additional information

This parameter represents additional information for each stored error. This can be used for code debugging. The last occurred error is shown at the top of the table.

ErrorHandler							
Index	Sub-index	Parameter name	Data type	Access	Per-sis-tence	Value range	Default
0x2824	0…20	ErrorAdditionalInformation	UNIT32	ro	Y	UINT32	None

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.

Table 119: SDO Abort Codes

SDO Abort Code	Description
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	Value range exceeded / invalid value for parameter (download only). Probably the value range has been exceeded.
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.
0x060A0023	Resource not available: SDO connection
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.

8.4 Object 0x300E: CAN driver status

This parameter represents the CAN transmit chips error and status register.

CAN							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x300E	1	CAN chip Error and Status Register	UINT32	ro	N	UINT32	None
0x300E	2	CAN chip Transmit Error Counter according CAN protocol specification, Version 2.0	UINT32	ro	N	UINT32	None
0x300E	3	CAN chip Receive Error Counter according CAN protocol specification, Version 2.0	UINT32	ro	N	UINT32	None

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 288
- 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 289

!	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.

Table 120: Behavior of saveable and volatile parameters

		Volatile parameters (persistence = "N")	
Operation	**Non-volatile parameters (persistence = "Y")**	**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.

!	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	1
0x1010	2	SaveCommunicationParameters	UINT32	rw	N	UINT32	1
0x1010	3	SaveApplicationParameters	UINT32	rw	N	UINT32	1
0x1010	4	SaveManufacturerDefinedParameters	UINT32	rw	N	UINT32	1

Value description

Table 121: Possible values of parameter 0x1010

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.

On read access to the appropriate sub-index the device provides information about its storage functionality: The value '1' indicates that the device needs this save command to save this parameter group permanently in the non-volatile memory.

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	Persistence	Value range	Default
0x1011	1	RestoreAllDefaultParameters	UINT32	rw	N	UINT32	1
0x1011	2	RestoreCommunicationDefaultParameters	UINT32	rw	N	UINT32	1
0x1011	3	RestoreApplicationDefaultParameters	UINT32	rw	N	UINT32	1
0x1011	4	RestoreManufacturerDefinedDefaultParameters	UINT32	rw	N	UINT32	1

Value description

Table 122: Possible values of parameter 0x1011

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.

After restoring the factory parameters the valve will generate a new start to get the factory setting values effective.

➪ Chapter "5.1 Device state machine (DSM)", page 50
➪ Chapter "2.10 Network management (NMT) state machine", page 17

The following table shows the necessary state changes to activate the restored values.

Table 123: State changes needed to activate the restored values

Parameter	Set the device state machine to 'INIT' #ControlWord# MHD = 000	Set the fieldbus network state machine to 'INIT'
<RestoreAllDefaultParameters>	X	X
<RestoreCommunicationDefaultParameters>		X
<RestoreApplicationDefaultParameters>	X	
<RestoreManufacturerDefinedDefaultParameters>	X	

On read access to the appropriate sub-index the device provides information about its parameter restoring capability:
The value '1' indicates that the device is able to restore this parameter group from the non-volatile memory.

10 Object dictionary

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

Table 124: Object dictionary (part 1 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x0002	0	N	dums08	CiA 301	DataType INTEGER8	INT8	ro	-	INT8	8
0x0003	0	N	dums16	CiA 301	DataType INTEGER16	INT16	ro	-	INT16	16
0x0004	0	N	dums32	CiA 301	DataType INTEGER32	INT32	ro	-	INT32	32
0x0005	0	N	dumu08	CiA 301	DataType UNSIGNED8	UINT8	ro	-	UINT8	8
0x0006	0	N	dumu16	CiA 301	DataType UNSIGNED16	UINT16	ro	-	UINT16	16
0x0007	0	N	dumu32	CiA 301	DataType UNSIGNED32	UINT32	ro	-	UINT32	32
0x0008	0	N	dumf32	CiA 301	DataType REAL32	REAL32	ro	-	REAL32	32
0x0009	0	N	dumchr	CiA 301	DataType VISIBLE_STRING	STRING(64)	rw	N	STRING(64)	""
0x1000	0	N	devtyp	CiA 301	CANopen DeviceType	UINT32	ro	-	UINT32	Device specific value
0x1001	0	Y	errreg	CiA 301	CANopen ErrorRegister	UINT8	ro	-	UINT8	None
0x1002	0	Y	manstsreg	CiA 301	CANopen ManufacturerStatusRegister	UINT32	ro	-	UINT32	
0x1003	0	N	preerrnum	CiA 301	CANopen NumberOfErrors	UINT8	rw	N	UINT8	0

Table 124: Object dictionary (part 2 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x1003	1…16	N	preerrfld[0…15]	CiA 301	CANopen StandardErrorField	UINT32	ro	-	UINT32	0
0x1005	0	N	sncmsgcob	CiA 301	CANopen CobIdSyncMessage	UINT32	rw	Y	1…0x400007FF	0x00000080
0x1006	0	N	comcycper	CiA 301	CANopen CommunicationCyclePeriod	UINT32	rw	Y	UINT32	0
0x1007	0	N	sncwinlen	CiA 301	CANopen SynchronousWindowLength	UINT32	rw	Y	UINT32	
0x1008	0	N	mandevnam	CiA 301	CANopen ManufacturerDeviceName	STRING(64)	ro	-	None	Device specific value
0x1009	0	N	manhdwver	CiA 301	CANopen ManufacturerHardwareVersion	STRING(64)	ro	-	None	Device specific value
0x100A	0	N	mansfwver	CiA 301	CANopen ManufacturerSoftwareVersion	STRING(64)	ro	-	None	Device specific value
0x100B	0	N	nodide	CiA 301	CANopen NodeId	UINT32	ro	-	UINT32	127
0x100C	0	N	grdtim	CiA 301	CANopen GuardTime	UINT16	rw	Y	UINT16	
0x100D	0	N	liftimfct	CiA 301	CANopen LifeTimeFactor	UINT8	rw	Y	UINT8	
0x1010	1	N	stopar[0]	CiA 301	CANopen, Save SaveAllParameters	UINT32	rw	N	UINT32	1
0x1010	2	N	stopar[1]	CiA 301	CANopen, Save SaveCommunicationParameters	UINT32	rw	N	UINT32	1
0x1010	3	N	stopar[2]	CiA 301	CANopen, Save SaveApplicationParameters	UINT32	rw	N	UINT32	1
0x1010	4	N	stopar[3]	CiA 301	CANopen, Save SaveManufacturerDefinedParameters	UINT32	rw	N	UINT32	1
0x1011	1	N	rstpar[0]	CiA 301	CANopen, Restore RestoreAllDefaultParameters	UINT32	rw	N	UINT32	1
0x1011	2	N	rstpar[1]	CiA 301	CANopen, Restore RestoreCommunicationDefaultParameters	UINT32	rw	N	UINT32	1

Table 124: Object dictionary (part 3 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x1011	3	N	rstpar[2]	CiA 301	CANopen, Restore RestoreApplicationDefaultParameters	UINT32	rw	N	UINT32	1
0x1011	4	N	rstpar[3]	CiA 301	CANopen, Restore RestoreManufacturerDefinedDefaultParameters	UINT32	rw	N	UINT32	1
0x1012	0	N	tspmsgcob	CiA 301	CANopen CobIdTimeStampMessage	UINT32	rw	Y	1…0x7FF	<NodeID> (0x100B) + 0x0081
0x1013	0	Y	hghrsltsp	CiA 301	CANopen HighResolutionTimeStamp	UINT32	rw	Y	UINT32	
0x1014	0	N	emgmsgcob	CiA 301	CANopen CobIdEmergencyMessage	UINT32	ro	-	0x81…0xFF	$NODEID + 0x0080
0x1015	0	N	emgmsginh	CiA 301	CANopen InhibitTimeEmergencyMessage	UINT16	rw	Y	UINT16	
0x1016	1	N	harbeacon	CiA 301	CANopen ConsumerHeartbeatTime	UINT32	rw	Y	UINT32	
0x1017	0	N	harbeapro	CiA 301	CANopen ProducerHeartbeatTime	UINT16	rw	Y	UINT16	
0x1018	1	N	ideobj[0]	CiA 301	CANopen, IdentityObject VendorId	UINT32	ro	-	0x28…0x28	40
0x1018	2	N	ideobj[1]	CiA 301	CANopen, IdentityObject ProductCode	UINT32	ro	-	0…0xFFFFFFFF	Device specific value
0x1018	3	N	ideobj[2]	CiA 301	CANopen, IdentityObject RevisionNumber	UINT32	ro	-	0…0xFFFFFFFF	Device specific value
0x1018	4	N	ideobj[3]	CiA 301	CANopen, IdentityObject SerialNumber	UINT32	ro	-	0…0xFFFFFFFF	Device specific value
0x1019	0	N	snccntmax	CiA 301	CANopen SynchronousCounterOverflowValue	UINT8	rw	Y	UINT8	
0x1029	1	N	comerrbeh	CiA 301	CANopen CommunicationErrorBehavior	UINT8	rw	Y	UINT8	
0x1200	1	N	sdoclncob	CiA 301	CANopen CobIDSDO	UINT32	ro	-	UINT32	$NODEID + 0x0600
0x1200	2	N	sdosrvcob	CiA 301	CANopen CobIDSDO	UINT32	ro	-	UINT32	$NODEID + 0x0580

Table 124: Object dictionary (part 4 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x1400	1	N	pdrcob[0]	CiA 301	CANopen, RPdo_CobIdUsedByPdo 1stRPdo_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	$NODEID + 0x0200
0x1400	2	N	pdrtrn[0]	CiA 301	CANopen, RPdo_TransmissionType 1stRPdo_TransmissionType	UINT8	rw	Y	UINT8	255
0x1400	5	N	pdrtim[0]	CiA 301	CANopen, RPdo_EventTimer 1stRPdo_EventTimer	UINT16	rw	Y	0…25000	
0x1401	1	N	pdrcob[1]	CiA 301	CANopen, RPdo_CobIdUsedByPdo 2ndRPdo_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	$NODEID + 0x0300
0x1401	2	N	pdrtrn[1]	CiA 301	CANopen, RPdo_TransmissionType 2ndRPdo_TransmissionType	UINT8	rw	Y	UINT8	255
0x1401	5	N	pdrtim[1]	CiA 301	CANopen, RPdo_EventTimer 2ndRPdo_EventTimer	UINT16	rw	Y	0…25000	
0x1402	1	N	pdrcob[2]	CiA 301	CANopen, RPdo_CobIdUsedByPdo 3rdRPdo_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	$NODEID + 0x0400
0x1402	2	N	pdrtrn[2]	CiA 301	CANopen, RPdo_TransmissionType 3rdRPdo_TransmissionType	UINT8	rw	Y	UINT8	255
0x1402	5	N	pdrtim[2]	CiA 301	CANopen, RPdo_EventTimer 3rdRPdo_EventTimer	UINT16	rw	Y	0…25000	
0x1403	1	N	pdrcob[3]	CiA 301	CANopen, RPdo_CobIdUsedByPdo 4thRPdo_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	$NODEID + 0x0500
0x1403	2	N	pdrtrn[3]	CiA 301	CANopen, RPdo_TransmissionType 4thRPdo_TransmissionType	UINT8	rw	Y	UINT8	255
0x1403	5	N	pdrtim[3]	CiA 301	CANopen, RPdo_EventTimer 4thRPdo_EventTimer	UINT16	rw	Y	0…25000	
0x1600	0	N	pdrmapnum	CiA 301	CANopen RPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	1
0x1600	1	N	pdrmap[0]	CiA 301	CANopen, RPdoMapping RPdoMapping1stApplicationObject	UINT32	rw	Y	UINT32	0x60400010
0x1600	2	N	pdrmap[1]	CiA 301	CANopen, RPdoMapping RPdoMapping2ndApplicationObject	UINT32	rw	Y	UINT32	0
0x1600	3	N	pdrmap[2]	CiA 301	CANopen, RPdoMapping RPdoMapping3rdApplicationObject	UINT32	rw	Y	UINT32	0

Table 124: Object dictionary (part 5 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x1600	4	N	pdrmap[3]	CiA 301	CANopen, RPdoMapping RPdoMapping4thApplicationObject	UINT32	rw	Y	UINT32	0
0x1600	5	N	pdrmap[4]	CiA 301	CANopen, RPdoMapping RPdoMapping5thApplicationObject	UINT32	rw	Y	UINT32	0
0x1600	6	N	pdrmap[5]	CiA 301	CANopen, RPdoMapping RPdoMapping6thApplicationObject	UINT32	rw	Y	UINT32	0
0x1600	7	N	pdrmap[6]	CiA 301	CANopen, RPdoMapping RPdoMapping7thApplicationObject	UINT32	rw	Y	UINT32	0
0x1600	8	N	pdrmap[7]	CiA 301	CANopen, RPdoMapping RPdoMapping8thApplicationObject	UINT32	rw	Y	UINT32	0
0x1601	0	N	pdrmapnum	CiA 301	CANopen RPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	2
0x1601	1…8	N	pdrmap[8…15]	CiA 301	CANopen RPdoMapping	UINT32	rw	Y	UINT32	…
0x1602	0	N	pdrmapnum	CiA 301	CANopen RPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	2
0x1602	1…8	N	pdrmap[16…23]	CiA 301	CANopen RPdoMapping	UINT32	rw	Y	UINT32	…
0x1603	0	N	pdrmapnum	CiA 301	CANopen RPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	3
0x1603	1…8	N	pdrmap[24…31]	CiA 301	CANopen RPdoMapping	UINT32	rw	Y	UINT32	…
0x1800	1	N	pdtcob[0]	CiA 301	CANopen, TPdo_CobIdUsedByPdo 1stTPdo_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	$NODEID + 0x0180
0x1800	2	N	pdttrn[0]	CiA 301	CANopen, TPdo_TransmissionType 1stTPdo_TransmissionType	UINT8	rw	Y	UINT8	255
0x1800	3	N	pdtinh[0]	CiA 301	CANopen, TPdo_InhibitTime 1stTPdo_InhibitTime	UINT16	rw	Y	UINT16	
0x1800	5	N	pdttim[0]	CiA 301	CANopen, TPdo_EventTimer 1stTPdo_EventTimer	UINT16	rw	Y	0…25000	
0x1801	1	N	pdtcob[1]	CiA 301	CANopen, TPdo_CobIdUsedByPdo 2ndTPdo_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	$NODEID + 0x0280

Table 124: Object dictionary (part 6 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x1801	2	N	pdttrn[1]	CiA 301	CANopen, TPdo_TransmissionType 2ndTPdo_TransmissionType	UINT8	rw	Y	UINT8	255
0x1801	3	N	pdtinh[1]	CiA 301	CANopen, TPdo_InhibitTime 2ndTPdo_InhibitTime	UINT16	rw	Y	UINT16	
0x1801	5	N	pdttim[1]	CiA 301	CANopen, TPdo_EventTimer 2ndTPdo_EventTimer	UINT16	rw	Y	0…25000	
0x1802	1	N	pdtcob[2]	CiA 301	CANopen, TPdo_CobIdUsedByPdo 3rdTPdo_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	$NODEID + 0x0380
0x1802	2	N	pdttrn[2]	CiA 301	CANopen, TPdo_TransmissionType 3rdTPdo_TransmissionType	UINT8	rw	Y	UINT8	255
0x1802	3	N	pdtinh[2]	CiA 301	CANopen, TPdo_InhibitTime 3rdTPdo_InhibitTime	UINT16	rw	Y	UINT16	
0x1802	5	N	pdttim[2]	CiA 301	CANopen, TPdo_EventTimer 3rdTPdo_EventTimer	UINT16	rw	Y	0…25000	
0x1803	1	N	pdtcob[3]	CiA 301	CANopen, TPdo_CobIdUsedByPdo 4thTPdo_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	$NODEID + 0x0480
0x1803	2	N	pdttrn[3]	CiA 301	CANopen, TPdo_TransmissionType 4thTPdo_TransmissionType	UINT8	rw	Y	UINT8	255
0x1803	3	N	pdtinh[3]	CiA 301	CANopen, TPdo_InhibitTime 4thTPdo_InhibitTime	UINT16	rw	Y	UINT16	
0x1803	5	N	pdttim[3]	CiA 301	CANopen, TPdo_EventTimer 4thTPdo_EventTimer	UINT16	rw	Y	0…25000	
0x1A00	0	N	pdtmapnum	CiA 301	CANopen TPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	1
0x1A00	1	N	pdtmap[0]	CiA 301	CANopen, PdoMapping PdoMapping1stApplicationObject	UINT32	rw	Y	UINT32	0x60410010
0x1A00	2	N	pdtmap[1]	CiA 301	CANopen, PdoMapping PdoMapping2ndApplicationObject	UINT32	rw	Y	UINT32	0
0x1A00	3	N	pdtmap[2]	CiA 301	CANopen, PdoMapping PdoMapping3rdApplicationObject	UINT32	rw	Y	UINT32	0
0x1A00	4	N	pdtmap[3]	CiA 301	CANopen, PdoMapping PdoMapping4thApplicationObject	UINT32	rw	Y	UINT32	0

Table 124: Object dictionary (part 7 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x1A00	5	N	pdtmap[4]	CiA 301	CANopen, PdoMapping PdoMapping5thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A00	6	N	pdtmap[5]	CiA 301	CANopen, PdoMapping PdoMapping6thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A00	7	N	pdtmap[6]	CiA 301	CANopen, PdoMapping PdoMapping7thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A00	8	N	pdtmap[7]	CiA 301	CANopen, PdoMapping PdoMapping8thApplicationObject	UINT32	rw	Y	UINT32	0
0x1A01	0	N	pdtmapnum	CiA 301	CANopen TPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	2
0x1A01	1…8	N	pdtmap[8…15]	CiA 301	CANopen PdoMapping	UINT32	rw	Y	UINT32	…
0x1A02	0	N	pdtmapnum	CiA 301	CANopen TPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	2
0x1A02	1…8	N	pdtmap[16…23]	CiA 301	CANopen PdoMapping	UINT32	rw	Y	UINT32	…
0x1A03	0	N	pdtmapnum	CiA 301	CANopen TPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	3
0x1A03	1…8	N	pdtmap[24…31]	CiA 301	CANopen PdoMapping	UINT32	rw	Y	UINT32	…
0x200F	0	N	pwrdly	Moog DCV	System PowerOnDelay	UINT8	rw	Y	0…10	0
0x2019	0	N	sfwver1	Moog DCV	System SoftwareVersion1	STRING(64)	ro	-		
0x201A	0	N	sfwver2	Moog DCV	System SoftwareVersion2	STRING(64)	ro	-		B99316-DV005-B-211F
0x201B	0	N	sfwver3	Moog DCV	System SoftwareVersion3	STRING(64)	ro	-		
0x2030	0	N	NULL_SRC	Moog DCV	ValveControllers SetpointSourceNull	INT16	ro	-	INT16	0
0x2031	0	N	NULL_SRCs32	Moog DCV	ValveControllers SetpointSourceNullS32	INT32	ro	-	INT32	0

Table 124: Object dictionary (part 8 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x2035	0	N	qsiz	Moog DCV	ValvePositionControl QSize	INT8	ro	-	INT8	
0x2036	0	N	prssiz	Moog DCV	ValvePressureControl PressureSize	INT8	ro	-	INT8	
0x2037	0	N	flwsiz	Moog DCV	ValveFlowControl FlowSize	INT8	ro	-	INT8	
0x2158	1	Y	stgsplctlout	Moog DCV	ValveMainStageControl MainStageControllerOutput	INT16	ro	-	INT16	
0x215B	1	Y	stgsplval	Moog DCV	ValveMainStageControl MainStagePositionActualValue	INT16	ro	-	INT16	None
0x215C	0	N	stgposcusogn	Moog DCV	ValveMainStageControl CustomerOverallGain	REAL32	rw	Y	0.00…2.00	1.00
0x215D	1	Y	stgsplctldvn	Moog DCV	ValveMainStageControl MainStagePositionControlDeviation	INT16	ro	-	INT16	
0x215E	1	Y	stgspldem	Moog DCV	ValveMainStageControl MainStageSplDemandValue	INT16	ro	-	INT16	
0x2171	0	Y	stgposint	Moog DCV	ValveMainStageControl MainStageIntegralPart	INT32	ro	-	INT32	
0x2303	1…16	N	cmpprsrmp[0…15]	Moog DCV	ValvePressureControl RampSlope	UINT16	rw	Y	UINT16	0
0x2304	1…16	N	cmpprspgn[0…15]	Moog DCV	ValvePressureControl ProportionalGain	REAL32	rw	Y	0.0…+inf	0.00
0x2305	1…16	N	cmpprsign[0…15]	Moog DCV	ValvePressureControl IntegratorGain	REAL32	rw	Y	0.0…+inf	0.00
0x2306	1…16	N	cmpprsigf[0…15]	Moog DCV	ValvePressureControl IntegratorFactor	REAL32	rw	Y	0.00…1.00	0.10
0x2307	1…16	N	cmpprsicr[0…15]	Moog DCV	ValvePressureControl IntegratorControlRange	INT16	rw	Y	0…32767	163
0x2308	1…16	N	cmpprsdgn[0…15]	Moog DCV	ValvePressureControl DifferentiatorGain	REAL32	rw	Y	REAL32	0.00
0x2309	1…16	N	cmpprsdtm[0…15]	Moog DCV	ValvePressureControl DifferentiatorT1	REAL32	rw	Y	0.0…+inf	0.00

Table 124: Object dictionary (part 9 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x230A	1…16	N	cmpprsupp[0…15]	Moog DCV	ValvePressureControl UpperOutputLimit	INT16	rw	Y	<LowerOutputLimit> (0x230B[1])…32767	16384
0x230B	1…16	N	cmpprslow[0…15]	Moog DCV	ValvePressureControl LowerOutputLimit	INT16	rw	Y	-32768…<UpperOutput-Limit> (0x230A[1])	-16384
0x230C	1…16	N	cmpprschy[0…15]	Moog DCV	ValvePressureControl HydraulicCapacity	REAL32	rw	Y	0.00…+inf	0.00
0x230D	1…16	N	cmpprsitf[0…15]	Moog DCV	ValvePressureControl ActiveTransducerInterface	INT8	rw	Y	1…4	1
0x230E	1…16	N	cmpprsptm[0…15]	Moog DCV	ValvePressureControl ProportionalGainTimeConstant	REAL32	rw	Y	0.0...+inf	0.00
0x230F	1…16	N	cmpprsitfb[0…15]	Moog DCV	ValvePressureControl TransducerInterfaceAreaB	INT8	rw	Y	0…4	None
0x2310	0	Y	cmpprsint	Moog DCV	ValvePressureControl IntegratorPart	REAL32	ro	-	REAL32	None
0x2311	0	Y	cmpprspro	Moog DCV	ValvePressureControl ProportionalPart	REAL32	ro	-	REAL32	None
0x2312	0	Y	cmpprsdt1	Moog DCV	ValvePressureControl DifferantialPart	REAL32	ro	-	REAL32	None
0x2313	0	N	cmpprsdir	Moog DCV	ValvePressureControl DirectionalDependentGain	REAL32	rw	Y	0.00…+inf	0.00
0x231A	1…16	N	cmpprsiup[0…15]	Moog DCV	ValvePressureControl IntegratorUpperOutputLimit	INT16	rw	Y	<IntegratorLowerOut-putLimit> (0x231B[1])…32767	16384
0x231B	1…16	N	cmpprsilo[0…15]	Moog DCV	ValvePressureControl IntegratorLowerOutputLimit	INT16	rw	Y	-32768…<IntegratorUp-perOutputLimit> (0x231A[1])	-16384
0x231C	1	N	prsref	Moog DCV	ValvePressureControl_SetpointConditioning Pressure_Reference	INT16	rw	N	0…32767	400
0x231C	2	N	prsrefuni	Moog DCV	ValvePressureControl_SetpointConditioning, Pressure_Reference PressureReference_Unit	UINT8	ro	-	UINT8	0x4E
0x231C	3	N	prsrefprf	Moog DCV	ValvePressureControl_SetpointConditioning, Pressure_Reference PressureReference_Prefix	INT8	ro	-	INT8	0

Table 124: Object dictionary (part 10 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x231D	0	N	cmpprsignsts	Moog DCV	ValvePressureControl IntegratorGainStatus	UINT8	ro	-	UINT8	None
0x2324	1…16	N	cmpprsfb-spgn[0…15]	Moog DCV	ValvePressureControl SpoolPositionFeedbackGain	REAL32	rw	Y	0.0…+inf	0.00
0x2330	0	N	cmpprsitfps	Moog DCV	ValvePressureControl TransducerInterfaceSystemPressure	INT8	rw	Y	0…4	0
0x233F	0	N	prschmmod	Moog DCV	ValvePressureControl PressureChamberModeSwitch	UINT8	rw	Y	0…1	
0x2344	1	N	prsvalabs	Moog DCV	ValvePressureControl PrsActualValueAbsolute	INT16	ro	-	INT16	
0x2350	0	N	prssetnum	Moog DCV	ValvePressureControl ParametersetSelection	UINT8	rw	Y	1…16	1
0x23D0	1	Y	qctlout	Moog DCV	ValveQControl Q_ControllerOutput	INT16	ro	-	INT16	
0x23E6	1	Y	swictlout	Moog DCV	ValvePQControl SwitchControllerOutput	INT16	ro	-	INT16	
0x23E7	0	N	swilimfac	Moog DCV	ValvePQControl LimitFactorForPressureAndFlowDemandValue	REAL32	ro	-	REAL32	0.00
0x23F0	1…4	N	prsfltbi[0…3]	Moog DCV	ValvePressureControl PressureValueFilterBCoeff	REAL32	rw	N	REAL32	0.00
0x23F1	1…4	N	prsfltai[0…3]	Moog DCV	ValvePressureControl PressureValueFilterACoeff	REAL32	rw	N	REAL32	0.00
0x23F2	0	N	cmpprsflt	Moog DCV	ValvePressureControl PressureActualValueFilterCutoffFrequency	REAL32	rw	Y	0.0…3333.3	0.00
0x23F3	0	N	prsfltord	Moog DCV	ValvePressureControl PressureActualValueFilterOrder	UINT8	rw	Y	0…3	0
0x2421	0	N	faisafupp	Moog DCV	ValveFailSafeWindowMonitoring UpperLimit	INT16	ro	-	<LowerLimit> (0x2422)…32767	16384
0x2422	0	N	faisaflow	Moog DCV	ValveFailSafeWindowMonitoring LowerLimit	INT16	ro	-	-32768…<UpperLimit> (0x2421)	-16384
0x242E	1	N	pltsplzrocor	Moog DCV	ValvePositionControl OffsetForPilotInDualStageMode	INT16	rw	Y	INT16	0

Table 124: Object dictionary (part 11 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x2804	0	Y	pwrsup	Moog DCV	Hardware_DiagnosticData PowerSupplyVoltage	UINT16	ro	-	UINT16	None
0x2805	0	Y	pcbtmp	Moog DCV	Temperature PcbTemperature	INT16	ro	-	INT16	None
0x280D	1	N	oprtim[0]	Moog DCV	HardwareDiagnosticData, OperationTimes CumulativePowerOnTime	UINT32	ro	-	UINT32	0
0x280D	2	N	oprtim[1]	Moog DCV	HardwareDiagnosticData, OperationTimes CumulativeValveActiveTime	UINT32	ro	-	UINT32	0
0x2822	1…5	N	errval[0…4]	Moog DCV	Error_Handler Internal_Error_Code	UINT32	ro	-	UINT32	0
0x2823	1…5	N	errtim[0…4]	Moog DCV	ErrorHandler ErrorTime	UINT32	ro	-	UINT32	0
0x2824	1…20	N	erraddinf[0…19]	Moog DCV	ErrorHandler ErrorAdditionalInformation	UINT32	ro	-	UINT32	
0x2827	0	N	pwrtim	Moog DCV	HardwareDiagnosticData TimeSinceLastPowerOn	UINT32	ro	-	UINT32	
0x2830	1…138	N	faurea[0…137]	Moog DCV	FaultReaction FaultReaction	UINT16	rw	Y	UINT16	…
0x2831	1…5	Y	fausts[0…4]	Moog DCV	FaultReaction FaultStatus	UINT32	ro	-	UINT32	None
0x2832	0	N	faudsc	Moog DCV	FaultReaction FaultReactionDescription	STRING(128)	ro	-	None	
0x2833	0	N	fauhis	Moog DCV	FaultReaction FaultHistoryNumber	UINT8	rw	N	0…8-1	None
0x2834	1…5	N	faustsret[0…4]	Moog DCV	FaultReaction FaultRetainStatus	UINT32	rw	N	UINT32	None
0x2855	1…5	N	pcbtmphstgrm[0…4]	Moog DCV	Hardware_DiagnosticData, TemperatureHistogram_TimeInMinutes	UINT32	ro	-	UINT32	None
0x2858	0	N	safmonena	Moog DCV	SafetyFunction SafetyFunctionMonitoringEnable	INT8	ro	-	0…1	0
0x2860	1	N	splvaldif	Moog DCV	ConditionMonitoring SpoolPosition_Difference	INT32	ro	-	INT32	

Table 124: Object dictionary (part 12 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x2861	0	N	splveltrs	Moog DCV	ConditionMonitoring Spool_Velocity_Threshold	REAL32	ro	-	REAL32	0.00
0x2862	1	N	spldstcum	Moog DCV	ConditionMonitoring Spool Distance Cumulative In Fullscale	REAL32	ro	-	REAL32	0.00
0x2862	4	N	spldstcumipu	Moog DCV	ConditionMonitoring, Spool Distance Cumulative In Fullscale Spool Distance Cumulative In Millimeter	REAL32	ro	-	REAL32	0.00
0x2862	5	N	spldstcumlsw	Moog DCV	ConditionMonitoring, Spool Distance Cumulative In Fullscale Spool Distance Cumulative Least Significant Word	UINT32	ro	-	UINT32	0
0x2862	6	N	spldstcummsw	Moog DCV	ConditionMonitoring, Spool Distance Cumulative In Fullscale Spool Distance Cumulative Most Significant Word	UINT32	ro	-	UINT32	0
0x2863	0	N	splvaldiftrs	Moog DCV	ConditionMonitoring SpoolPosition Difference Threshold	INT16	ro	-	INT16	
0x2900	0	N	errflg	Moog DCV	FaultReaction CustomerDefinedErrorFlag	INT8	rw	N	0…127	
0x2901	0	N	ev0exp	Moog DCV	Eventhandler Expression_0	STRING(192)	rw	Y		
0x2902	0	N	ev1exp	Moog DCV	Eventhandler Expression_1	STRING(192)	rw	Y		
0x2903	0	N	ev2exp	Moog DCV	Eventhandler Expression_2	STRING(192)	rw	Y		
0x2904	0	N	ev3exp	Moog DCV	Eventhandler Expression_3	STRING(192)	rw	Y		
0x2905	0	N	ev4exp	Moog DCV	Eventhandler Expression_4	STRING(192)	rw	Y		
0x2906	0	N	ev5exp	Moog DCV	Eventhandler Expression_5	STRING(192)	rw	Y		
0x2907	0	N	ev6exp	Moog DCV	Eventhandler Expression_6	STRING(192)	rw	Y		
0x2908	0	N	ev7exp	Moog DCV	Eventhandler Expression_7	STRING(192)	rw	Y		
0x2909	1…8	Y	evtena[0…7]	Moog DCV	Eventhandler Enable	INT8	rw	Y	0, 1	

Table 124: Object dictionary (part 13 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x290B	1…8	Y	vars08[0…7]	Moog DCV	Eventhandler, Integer08 Integer08_1…8	INT8	rw	N	INT8	None
0x290C	1…16	Y	vars16[0…15]	Moog DCV	Eventhandler, Integer16 Integer16_0…15	INT16	rw	N	INT16	None
0x290D	1…8	Y	vars32[0…7]	Moog DCV	Eventhandler, Integer32 Integer32_1…8	INT32	rw	N	INT32	None
0x290E	1…8	Y	varu08[0…7]	Moog DCV	Eventhandler, Unsigned08 Unsigned08_1…8	UINT8	rw	N	UINT8	None
0x290F	1…8	Y	varu16[0…7]	Moog DCV	Eventhandler, Unsigned16 Unsigned16_1…8	UINT16	rw	N	UINT16	None
0x2910	1…8	Y	varu32[0…7]	Moog DCV	Eventhandler, Unsigned32 Unsigned32_1…8	UINT32	rw	N	UINT32	None
0x2911	1…8	Y	varf32[0…7]	Moog DCV	Eventhandler, Float32 Float32_1…8	REAL32	rw	N	REAL32	None
0x3002	0	N	modide	Moog DCV	CAN ModuleIdentifier	UINT8	rw	Y	1…255	127
0x3003	0	N	canbdr	Moog DCV	CAN Bitrate	UINT32	rw	Y	20000L…1000000L	500000L
0x3004	0	N	nmtsta	Moog DCV	CAN NmtState	UINT16	ro	-	UINT16	None
0x3005	0	N	nmtctl	Moog DCV	CAN Nmt_SetState	UINT16	rw	N	1…130	None
0x300E	1	N	canhdwsts	Moog DCV	CAN CAN chip Error and Status Register	UINT32	ro	-	UINT32	0
0x300E	2	N	cantrxerrctr	Moog DCV	CAN, CAN chip Error and Status Register CAN chip Transmit Error Counter according CAN protocol spec Version 2.0	UINT32	ro	-	UINT32	0
0x300E	3	N	canrcverrctr	Moog DCV	CAN, CAN chip Error and Status Register CAN chip Receive Error Counter according CAN protocol spec Version 2.0	UINT32	ro	-	UINT32	0
0x3010	1	N	pdttrnman[0]	Moog DCV	CAN, TPdoManufacturerTransmissionType 1stTPdoManufacturerTransmissionType	UINT8	rw	Y	UINT8	

Table 124: Object dictionary (part 14 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x3010	2	N	pdttrnman[1]	Moog DCV	CAN, TPdoManufacturerTransmissionType 2ndTPdoManufacturerTransmissionType	UINT8	rw	Y	UINT8	
0x3010	3	N	pdttrnman[2]	Moog DCV	CAN, TPdoManufacturerTransmissionType 3rdTPdoManufacturerTransmissionType	UINT8	rw	Y	UINT8	
0x3010	4	N	pdttrnman[3]	Moog DCV	CAN, TPdoManufacturerTransmissionType 4thTPdoManufacturerTransmissionType	UINT8	rw	Y	UINT8	
0x3011	0	N	pdttrg	Moog DCV	CAN TPdoTrigger	UINT8	rw	N	0…4	
0x3012	1	N	pdrctr[0]	Moog DCV	CAN, WIDECAN_RPdoCounter 1stRPdoCounter	UINT32	rw	N	UINT32	
0x3012	2	N	pdrctr[1]	Moog DCV	CAN, WIDECAN_RPdoCounter 2ndRPdoCounter	UINT32	rw	N	UINT32	
0x3012	3	N	pdrctr[2]	Moog DCV	CAN, WIDECAN_RPdoCounter 3rdRPdoCounter	UINT32	rw	N	UINT32	
0x3012	4	N	pdrctr[3]	Moog DCV	CAN, WIDECAN_RPdoCounter 4thRPdoCounter	UINT32	rw	N	UINT32	
0x3030	0	N	lvdfrq	Moog DCV	Hardware BasicSampleFrequency	UINT16	ro	-	20000	20000
0x3100	0	Y	fcntyp	Moog DCV	FunctionGenerator, FunctionGenType FunctionGenType	INT8	rw	N	0…5	-
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

Table 124: Object dictionary (part 15 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
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
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

Table 124: Object dictionary (part 16 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
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	INT8	rw	Y	0…12	0
0x3204	0	Y	an0val	Moog DCV	AnalogInput0 AnaInpActualValue0	INT16	ro	-	INT16	None
0x3207	1	N	an0ref[0]	Moog DCV	AnalogInput0, AnaInpCustomerScaFactor0 AnaInpScaNumerator0	INT16	rw	Y	INT16	16384
0x3207	2	N	an0ref[1]	Moog DCV	AnalogInput0, AnaInpCustomerScaFactor0 AnaInpScaDenominator0	INT16	rw	Y	INT16	16384
0x3207	3	N	an0ref[2]	Moog DCV	AnalogInput0, AnaInpCustomerScaFactor0 AnaInpScaOffset0	INT16	rw	Y	INT16	0
0x3208	0	N	an1typ	Moog DCV	AnalogInput1 AnaInpType1	INT8	rw	Y	0…12	0
0x320C	0	Y	an1val	Moog DCV	AnalogInput1 AnaInpActualValue1	INT16	ro	-	INT16	None
0x320F	1	N	an1ref[0]	Moog DCV	AnalogInput1, AnaInpCustomerScaFactor1 AnaInpScaNumerator1	INT16	rw	Y	INT16	16384
0x320F	2	N	an1ref[1]	Moog DCV	AnalogInput1, AnaInpCustomerScaFactor1 AnaInpScaDenominator1	INT16	rw	Y	INT16	16384
0x320F	3	N	an1ref[2]	Moog DCV	AnalogInput1, AnaInpCustomerScaFactor1 AnaInpScaOffset1	INT16	rw	Y	INT16	0

Table 124: Object dictionary (part 17 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x3210	0	N	an2typ	Moog DCV	AnalogInput2 AnaInpType2	INT8	rw	Y	1…12	2
0x3214	0	Y	an2val	Moog DCV	AnalogInput2 AnaInpActualValue2	INT16	ro	-	INT16	None
0x3218	0	N	an3typ	Moog DCV	AnalogInput3 AnaInpType3	INT8	rw	Y	1…12	2
0x321C	0	Y	an3val	Moog DCV	AnalogInput3 AnaInpActualValue3	INT16	ro	-	INT16	None
0x321F	1	N	an3ref[0]	Moog DCV	AnalogInput3, AnaInpCustomerScaFactor3 AnaInpScaNumerator3	INT16	rw	Y	INT16	16384
0x321F	2	N	an3ref[1]	Moog DCV	AnalogInput3, AnaInpCustomerScaFactor3 AnaInpScaDenominator3	INT16	rw	Y	INT16	16384
0x321F	3	N	an3ref[2]	Moog DCV	AnalogInput3, AnaInpCustomerScaFactor3 AnaInpScaOffset3	INT16	rw	Y	INT16	0
0x3220	0	N	an4typ	Moog DCV	AnalogInput4 AnaInpType4	INT8	rw	Y	1…12	2
0x3224	0	Y	an4val	Moog DCV	AnalogInput4 AnaInpActualValue4	INT16	ro	-	INT16	None
0x3229	1	N	an4ref[0]	Moog DCV	AnalogInput4, AnaInpCustomerScaFactor4 AnaInpScaNumerator4	INT16	rw	Y	INT16	16384
0x3229	2	N	an4ref[1]	Moog DCV	AnalogInput4, AnaInpCustomerScaFactor4 AnaInpScaDenominator4	INT16	rw	Y	INT16	16384
0x3229	3	N	an4ref[2]	Moog DCV	AnalogInput4, AnaInpCustomerScaFactor4 AnaInpScaOffset4	INT16	rw	Y	INT16	0
0x322A	1	N	an2ref[0]	Moog DCV	AnalogInput2, AnaInpCustomerScaFactor2 AnaInpScaNumerator2	INT16	rw	Y	INT16	16384
0x322A	2	N	an2ref[1]	Moog DCV	AnalogInput2, AnaInpCustomerScaFactor2 AnaInpScaDenominator2	INT16	rw	Y	INT16	16384
0x322A	3	N	an2ref[2]	Moog DCV	AnalogInput2, AnaInpCustomerScaFactor2 AnaInpScaOffset2	INT16	rw	Y	INT16	0
0x3240	0	N	da0par	Moog DCV	AnalogOutput0 AnaOutMappingParameter0	UINT32	rw	Y	UINT32	0x63010110

Table 124: Object dictionary (part 18 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x3243	0	N	da0typ	Moog DCV	AnalogOutput0 Analog0OutputType	UINT8	rw	Y	0…1	0
0x3244	1	N	da0ref[0]	Moog DCV	AnalogOutput0, AnaOutCustomerScalingFactor0 AnaOutScaNumerator0	INT16	rw	Y	INT16	16384
0x3244	2	N	da0ref[1]	Moog DCV	AnalogOutput0, AnaOutCustomerScalingFactor0 AnaOutScaDenominator0	INT16	rw	Y	INT16	16384
0x3244	3	N	da0ref[2]	Moog DCV	AnalogOutput0, AnaOutCustomerScalingFactor0 AnaOutScaOffset0	INT16	rw	Y	INT16	0
0x3245	0	Y	da0val	Moog DCV	AnalogOutput0 AnaOutValue0	INT16	ro	-	Depending on <AnaOut-Type0> (0x3243)	None
0x3260	0	N	da1par	Moog DCV	AnalogOutput1 AnaOutMappingParameter1	UINT32	rw	Y	UINT32	0x63810110
0x3263	0	N	da1typ	Moog DCV	AnalogOutput1 AnaOutType1	UINT8	rw	Y	0…1	0
0x3264	0	N	vlvtrdpar	Moog DCV	ValveActualValueConditioning Port	UINT32	rw	N	UINT32	None
0x3265	1	N	da1ref[0]	Moog DCV	AnalogOutput1, AnaOutCustomerScalingFactor1 AnaOutScaNumerator1	INT16	rw	Y	INT16	16384
0x3265	2	N	da1ref[1]	Moog DCV	AnalogOutput1, AnaOutCustomerScalingFactor1 AnaOutScaDenominator1	INT16	rw	Y	INT16	16384
0x3265	3	N	da1ref[2]	Moog DCV	AnalogOutput1, AnaOutCustomerScalingFactor1 AnaOutScaOffset1	INT16	rw	Y	INT16	0
0x3266	0	Y	da1val	Moog DCV	AnalogOutput1 AnaOutValue1	INT16	ro	-	Depending on <AnaOut-Type1> (0x3263)	None
0x3270	0	N	vlvtrdstc	Moog DCV	ValveActualValueConditioning DataStructure	DOMAIN	rw	Y	None	None
0x32D6	1	N	mon24vbrntim[0]	Moog DCV	SupplyMonitoring, 24V_brn_hysteresis_time brn_hyst_time_bad	UINT32	rw	Y	0…25000	1
0x32D6	2	N	mon24vbrntim[1]	Moog DCV	SupplyMonitoring, 24V_brn_hysteresis_time brn_hyst_time_good	UINT32	rw	Y	0…25000	1
0x32E4	0	N	prsmdltyp	Moog DCV	PressureModule PressureSensorModuleTyp	INT8	rw	Y	0…0	0

Table 124: Object dictionary (part 19 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x32E5	0	N	prsmdlcormod	Moog DCV	PressureModule PressureSensorOffsetCorrectionMode	INT8	rw	Y	0…3	0
0x32E6	0	N	prsmdlcorexc	Moog DCV	PressureModule PressureSensorOffsetCorrectionExecution	INT8	rw	N	0…1	0
0x32E7	0	N	prsmdlcorupp	Moog DCV	PressureModule PressureSensorOffsetCorrectionBoundary	REAL32	ro	-	0.00…100.00	1.50
0x32E8	0	N	prsmdlcortrs	Moog DCV	PressureModule PressureSensorOffsetCorrectionTriggerThreshold	INT16	rw	Y	-32767…32767	1
0x32E9	0	N	prsmdlcorpar	Moog DCV	PressureModule PressureSensorOffsetCorrectionMappingParameter	UINT32	rw	Y	UINT32	0x32E60010
0x32F2	1	Y	prsmd0val	Moog DCV	PressureModule PressureSensorActualValue0	INT16	ro	-	INT16	None
0x32F3	1	Y	prsmd1val	Moog DCV	PressureModule PressureSensorActualValue1	INT16	ro	-	INT16	None
0x32F4	1	Y	prsmd2val	Moog DCV	PressureModule PressureSensorActualValue2	INT16	ro	-	INT16	None
0x32F8	0	N	prsmd0cor	Moog DCV	PressureModule PressureSensorCorrectionOffset0	INT16	ro	-	INT16	0
0x32F9	0	N	prsmd1cor	Moog DCV	PressureModule PressureSensorCorrectionOffset1	INT16	ro	-	INT16	0
0x32FA	0	N	prsmd2cor	Moog DCV	PressureModule PressureSensorCorrectionOffset2	INT16	ro	-	INT16	0
0x3300	1	Y	pltspldem	Moog DCV	ValvePositionControl PilotSplDemandValue	INT16	ro	-	INT16	
0x3301	1	Y	pltsplval	Moog DCV	ValvePositionControl PilotActualValue	INT16	ro	-	INT16	None
0x3302	1	Y	pltsplctldvn	Moog DCV	ValvePositionControl PilotSpoolControlDeviation	INT16	ro	-	INT16	
0x3303	1	Y	pltsplset	Moog DCV	ValvePositionControl PilotSpoolPositionSetpoint	INT16	ro	-	INT16	
0x3307	0	N	spgposmin	Moog DCV	HardwareDiagnosticData SpringPositionMinimum	INT16	ro	-	INT16	

Table 124: Object dictionary (part 20 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x3308	0	N	spgposmax	Moog DCV	HardwareDiagnosticData SpringPositionMaximum	INT16	ro	-	INT16	
0x330A	0	N	pltsplmontyp	Moog DCV	ValvePositionControl PilotSpoolControlMonitoring_Type	INT8	rw	Y	0…0x01	
0x330B	1	N	pltsplmontim	Moog DCV	ValvePositionControl PilotSpoolControlMonitoring_DelayTime	UINT16	rw	Y	UINT16	30
0x330B	2	N	timuni	Moog DCV	ValvePositionControl, PilotSpoolControlMonitoring_DelayTime Unit	UINT8	ro	-	UINT8	0x03
0x330B	3	N	timprf	Moog DCV	ValvePositionControl, PilotSpoolControlMonitoring_DelayTime Prefix	INT8	ro	-	INT8	-3
0x330D	1	N	pltsplmonupp	Moog DCV	ValvePositionControl PilotSpoolControlMonitoring_UpperThreshold	INT16	rw	Y	INT16	512
0x330D	2	N	spluni	Moog DCV	ValvePositionControl, PilotSpoolControlMonitoring_UpperThreshold Unit	UINT8	ro	-	UINT8	0x00
0x330D	3	N	splprf	Moog DCV	ValvePositionControl, PilotSpoolControlMonitoring_UpperThreshold Prefix	INT8	ro	-	INT8	0x00
0x330E	1	N	pltsplmonlow	Moog DCV	ValvePositionControl PilotSpoolControlMonitoring_LowerThreshold	INT16	rw	Y	INT16	-512
0x330E	2	N	spluni	Moog DCV	ValvePositionControl, PilotSpoolControlMonitoring_LowerThreshold Unit	UINT8	ro	-	UINT8	0x00
0x330E	3	N	splprf	Moog DCV	ValvePositionControl, PilotSpoolControlMonitoring_LowerThreshold Prefix	INT8	ro	-	INT8	0x00
0x3310	0	N	prspar	Moog DCV	ValvePressureControl PrsSetpointMappingParameter	UINT32	rw	Y	UINT32	0x63800110
0x3320	0	N	qpar	Moog DCV	ValvePositionControl QSetpointMappingParameter	UINT32	rw	Y	UINT32	0x63000110
0x332C	0	N	flwpar	Moog DCV	ValveFlowControl FlowSetpointMappingParameter	UINT32	rw	N	UINT32	0x67000110
0x3404	1	Y	prstrd	Moog DCV	PressureTransducer ActualValue	INT16	ro	-	INT16	
0x351A	0	Y	lvdval	Moog DCV	Hardware_Lvdt LvdtActualValue	INT16	ro	-	INT16	None

Table 124: Object dictionary (part 21 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x403F	0	N	ctllocdef	Moog DCV	Device LocalControlWordDefault	UINT16	rw	Y	UINT16	Device specific value
0x4040	0	N	ctlloc	Moog DCV	Device LocalControlWord	UINT16	rw	N	UINT16	<LocalControlWord Default> (0x403F)
0x4042	0	N	devmoddef	Moog DCV	Device DeviceModeDefault	INT8	rw	Y	1…2	Device specific value
0x4043	0	N	ctlmoddef	Moog DCV	Device ControlModeDefault	INT8	rw	Y	1…5, 14	Device specific value
0x4347	0	N	splchrtbl	Moog DCV	ActuatorConditioning, LookUpTableNegative LookUpTableNegative	DOMAIN	rw	Y	None	Device specific value
0x4348	1…128	N	splchrtbl[0…127]	Moog DCV	ActuatorConditioning, LookUpTableNegative LookUpTableNegative	INT16	rw	Y	INT16	Device specific value
0x4349	0	N	splchrtbl[128]	Moog DCV	ActuatorConditioning LookUpTableCenter	INT16	rw	Y	INT16	Device specific value
0x434A	1…128	N	splchrtbl[129…256]	Moog DCV	ActuatorConditioning, LookUpTable LookUpTablePositive	INT16	rw	Y	INT16	Device specific value
0x5005	0	N	locsncmsgcob	Moog DCV	LocalCAN LocalCAN_CobIdSyncMessage	UINT32	rw	Y	1…0x400007FF	<NodeID> (0x100B) + 0x0001
0x5006	0	N	loccomcycper	Moog DCV	LocalCAN LocalCAN_CommunicationCyclePeriod	UINT32	rw	Y	UINT32	
0x5007	0	N	locsncwinlen	Moog DCV	LocalCAN LocalCAN_SynchronousWindowLength	UINT32	rw	Y	UINT32	
0x500B	0	N	locnodide	Moog DCV	LocalCAN LocalCAN_NodeId	UINT32	ro	-	UINT32	127
0x500C	0	N	locgrdtim	Moog DCV	LocalCAN LocalCAN_GuardTime	UINT16	rw	Y	UINT16	
0x500D	0	N	locliftimfct	Moog DCV	LocalCAN LocalCAN_LifeTimeFactor	UINT8	rw	Y	UINT8	

Table 124: Object dictionary (part 22 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x500E	1	N	loccanhdwsts	Moog DCV	LocalCAN CAN_chip_Error_and_Status_Register	UINT32	ro	-	UINT32	0
0x500E	2	N	loccantrxerrctr	Moog DCV	LocalCAN, CAN_chip_Error_and_Status_Register CAN_chip_Transmit_Error_Counter_according_CAN_protocol_spec_Version_2_0	UINT32	ro	-	UINT32	0
0x500E	3	N	loccanrcverrctr	Moog DCV	LocalCAN, CAN_chip_Error_and_Status_Register CAN_chip_Receive_Error_Counter_according_CAN_protocol_spec_Version_2_0	UINT32	ro	-	UINT32	0
0x5012	0	N	loctspmsgcob	Moog DCV	LocalCAN LocalCAN_CobIdTimeStampMessage	UINT32	rw	Y	1…0x7FF	<NodeID> (0x100B) + 0x0081
0x5013	0	Y	lochghrsltsp	Moog DCV	LocalCAN LocalCAN_HighResolutionTimeStamp	UINT32	rw	Y	UINT32	
0x5014	0	N	locemgmsgcob	Moog DCV	LocalCAN LocalCAN_CobIdEmergencyMessage	UINT32	ro	-	0x81…0xFF	$NODEID + 0x0080
0x5015	0	N	locemgmsginh	Moog DCV	LocalCAN LocalCAN_InhibitTimeEmergencyMessage	UINT16	rw	Y	UINT16	
0x5016	1	N	locharbeacon	Moog DCV	LocalCAN LocalCAN_ConsumerHeartbeatTime	UINT32	rw	Y	UINT32	
0x5017	0	N	locharbeapro	Moog DCV	LocalCAN LocalCAN_ProducerHeartbeatTime	UINT16	rw	Y	UINT16	
0x5019	0	N	locsnccntmax	Moog DCV	LocalCAN LocalCAN_SynchronousCounterOverflowValue	UINT8	rw	Y	UINT8	
0x5029	1	N	loccomerrbeh	Moog DCV	LocalCAN LocalCAN_CommunicationErrorBehavior	UINT8	rw	Y	UINT8	
0x5102	0	N	locmodide	Moog DCV	LocalCAN LocalCANModuleIdentifier	UINT8	rw	Y	1…255	127
0x5103	0	N	loccanbdr	Moog DCV	LocalCAN LocalCANBitrate	UINT32	rw	Y	10000…1000000	500000
0x5104	0	N	locnmtsta	Moog DCV	LocalCAN LocalCAN_NmtState	UINT16	ro	-	UINT16	
0x5105	0	N	locnmtctl	Moog DCV	LocalCAN LocalCAN_Nmt_SetState	UINT16	rw	N	0…130	128

Table 124: Object dictionary (part 23 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x5110	1	N	locpdttrnman[0]	Moog DCV	LocalCAN, LocalCAN_TPdoManufacturerTransmissionType 1stLocTPdoManufacturerTransmissionType	UINT8	rw	Y	UINT8	
0x5110	2	N	locpdttrnman[1]	Moog DCV	LocalCAN, LocalCAN_TPdoManufacturerTransmissionType 2ndLocTPdoManufacturerTransmissionType	UINT8	rw	Y	UINT8	
0x5110	3	N	locpdttrnman[2]	Moog DCV	LocalCAN, LocalCAN_TPdoManufacturerTransmissionType 3rdLocTPdoManufacturerTransmissionType	UINT8	rw	Y	UINT8	
0x5110	4	N	locpdttrnman[3]	Moog DCV	LocalCAN, LocalCAN_TPdoManufacturerTransmissionType 4thLocTPdoManufacturerTransmissionType	UINT8	rw	Y	UINT8	
0x5111	0	N	locpdttrg	Moog DCV	LocalCAN LocalCANTPdoTrigger	UINT8	rw	N	0…4	None
0x5112	1	N	locpdrctr[0]	Moog DCV	LocalCAN, LocalCAN_LOCCAN_LocRPdoCounter 1stLocRPdoCounter	UINT32	rw	N	UINT32	
0x5112	2	N	locpdrctr[1]	Moog DCV	LocalCAN, LocalCAN_LOCCAN_LocRPdoCounter 2ndLocRPdoCounter	UINT32	rw	N	UINT32	
0x5112	3	N	locpdrctr[2]	Moog DCV	LocalCAN, LocalCAN_LOCCAN_LocRPdoCounter 3rdLocRPdoCounter	UINT32	rw	N	UINT32	
0x5112	4	N	locpdrctr[3]	Moog DCV	LocalCAN, LocalCAN_LOCCAN_LocRPdoCounter 4thLocRPdoCounter	UINT32	rw	N	UINT32	
0x5200	1	N	locsdoclncob	Moog DCV	LocalCAN LocalCAN_CobIDSDO	UINT32	ro	-	UINT32	$NODEID + 0x0600
0x5200	2	N	locsdosrvcob	Moog DCV	LocalCAN LocalCAN_CobIDSDO	UINT32	ro	-	UINT32	$NODEID + 0x0580
0x5203	0	N	prspsnom	Moog DCV	ValveFlowControl NominalSupplyPressure	INT16	rw	Y	1…32767	350
0x520F	1	Y	flwctlout	Moog DCV	ValveFlowControl FlowControllerOutput	INT16	ro	-	INT16	
0x5215	0	N	vlvqan	Moog DCV	ValveFlowControl RatedValveFlowASide	REAL32	rw	Y	0.01…+inf	14.43
0x5216	0	N	vlvqbn	Moog DCV	ValveFlowControl RatedValveFlowBSide	REAL32	rw	Y	0.01…+inf	14.59
0x5219	0	N	flwalpha	Moog DCV	ValvePressureControl CylinderAlpha	REAL32	ro	-	REAL32	0.00

Table 124: Object dictionary (part 24 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x521D	0	N	prssensmax	Moog DCV	ValveFlowControl PressureSensorMaximum	INT16	rw	Y	0…1000	350
0x521E	0	N	vlvdpn	Moog DCV	ValveFlowControl RatedValvePressureDrop	REAL32	rw	Y	1.00…+inf	34.50
0x521F	0	N	flwqamax	Moog DCV	ValveFlowControl MaximalFlowQAMax	REAL32	ro	-	REAL32	0.00
0x5400	1	N	locpdrcob[0]	Moog DCV	LocalCAN, LocalRPdo1_Configuration LocalRPdo1_CobIdUsedByPdo	UINT32	rw	Y	1...0x000007FF | 0x80000000	0x0200+127
0x5400	2	N	locpdrtrn[0]	Moog DCV	LocalCAN, LocalRPdo1_Configuration LocalRPdo1_TransmissionType	UINT8	rw	Y	UINT8	255
0x5400	5	N	locpdrtim[0]	Moog DCV	LocalCAN, LocalRPdo1_Configuration LocalRPdo1_EventTimer	UINT16	rw	Y	0…25000	0
0x5401	1	N	locpdrcob[1]	Moog DCV	LocalCAN, LocalRPdo2_Configuration LocalRPdo2_CobIdUsedByPdo	UINT32	rw	Y	1...0x000007FF | 0x80000000	0x0300+127
0x5401	2	N	locpdrtrn[1]	Moog DCV	LocalCAN, LocalRPdo2_Configuration LocalRPdo2_TransmissionType	UINT8	rw	Y	UINT8	255
0x5401	5	N	locpdrtim[1]	Moog DCV	LocalCAN, LocalRPdo2_Configuration LocalRPdo2_EventTimer	UINT16	rw	Y	0…25000	0
0x5402	1	N	locpdrcob[2]	Moog DCV	LocalCAN, LocalRPdo3_Configuration LocalRPdo3_CobIdUsedByPdo	UINT32	rw	Y	1...0x000007FF | 0x80000000	0x0400+127
0x5402	2	N	locpdrtrn[2]	Moog DCV	LocalCAN, LocalRPdo3_Configuration LocalRPdo3_TransmissionType	UINT8	rw	Y	UINT8	255
0x5402	5	N	locpdrtim[2]	Moog DCV	LocalCAN, LocalRPdo3_Configuration LocalRPdo3_EventTimer	UINT16	rw	Y	0…25000	0
0x5403	1	N	locpdrcob[3]	Moog DCV	LocalCAN, LocalRPdo4_Configuration LocalRPdo4_CobIdUsedByPdo	UINT32	rw	Y	1...0x000007FF | 0x80000000	0x0500+127
0x5403	2	N	locpdrtrn[3]	Moog DCV	LocalCAN, LocalRPdo4_Configuration LocalRPdo4_TransmissionType	UINT8	rw	Y	UINT8	255
0x5403	5	N	locpdrtim[3]	Moog DCV	LocalCAN, LocalRPdo4_Configuration LocalRPdo4_EventTimer	UINT16	rw	Y	0…25000	0
0x5600	0	N	locpdrmapnum	Moog DCV	LocalCAN, LocalRPdo1_Mapping LocalRPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	1

Table 124: Object dictionary (part 25 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x5600	1…8	N	locpdrmap[0…7]	Moog DCV	LocalCAN, LocalRPdo1_Mapping LocalRPdo1_ApplicPara1…8	UINT32	rw	Y	UINT32	…
0x5601	0	N	locpdrmapnum	Moog DCV	LocalCAN, LocalRPdo1_Mapping LocalRPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	2
0x5601	1…8	N	locpdrmap[8…15]	Moog DCV	LocalCAN, LocalRPdo2_Mapping LocalRPdo2_ApplicPara1…8	UINT32	rw	Y	UINT32	…
0x5602	0	N	locpdrmapnum	Moog DCV	LocalCAN, LocalRPdo1_Mapping LocalRPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	1
0x5602	1…8	N	locpdrmap[16…23]	Moog DCV	LocalCAN, LocalRPdo3_Mapping LocalRPdo3_ApplicPara1…8	UINT32	rw	Y	UINT32	…
0x5603	0	N	locpdrmapnum	Moog DCV	LocalCAN, LocalRPdo1_Mapping LocalRPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	2
0x5603	1…8	N	locpdrmap[24…31]	Moog DCV	LocalCAN, LocalRPdo4_Mapping LocalRPdo4_ApplicPara1…8	UINT32	rw	Y	UINT32	…
0x5800	1	N	locpdtcob[0]	Moog DCV	LocalCAN, LocalTPdo1_Configuration LocalTPdo1_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	0x0180+127
0x5800	2	N	locpdttrn[0]	Moog DCV	LocalCAN, LocalTPdo1_Configuration LocalTPdo1_TransmissionType	UINT8	rw	Y	UINT8	255
0x5800	5	N	locpdttim[0]	Moog DCV	LocalCAN, LocalTPdo1_Configuration LocalTPdo1_EventTimer	UINT16	rw	Y	0…25000	0
0x5801	1	N	locpdtcob[1]	Moog DCV	LocalCAN, LocalTPdo2_Configuration LocalTPdo2_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	0x0280+127
0x5801	2	N	locpdttrn[1]	Moog DCV	LocalCAN, LocalTPdo2_Configuration LocalTPdo2_TransmissionType	UINT8	rw	Y	UINT8	255
0x5801	5	N	locpdttim[1]	Moog DCV	LocalCAN, LocalTPdo2_Configuration LocalTPdo2_EventTimer	UINT16	rw	Y	0…25000	0
0x5802	1	N	locpdtcob[2]	Moog DCV	LocalCAN, LocalTPdo3_Configuration LocalTPdo3_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	0x0380+127
0x5802	2	N	locpdttrn[2]	Moog DCV	LocalCAN, LocalTPdo3_Configuration LocalTPdo3_TransmissionType	UINT8	rw	Y	UINT8	255
0x5802	5	N	locpdttim[2]	Moog DCV	LocalCAN, LocalTPdo3_Configuration LocalTPdo3_EventTimer	UINT16	rw	Y	0…25000	0

Table 124: Object dictionary (part 26 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x5803	1	N	locpdtcob[3]	Moog DCV	LocalCAN, LocalTPdo4_Configuration LocalTPdo4_CobIdUsedByPdo	UINT32	rw	Y	1…0x800007FF	0x0480+127
0x5803	2	N	locpdttrn[3]	Moog DCV	LocalCAN, LocalTPdo4_Configuration LocalTPdo4_TransmissionType	UINT8	rw	Y	UINT8	255
0x5803	5	N	locpdttim[3]	Moog DCV	LocalCAN, LocalTPdo4_Configuration LocalTPdo4_EventTimer	UINT16	rw	Y	0…25000	0
0x5857	1…16	N	cmppr-signswitrs[0…15]	Moog DCV	ValvePressureControl IntegratorGainSwitchThreshold	UINT32	rw	Y	UINT32	5000
0x5858	1…16	N	cmpprsfbsp-gn_2[0…15]	Moog DCV	ValvePressureControl SpoolPositionFeedbackGain_2	REAL32	rw	Y	0.0…+inf	0.00
0x585D	0	N	cmpfrcroda	Moog DCV	ValvePressureControl CylinderRodDiameterA	REAL32	rw	Y	0.0…<CylinderPistonDi-ameter> (0x585F)	0.00
0x585E	0	N	cmpfrcrodb	Moog DCV	ValvePressureControl CylinderRodDiameterB	REAL32	rw	Y	0.0…<CylinderPistonDi-ameter> (0x585F)	0.00
0x585F	0	N	cmpfrcpst	Moog DCV	ValvePressureControl CylinderPistonDiameter	REAL32	rw	Y	0.00…+inf	1000000.00
0x5860	1…2	Y	prsintprlvar[0…1]	Moog DCV	ValvePressureControl IntegralProportionalPart	INT16	ro	-	INT16	None
0x5861	1…16	N	prsintfbapgn[0…15]	Moog DCV	ValvePressureControl IntegralProportionalPartPGain	REAL32	rw	Y	REAL32	0.00
0x5862	0	Y	cmpprsdt1_2	Moog DCV	ValvePressureControl DifferantialPart_2	REAL32	ro	-	REAL32	None
0x5863	1…16	N	cmpprs-dgn_2[0…15]	Moog DCV	ValvePressureControl DifferentiatorGain_2	REAL32	rw	Y	REAL32	0.00
0x5864	1…16	N	cmpprsdt-m_2[0…15]	Moog DCV	ValvePressureControl DifferentiatorT1_2	REAL32	rw	Y	0.0…+inf	0.00
0x5865	1…16	N	prsoutupp[0…15]	Moog DCV	ValvePressureControl UpperControllerOutputLimit	INT16	rw	Y	<LowerOutputLimit> (0x5866[1])…32767	16384
0x5866	1…16	N	prsoutlow[0…15]	Moog DCV	ValvePressureControl LowerControllerOutputLimit	INT16	rw	Y	-32768…<UpperOutput-Limit> (0x5865[1])	-16384
0x5867	1…16	N	prsfwp[0…15]	Moog DCV	ValvePressureControl FeedForwardGain	REAL32	rw	Y	REAL32	0.00

Table 124: Object dictionary (part 27 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x5868	1…16	N	prsffwpar[0…15]	Moog DCV	ValvePressureControl FeedForwardParameter	UINT32	rw	Y	UINT32	0x63800110
0x5869	0	N	prsintprlpgn	Moog DCV	ValvePressureControl IntegratorPreloadGain	REAL32	rw	Y	REAL32	0.00
0x586A	0	N	prsintprlpar	Moog DCV	ValvePressureControl IntegratrPreloadParameter	UINT32	rw	Y	UINT32	0x63800110
0x586B	0	N	prsintprlmod	Moog DCV	ValvePressureControl IntegratorPreloadMode	UINT8	rw	Y	0…2	0
0x586C	1…16	N	splprsmod[0…15]	Moog DCV	ValvePressureControl pQSwitchingMode	UINT8	rw	Y	0…2	0
0x586D	0	N	prsdemsgnmod	Moog DCV	ValvePressureControl_SetpointConditioning PressureDemandSignMode	UINT8	rw	Y	0…1	0
0x5870	1…16	N	prsffwofs[0…15]	Moog DCV	ValvePressureControl FeedForwardOffset	INT16	rw	Y	INT16	0
0x5872	0	Y	prsctlout	Moog DCV	ValvePressureControl, PrsControllerOutput PressureControllerOutput	INT16	ro	-	INT16	
0x5873	1	Y	prsdemcnd	Moog DCV	ValvePressureControl_SetpointConditioning PrsDemandAfterSetpointConditioning	INT32	ro	-	INT32	
0x5A00	0	N	locpdtmapnum	Moog DCV	LocalCAN, LocalTPdo1_Mapping LocalTPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	1
0x5A00	1…8	N	locpdtmap[0…7]	Moog DCV	LocalCAN, LocalTPdo1_Mapping LocalTPdo1_ApplicPara1…8	UINT32	rw	Y	UINT32	…
0x5A01	0	N	locpdtmapnum	Moog DCV	LocalCAN, LocalTPdo1_Mapping LocalTPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	2
0x5A01	1…8	N	locpdtmap[8…15]	Moog DCV	LocalCAN, LocalTPdo2_Mapping LocalTPdo2_ApplicPara1…8	UINT32	rw	Y	UINT32	…
0x5A02	0	N	locpdtmapnum	Moog DCV	LocalCAN, LocalTPdo1_Mapping LocalTPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	1
0x5A02	1…8	N	locpdtmap[16…23]	Moog DCV	LocalCAN, LocalTPdo3_Mapping LocalTPdo3_ApplicPara1…8	UINT32	rw	Y	UINT32	…
0x5A03	0	N	locpdtmapnum	Moog DCV	LocalCAN, LocalTPdo1_Mapping LocalTPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…255	2

Table 124: Object dictionary (part 28 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x5A03	1…8	N	locpdtmap[24…31]	Moog DCV	LocalCAN, LocalTPdo4_Mapping LocalTPdo4_ApplicPara1…8	UINT32	rw	Y	UINT32	…
0x5B02	0	N	locsrn	Moog DCV	LocalCAN LocalCANStartRemoteNode	UINT8	rw	N	0…4	None
0x5B10	0	N	locrempar	Moog DCV	LocalCAN LocalCANRemoteParameter	UINT32	rw	N	UINT32	None
0x5B11	0	N	locremadr	Moog DCV	LocalCAN LocalCANRemoteParameterAdress	UINT32	rw	N	UINT32	None
0x5B12	0	N	locremnod	Moog DCV	LocalCAN LocalCANRemoteNodeId	UINT8	rw	N	0…127	None
0x5B13	0	N	locremtrn	Moog DCV	LocalCAN LocalCANRemoteTransmission	INT8	rw	N	-1…2	None
0x5B14	0	N	locbuster	Moog DCV	LocalCAN LocalCANTerminationResistor	INT8	rw	Y	0…1	0
0x5E41	1…3	N	digouttyp[0…2]	Moog DCV	DigitalOutput, DigitalOutputConfiguration DigitalOutputConfiguration0…2	UINT8	ro	-	0…4	…
0x5E42	1…3	Y	digout[0…2]	Moog DCV	DigitalOutput, DigitalOutputValue DigitalOutputValue0…2	INT8	rw	N	0…1	None
0x5E44	1…3	Y	digoutmon[0…2]	Moog DCV	DigitalOutput, DigitalOutputState DigitalOutputState0…2	INT8	ro	-	0…1	None
0x5E4A	0	N	vlvenamod	Moog DCV	DeviceEnable ValveEnableMode	INT8	ro	-	INT8	2
0x5E4B	0	Y	vlvenasts	Moog DCV	DeviceEnable ValveEnableSignalStatus	UINT8	ro	-	UINT8	None
0x6040	0	Y	ctlwrd	CiA 408	Device ControlWord	UINT16	rw	N	UINT16	None
0x6041	0	Y	stswrd	CiA 408	Device StatusWord	UINT16	ro	-	UINT16	None
0x6042	0	N	devmod	CiA 408	Device DeviceMode	INT8	rw	N	1…2	<Device-ModeDe-fault> (0x4042)

Table 124: Object dictionary (part 29 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6043	0	N	ctlmod	CiA 408	Device ControlMode	INT8	rw	N	1…5, 14	<ControlM-odeDefault> (0x4043)
0x604E	0	Y	errcod	CiA 408	Device DeviceErrorCode	UINT16	ro	-	UINT16	None
0x604F	0	N	locmod	CiA 408	Device Local	INT8	rw	Y	0…1	0
0x6050	0	N	devver	CiA 408	Device DeviceVersion	STRING(128)	ro	-	None	
0x6051	0	N	devcodnum	CiA 408	Device CodeNumber	UINT16	rw	Y	UINT16	0
0x6052	0	N	sernum	CiA 408	Device SerialNumber	STRING(64)	ro	-	None	
0x6053	0	N	devdsc	CiA 408	Device Description	STRING(64)	rw	Y	None	
0x6054	0	N	devmdldsc	CiA 408	Device ModelDescription	STRING(64)	ro	-	None	
0x6055	0	N	devmdlurl	CiA 408	Device ModelURL	STRING(64)	ro	-	None	www.moog.c om
0x6056	0	N	devprmcod	CiA 408	Device ParameterSetCode	UINT8	rw	Y	0…254	0
0x6057	0	N	devvennam	CiA 408	Device VendorName	STRING(64)S TRING(128)	ro	-	None	Moog
0x605F	0	N	devcap	CiA 408	Device Capability	UINT32	ro	-	16777216…	Device spe-cific value
0x6100	0	N	vlvtrdmax	CiA 408	ValveActualValueConditioning InterfaceNumberMax	UINT8	ro	-	UINT8	4
0x6101	0	N	vlvtrditf	CiA 408	ValveActualValueConditioning InterfaceNumber	UINT8	rw	N	1…4	None
0x6102	0	N	vlvtrdtyp	CiA 408	ValveActualValueConditioning Type	INT8	rw	N	-2, 0, 2, 5	None
0x6103	0	N	vlvtrdsgn	CiA 408	ValveActualValueConditioning Sign	INT8	rw	N	-1…1	None

Table 124: Object dictionary (part 30 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6104	1	Y	vlvtrdval	CiA 408	ValveActualValueConditioning ActualValue	INT16	ro	-	INT16	None
0x6110	1	Y	trditfval[0]	CiA 408	ValveActualValueConditioning, ActualValue ActualValue1	INT16	ro	-	INT16	None
0x6111	1	Y	trditfval[1]	CiA 408	ValveActualValueConditioning, ActualValue ActualValue2	INT16	ro	-	INT16	None
0x6112	1	Y	trditfval[2]	CiA 408	ValveActualValueConditioning, ActualValue ActualValue3	INT16	ro	-	INT16	None
0x6113	1	Y	trditfval[3]	CiA 408	ValveActualValueConditioning, ActualValue ActualValue4	INT16	ro	-	INT16	None
0x6120	1	N	trdprsmin	CiA 408	ValveActualValueConditioning PressureOutputMinimum	INT16	rw	N	INT16	None
0x6121	1	N	trdprsmax	CiA 408	ValveActualValueConditioning PressureOutputMaximum	INT16	rw	N	INT16	None
0x6122	1	N	trdprsare	CiA 408	ValveActualValueConditioning PressureArea	INT16	rw	N	INT16	None
0x6123	1	N	trdprsofs	CiA 408	ValveActualValueConditioning PressureOutputOffset	INT16	rw	N	INT16	None
0x6124	1	N	trdprssigmin	CiA 408	ValveActualValueConditioning PressureInputMinimum	INT16	rw	N	INT16	None
0x6125	1	N	trdprssigmax	CiA 408	ValveActualValueConditioning PressureInputMaximum	INT16	rw	N	INT16	None
0x6128	1	N	trdgenmin	CiA 408	ValveActualValueConditioning GeneralOutputMinimum	INT16	rw	N	INT16	None
0x6129	1	N	trdgenmax	CiA 408	ValveActualValueConditioning GeneralOutputMaximum	INT16	rw	N	INT16	None
0x612B	1	N	trdgenofs	CiA 408	ValveActualValueConditioning GeneralOutputOffset	INT16	rw	N	INT16	None
0x612C	1	N	trdgensigmin	CiA 408	ValveActualValueConditioning GeneralInputMinimum	INT32	rw	N	INT32	None
0x612D	1	N	trdgensigmax	CiA 408	ValveActualValueConditioning GeneralInputMaximum	INT32	rw	N	INT32	None

Table 124: Object dictionary (part 31 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6300	1	Y	qset	CiA 408	ValveQControl, QSetpoint QSetpoint	INT16	rw	N	INT16	None
0x6300	2	N	spluni	CiA 408	ValveQControl, QSetpoint Unit	UINT8	ro	-	UINT8	0
0x6300	3	N	splprf	CiA 408	ValveQControl, QSetpoint Prefix	INT8	ro	-	INT8	0
0x6301	1	Y	splval	CiA 408	ValvePositionControl, SplActualValue SplActualValue	INT16	ro	-	INT16	None
0x6301	2	N	spluni	CiA 408	ValvePositionControl, SpoolActualValue Unit	UINT8	ro	-	UINT8	0
0x6301	3	N	splprf	CiA 408	ValvePositionControl, SpoolActualValue Prefix	INT8	ro	-	INT8	0
0x6310	1	Y	qdem	CiA 408	ValveQControl, QDemandValue QDemandValue	INT16	ro	-	INT16	None
0x6310	2	N	spluni	CiA 408	ValveQControl, QDemandValue Unit	UINT8	ro	-	UINT8	0
0x6310	3	N	splprf	CiA 408	ValveQControl, QDemandValue Prefix	INT8	ro	-	INT8	0
0x6311	1	N	spldemref	CiA 408	ValvePositionControl, QDemandReferenceValue QDemandReferenceValue	INT16	ro	-	INT16	16384
0x6311	2	N	spluni	CiA 408	ValvePositionControl, QDemandReferenceValue Unit	UINT8	ro	-	UINT8	0
0x6311	3	N	splprf	CiA 408	ValvePositionControl, QDemandReferenceValue Prefix	INT8	ro	-	INT8	0
0x6312	1	Y	cmnsplset	CiA 408	ValvePositionControl, SpoolPositionSetpoint SpoolPositionSetpoint	INT16	ro	-	INT16	
0x6312	2	N	spluni	CiA 408	ValvePositionControl, SpoolPositionSetpoint Unit	UINT8	ro	-	UINT8	0
0x6312	3	N	splprf	CiA 408	ValvePositionControl, SpoolPositionSetpoint Prefix	INT8	ro	-	INT8	0
0x6313	1	Y	cmnspldem	CiA 408	ValvePositionControl, SpoolPositionDemandValue SpoolPositionDemandValue	INT16	ro	-	INT16	

Table 124: Object dictionary (part 32 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6313	2	N	spluni	CiA 408	ValvePositionControl, SpoolPositionDemandValue Unit	UINT8	ro	-	UINT8	0
0x6313	3	N	splprf	CiA 408	ValvePositionControl, SpoolPositionDemandValue Prefix	INT8	ro	-	INT8	0
0x6314	1	N	qsethld	CiA 408	ValveQControl, QHoldSetpoint QHoldSetPoint	INT16	rw	Y	INT16	None
0x6314	2	N	spluni	CiA 408	ValveQControl, QHoldSetpoint Unit	UINT8	ro	-	UINT8	0
0x6314	3	N	splprf	CiA 408	ValveQControl, QHoldSetpoint Prefix	INT8	ro	-	INT8	0
0x6320	1	N	spllimupp	CiA 408	ValveQControl_SetpointConditioning_Limit, UpperLimit UpperLimit	INT16	rw	Y	<LowerLimit> (0x6321)…32767	16384
0x6320	2	N	spluni	CiA 408	ValveQControl_SetpointConditioning_Limit, UpperLimit Unit	UINT8	ro	-	UINT8	0
0x6320	3	N	splprf	CiA 408	ValveQControl_SetpointConditioning_Limit, UpperLimit Prefix	INT8	ro	-	INT8	0
0x6321	1	N	spllimlow	CiA 408	ValveQControl_SetpointConditioning_Limit, LowerLimit LowerLimit	INT16	rw	Y	-32768…<UpperLimit> (0x6320)	-16384
0x6321	2	N	spluni	CiA 408	ValveQControl_SetpointConditioning_Limit, LowerLimit Unit	UINT8	ro	-	UINT8	0
0x6321	3	N	splprf	CiA 408	ValveQControl_SetpointConditioning_Limit, LowerLimit Prefix	INT8	ro	-	INT8	0
0x6322	0	N	spldemfct	CiA 408	ValveQControl_SetpointConditioning_Scaling Factor	UINT32	rw	Y	UINT32	0x00010001
0x6323	1	N	spldemofs	CiA 408	ValveQControl_SetpointConditioning_Scaling, Offset Offset	INT16	rw	Y	INT16	0
0x6323	2	N	spluni	CiA 408	ValveQControl_SetpointConditioning_Scaling, Offset Unit	UINT8	ro	-	UINT8	0
0x6323	3	N	splprf	CiA 408	ValveQControl_SetpointConditioning_Scaling, Offset Prefix	INT8	ro	-	INT8	0
0x6324	1	N	splzrocor	CiA 408	ActuatorConditioning, Offset Offset	INT16	rw	Y	INT16	0

Table 124: Object dictionary (part 33 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6324	2	N	spluni	CiA 408	ActuatorConditioning, Offset Unit	UINT8	ro	-	UINT8	0
0x6324	3	N	splprf	CiA 408	ActuatorConditioning, Offset Prefix	INT8	ro	-	INT8	0
0x6330	0	N	splrmptyp	CiA 408	ValveQControl_SetpointConditioning_Ramp Type	INT8	rw	Y	0…3	0
0x6331	1	N	splrmpacl	CiA 408	ValveQControl_SetpointConditioning_Ramp, AccelerationTime AccelerationTime	UINT16	rw	Y	UINT16	0
0x6331	2	N	timuni	CiA 408	ValveQControl_SetpointConditioning_Ramp, AccelerationTime Unit	UINT8	ro	-	UINT8	3
0x6331	3	N	splrmpaclprf	CiA 408	ValveQControl_SetpointConditioning_Ramp, AccelerationTime AccelerationTime_Prefix	INT8	rw	Y	-4…0	-3
0x6332	1	N	splrmpaclpos	CiA 408	ValveQControl_SetpointConditioning_Ramp, AccelerationTimePositive AccelerationTimePositive	UINT16	rw	Y	UINT16	0
0x6332	2	N	timuni	CiA 408	ValveQControl_SetpointConditioning_Ramp, AccelerationTimePositive Unit	UINT8	ro	-	UINT8	3
0x6332	3	N	splrmpaclposprf	CiA 408	ValveQControl_SetpointConditioning_Ramp, AccelerationTimePositive AccelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3
0x6333	1	N	splrmpaclneg	CiA 408	ValveQControl_SetpointConditioning_Ramp, AccelerationTimeNegative AccelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x6333	2	N	timuni	CiA 408	ValveQControl_SetpointConditioning_Ramp, AccelerationTimeNegative Unit	UINT8	ro	-	UINT8	3
0x6333	3	N	splrmpaclnegprf	CiA 408	ValveQControl_SetpointConditioning_Ramp, AccelerationTimeNegative AccelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3

Table 124: Object dictionary (part 34 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6334	1	N	splrmpdcl	CiA 408	ValveQControl_SetpointConditioning_Ramp, DecelerationTime DecelerationTime	UINT16	rw	Y	UINT16	0
0x6334	2	N	timuni	CiA 408	ValveQControl_SetpointConditioning_Ramp, DecelerationTime Unit	UINT8	ro	-	UINT8	3
0x6334	3	N	splrmpdclprf	CiA 408	ValveQControl_SetpointConditioning_Ramp, DecelerationTime DecelerationTime_Prefix	INT8	rw	Y	-4…0	-3
0x6335	1	N	splrmpdclpos	CiA 408	ValveQControl_SetpointConditioning_Ramp, DecelerationTimePositive DecelerationTimePositive	UINT16	rw	Y	UINT16	0
0x6335	2	N	timuni	CiA 408	ValveQControl_SetpointConditioning_Ramp, DecelerationTimePositive Unit	UINT8	ro	-	UINT8	3
0x6335	3	N	splrmpdclposprf	CiA 408	ValveQControl_SetpointConditioning_Ramp, DecelerationTimePositive DecelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3
0x6336	1	N	splrmpdclneg	CiA 408	ValveQControl_SetpointConditioning_Ramp, DecelerationTimeNegative DecelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x6336	2	N	timuni	CiA 408	ValveQControl_SetpointConditioning_Ramp, DecelerationTimeNegative Unit	UINT8	ro	-	UINT8	3
0x6336	3	N	splrmpdclnegprf	CiA 408	ValveQControl_SetpointConditioning_Ramp, DecelerationTimeNegative DecelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3
0x6340	0	N	spldirtyp	CiA 408	ActuatorConditioning Type	INT8	rw	Y	0…2	0
0x6341	0	N	spldirfct	CiA 408	ActuatorConditioning Factor	UINT32	rw	Y	UINT32	0x00010001
0x6342	0	N	spldbdtyp	CiA 408	ActuatorConditioning Type	INT8	rw	Y	0…2	0

Table 124: Object dictionary (part 35 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6343	1	N	spldbdsida	CiA 408	ActuatorConditioning, ASide ASide	INT16	rw	Y	0...16384	0
0x6343	2	N	spluni	CiA 408	ActuatorConditioning, ASide Unit	UINT8	ro	-	UINT8	0
0x6343	3	N	splprf	CiA 408	ActuatorConditioning, ASide Prefix	INT8	ro	-	INT8	0
0x6344	1	N	spldbdsidb	CiA 408	ActuatorConditioning, BSide BSide	INT16	rw	Y	0...16384	0
0x6344	2	N	spluni	CiA 408	ActuatorConditioning, BSide Unit	UINT8	ro	-	UINT8	0
0x6344	3	N	splprf	CiA 408	ActuatorConditioning, BSide Prefix	INT8	ro	-	INT8	0
0x6345	1	N	spldbdtrs	CiA 408	ActuatorConditioning, Threshold Threshold	INT16	rw	Y	0...16383	0
0x6345	2	N	spluni	CiA 408	ActuatorConditioning, Threshold Unit	UINT8	ro	-	UINT8	0
0x6345	3	N	splprf	CiA 408	ActuatorConditioning, Threshold Prefix	INT8	ro	-	INT8	0
0x6346	0	N	splchrtyp	CiA 408	ActuatorConditioning Type	INT8	rw	Y	-1…0	0
0x6350	1	Y	splctldvn	CiA 408	ValvePositionControl SpoolControlDeviation	INT16	ro	-	INT16	None
0x6350	2	N	spluni	CiA 408	ValvePositionControl, SpoolControlDeviation Unit	UINT8	ro	-	UINT8	0
0x6350	3	N	splprf	CiA 408	ValvePositionControl, SpoolControlDeviation Prefix	INT8	ro	-	INT8	0
0x6351	0	N	splmontyp	CiA 408	ValvePositionControl SplControlMonitoring_Type	INT8	rw	Y	0…0x01	0
0x6352	1	N	splmontim	CiA 408	ValvePositionControl SplControlMonitoring_DelayTime	UINT16	rw	Y	UINT16	30
0x6352	2	N	timuni	CiA 408	ValvePositionControl, SplControlMonitoring_DelayTime Unit	UINT8	ro	-	UINT8	3

Table 124: Object dictionary (part 36 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6352	3	N	timprf	CiA 408	ValvePositionControl, SplControlMonitoring_DelayTime Prefix	INT8	ro	-	INT8	-3
0x6354	1	N	splmonupp	CiA 408	ValvePositionControl SplControlMonitoring_UpperThreshold	INT16	rw	Y	INT16	512
0x6354	2	N	spluni	CiA 408	ValvePositionControl, SplControlMonitoring_UpperThreshold Unit	UINT8	ro	-	UINT8	0
0x6354	3	N	splprf	CiA 408	ValvePositionControl, SplControlMonitoring_UpperThreshold Prefix	INT8	ro	-	INT8	0
0x6355	1	N	splmonlow	CiA 408	ValvePositionControl SplControlMonitoring_LowerThreshold	INT16	rw	Y	INT16	-512
0x6355	2	N	spluni	CiA 408	ValvePositionControl, SplControlMonitoring_LowerThreshold Unit	UINT8	ro	-	UINT8	0
0x6355	3	N	splprf	CiA 408	ValvePositionControl, SplControlMonitoring_LowerThreshold Prefix	INT8	ro	-	INT8	0
0x6380	1	Y	prsset	CiA 408	ValvePressureControl, PrsSetpoint PrsSetpoint	INT16	rw	N	INT16	None
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	None
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, PrsDemandValue PrsDemandValue	INT16	ro	-	INT16	None
0x6390	2	N	prsuni	CiA 408	ValvePressureControl, PrsDemandValue Unit	UINT8	ro	-	UINT8	0
0x6390	3	N	prsprf	CiA 408	ValvePressureControl, PrsDemandValue Prefix	INT8	ro	-	INT8	0

Table 124: Object dictionary (part 37 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6391	1	N	prsdemref	CiA 408	ValvePressureControl_SetpointConditioning, PrsReferenceValue PrsReferenceValue	INT16	ro	-	INT16	16384
0x6391	2	N	prsuni	CiA 408	ValvePressureControl_SetpointConditioning, PrsReferenceValue Unit	UINT8	ro	-	UINT8	0
0x6391	3	N	prsprf	CiA 408	ValvePressureControl_SetpointConditioning, PrsReferenceValue Prefix	INT8	ro	-	INT8	0
0x6394	1	N	prssethld	CiA 408	ValvePressureControl_SetpointConditioning, PrsHoldSetpoint PrsHoldSetpoint	INT16	rw	Y	INT16	None
0x6394	2	N	prsuni	CiA 408	ValvePressureControl_SetpointConditioning, PrsSetpoint Unit	UINT8	ro	-	UINT8	0
0x6394	3	N	prsprf	CiA 408	ValvePressureControl_SetpointConditioning, PrsSetpoint Prefix	INT8	ro	-	INT8	0
0x63A0	1	N	prslimupp	CiA 408	ValvePressureControl_SetpointConditioning_Limit UpperLimit	INT16	rw	Y	<LowerLimit> (0x63A1)…32767	16384
0x63A0	2	N	prsuni	CiA 408	ValvePressureControl_SetpointConditioning_Limit, UpperLimit Unit	UINT8	ro	-	UINT8	0
0x63A0	3	N	prsprf	CiA 408	ValvePressureControl_SetpointConditioning_Limit, UpperLimit Prefix	INT8	ro	-	INT8	0
0x63A1	1	N	prslimlow	CiA 408	ValvePressureControl_SetpointConditioning_Limit LowerLimit	INT16	rw	Y	-32768…<UpperLimit> (0x63A0)	-16384
0x63A1	2	N	prsuni	CiA 408	ValvePressureControl_SetpointConditioning_Limit, LowerLimit Unit	UINT8	ro	-	UINT8	0
0x63A1	3	N	prsprf	CiA 408	ValvePressureControl_SetpointConditioning_Limit, LowerLimit Prefix	INT8	ro	-	INT8	0
0x63A2	0	N	prsdemfct	CiA 408	ValvePressureControl_SetpointConditioning_Scaling Factor	UINT32	rw	Y	UINT32	0x00010001
0x63A3	1	N	prsdemofs	CiA 408	ValvePressureControl_SetpointConditioning_Scaling Offset	INT16	rw	Y	INT16	0
0x63A3	2	N	prsuni	CiA 408	ValvePressureControl_SetpointConditioning_Scaling, Offset Unit	UINT8	ro	-	UINT8	0
0x63A3	3	N	prsprf	CiA 408	ValvePressureControl_SetpointConditioning_Scaling, Offset Prefix	INT8	ro	-	INT8	0

Table 124: Object dictionary (part 38 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x63B0	0	N	prsrmptyp	CiA 408	ValvePressureControl_SetpointConditioning_Ramp Type	INT8	rw	Y	0…0x03	0
0x63B1	1	N	prsrmpacl	CiA 408	ValvePressureControl_SetpointConditioning_Ramp AccelerationTime	UINT16	rw	Y	UINT16	0
0x63B1	2	N	timuni	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, AccelerationTime Unit	UINT8	ro	-	UINT8	3
0x63B1	3	N	prsrmpaclprf	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, AccelerationTime AccelerationTime_Prefix	INT8	rw	Y	-4…0	-3
0x63B2	1	N	prsrmpaclpos	CiA 408	ValvePressureControl_SetpointConditioning_Ramp AccelerationTimePositive	UINT16	rw	Y	UINT16	0
0x63B2	2	N	timuni	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, AccelerationTimePositive Unit	UINT8	ro	-	UINT8	3
0x63B2	3	N	prsrmpaclposprf	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, AccelerationTimePositive AccelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3
0x63B3	1	N	prsrmpaclneg	CiA 408	ValvePressureControl_SetpointConditioning_Ramp AccelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x63B3	2	N	timuni	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, AccelerationTimeNegative Unit	UINT8	ro	-	UINT8	3
0x63B3	3	N	prsrmpaclnegprf	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, AccelerationTimeNegative AccelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3
0x63B4	1	N	prsrmpdcl	CiA 408	ValvePressureControl_SetpointConditioning_Ramp DecelerationTime	UINT16	rw	Y	UINT16	0
0x63B4	2	N	timuni	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, DecelerationTime Unit	UINT8	ro	-	UINT8	3
0x63B4	3	N	prsrmpdclprf	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, DecelerationTime DecelerationTime_Prefix	INT8	rw	Y	-4…0	-3

Table 124: Object dictionary (part 39 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x63B5	1	N	prsrmpdclpos	CiA 408	ValvePressureControl_SetpointConditioning_Ramp DecelerationTimePositive	UINT16	rw	Y	UINT16	0
0x63B5	2	N	timuni	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, DecelerationTimePositive Unit	UINT8	ro	-	UINT8	3
0x63B5	3	N	prsrmpdclposprf	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, DecelerationTimePositive DecelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3
0x63B6	1	N	prsrmpdclneg	CiA 408	ValvePressureControl_SetpointConditioning_Ramp DecelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x63B6	2	N	timuni	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, DecelerationTimeNegative Unit	UINT8	ro	-	UINT8	3
0x63B6	3	N	prsrmpdclnegprf	CiA 408	ValvePressureControl_SetpointConditioning_Ramp, DecelerationTimeNegative DecelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3
0x63D0	1	Y	prsctldvn	CiA 408	ValvePressureControl 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	N	prsmontyp	CiA 408	ValvePressureControl_PrsControlMonitoring Type	INT8	rw	Y	0…0x01	0
0x63D2	1	N	prsmontim	CiA 408	ValvePressureControl_PrsControlMonitoring DelayTime	UINT16	rw	Y	UINT16	30
0x63D2	2	N	timuni	CiA 408	ValvePressureControl_PrsControlMonitoring, DelayTime Unit	UINT8	ro	-	UINT8	3
0x63D2	3	N	timprf	CiA 408	ValvePressureControl_PrsControlMonitoring, DelayTime Prefix	INT8	ro	-	INT8	-3
0x63D4	1	N	prsmonupp	CiA 408	ValvePressureControl_PrsControlMonitoring UpperThreshold	INT16	rw	Y	INT16	512

Table 124: Object dictionary (part 40 of 42)

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_PrsControlMonitoring, UpperThreshold Unit	UINT8	ro	-	UINT8	0
0x63D4	3	N	prsprf	CiA 408	ValvePressureControl_PrsControlMonitoring, UpperThreshold Prefix	INT8	ro	-	INT8	0
0x63D5	1	N	prsmonlow	CiA 408	ValvePressureControl_PrsControlMonitoring LowerThreshold	INT16	rw	Y	INT16	-512
0x63D5	2	N	prsuni	CiA 408	ValvePressureControl_PrsControlMonitoring, LowerThreshold Unit	UINT8	ro	-	UINT8	0
0x63D5	3	N	prsprf	CiA 408	ValvePressureControl_PrsControlMonitoring, LowerThreshold Prefix	INT8	ro	-	INT8	0
0x6700	1	Y	flwset	CiA 408	ValveFlowControl, FlowSetpoint FlowSetpoint	INT16	rw	N	INT16	None
0x6700	2	N	flwuni	CiA 408	ValveFlowControl, FlowSetpoint Unit	UINT8	ro	-	UINT8	0
0x6700	3	N	flwprf	CiA 408	ValveFlowControl, FlowSetpoint Prefix	INT8	ro	-	INT8	0
0x6701	1	Y	flwval	CiA 408	ValveFlowControl FlowActualValue	INT16	ro	-	INT16	
0x6701	2	N	flwprf	CiA 408	ValveFlowControl, FlowSetpoint Prefix	INT8	ro	-	INT8	0
0x6710	1	Y	flwdem	CiA 408	ValveFlowControl, FlowDemandValue FlowDemandValue	INT16	ro	-	INT16	
0x6710	2	N	flwuni	CiA 408	ValveFlowControl, FlowDemandValue Unit	UINT8	ro	-	UINT8	0
0x6710	3	N	flwprf	CiA 408	ValveFlowControl, FlowDemandValue Prefix	INT8	ro	-	INT8	0
0x6711	1	N	flwdemref	CiA 408	ValveFlowControl FlowDemandReferenceValue	INT16	rw	Y	INT16	16384
0x6711	2	N	flwuni	CiA 408	ValveFlowControl, FlowSetpoint Unit	UINT8	ro	-	UINT8	0
0x6711	3	N	flwprf	CiA 408	ValveFlowControl, FlowSetpoint Prefix	INT8	ro	-	INT8	0

Table 124: Object dictionary (part 41 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6714	1	N	flwsethld	CiA 408	ValveFlowControl FlowHoldSetpoint	INT16	rw	Y	-16384…16384	0
0x6714	2	N	flwuni	CiA 408	ValveFlowControl, FlowHoldSetpoint Unit	UINT8	ro	-	UINT8	0
0x6714	3	N	flwprf	CiA 408	ValveFlowControl, FlowHoldSetpoint Prefix	INT8	ro	-	INT8	0
0x6720	1	N	flwlimupp	CiA 408	ValveFlowControl_SetpointConditioning_Limit UpperLimit	INT16	rw	Y	<LowerLimit> (0x6721)…32767	16384
0x6720	2	N	flwuni	CiA 408	ValveFlowControl_SetpointConditioning_Limit, UpperLimit Unit	UINT8	ro	-	UINT8	0
0x6720	3	N	flwprf	CiA 408	ValveFlowControl_SetpointConditioning_Limit, UpperLimit Prefix	INT8	ro	-	INT8	0
0x6721	1	N	flwlimlow	CiA 408	ValveFlowControl_SetpointConditioning_Limit LowerLimit	INT16	rw	Y	-32768…<UpperLimit> (0x6720)	-16384
0x6721	2	N	flwuni	CiA 408	ValveFlowControl_SetpointConditioning_Limit, LowerLimit Unit	UINT8	ro	-	UINT8	0
0x6721	3	N	flwprf	CiA 408	ValveFlowControl_SetpointConditioning_Limit, LowerLimit Prefix	INT8	ro	-	INT8	0
0x6722	0	N	flwdemfct	CiA 408	ValveFlowControl_SetpointConditioning_Scaling FlowDemandFactor	UINT32	rw	Y	UINT32	0x00010001
0x6723	1	N	flwdemofs	CiA 408	ValveFlowControl_SetpointConditioning_Scaling FlowDemandOffset	INT16	rw	Y	INT16	
0x6723	2	N	flwuni	CiA 408	ValveFlowControl_SetpointConditioning_Scaling, FlowDemandOffset Unit	UINT8	ro	-	UINT8	0
0x6723	3	N	flwprf	CiA 408	ValveFlowControl_SetpointConditioning_Scaling, FlowDemandOffset Prefix	INT8	ro	-	INT8	0
0x6730	0	N	flwrmptyp	CiA 408	ValveFlowControl_SetpointConditioning_Ramp Type	INT8	rw	Y	0…0x03	
0x6731	1	N	flwrmpacl	CiA 408	ValveFlowControl_SetpointConditioning_Ramp FlowAccelerationTime	UINT16	rw	Y	UINT16	

Table 124: Object dictionary (part 42 of 42)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6731	2	N	flwrmpaclprf	CiA 408	ValveFlowControl_SetpointConditioning_Ramp, FlowAccelerationTime FlowAccelerationTime_Prefix	INT8	rw	Y	-4…0	-3
0x6732	1	N	flwrmpaclpos	CiA 408	ValveFlowControl_SetpointConditioning_Ramp FlowAccelerationTimePositive	UINT16	rw	Y	UINT16	
0x6732	2	N	flwrmpaclposprf	CiA 408	ValveFlowControl_SetpointConditioning_Ramp, FlowAccelerationTimePositive FlowAccelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3
0x6733	1	N	flwrmpaclneg	CiA 408	ValveFlowControl_SetpointConditioning_Ramp FlowAccelerationTimeNegative	UINT16	rw	Y	UINT16	
0x6733	2	N	flwrmpaclnegprf	CiA 408	ValveFlowControl_SetpointConditioning_Ramp, FlowAccelerationTimeNegative FlowAccelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3
0x6734	1	N	flwrmpdcl	CiA 408	ValveFlowControl_SetpointConditioning_Ramp FlowDecelerationTime	UINT16	rw	Y	UINT16	
0x6734	2	N	flwrmpdclprf	CiA 408	ValveFlowControl_SetpointConditioning_Ramp, FlowDecelerationTime FlowDecelerationTime_Prefix	INT8	rw	Y	-4…0	-3
0x6735	1	N	flwrmpdclpos	CiA 408	ValveFlowControl_SetpointConditioning_Ramp FlowDecelerationTimePositive	UINT16	rw	Y	UINT16	
0x6735	2	N	flwrmpdclposprf	CiA 408	ValveFlowControl_SetpointConditioning_Ramp, FlowDecelerationTimePositive FlowDecelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3
0x6736	1	N	flwrmpdclneg	CiA 408	ValveFlowControl_SetpointConditioning_Ramp FlowDecelerationTimeNegative	UINT16	rw	Y	UINT16	
0x6736	2	N	flwrmpdclnegprf	CiA 408	ValveFlowControl_SetpointConditioning_Ramp, FlowDecelerationTimeNegative FlowDecelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3
0x6750	1	Y	flwctldvn	CiA 408	ValveFlowControl FlowControlDeviationValue	INT16	ro	-	INT16	
0x6750	2	N	flwuni	CiA 408	ValveFlowControl, FlowControlDeviationValue Unit	UINT8	ro	-	UINT8	0

TAKE A CLOSER LOOK.

Moog designs a range of products that complement the performance of those featured in this user manual. Visit our Web site for more information and the Moog facility nearest you.

Australia
+61 3 9561 6044
info.australia@moog.com

Brazil
+55 11 3572 0400
info.brazil@moog.com

Canada
+1 716 652 2000
info.canada@moog.com

China
+86 21 2893 1600
info.china@moog.com

Finland
+358 10 422 1840
info.finland@moog.com

France
+33 1 4560 7000
info.france@moog.com

Germany
+49 7031 622 0
info.germany@moog.com

Hong Kong
+852 2 635 3200
info.hongkong@moog.com

India
+91 80 4057 6666
info.india@moog.com

Ireland
+353 21 451 9000
info.ireland@moog.com

Italy
+39 0332 421 111
info.italy@moog.com

Japan
+81 46 355 3767
info.japan@moog.com

Korea
+82 31 764 6711
info.korea@moog.com

Luxembourg
+352 40 46 401
info.luxembourg@moog.com

The Netherlands
+31 252 462 000
info.thenetherlands@moog.com

Norway
+47 6494 1948
info.norway@moog.com

Russia
+7 8 31 713 1811
info.russia@moog.com

Singapore
+65 677 36238
info.singapore@moog.com

South Africa
+27 12 653 6768
info.southafrica@moog.com

Spain
+34 902 133 240
info.spain@moog.com

Turkey
+90 216 663 6020
info.turkey@moog.com

United Kingdom
+44 (0) 1684 858000
info.uk@moog.com

USA
+1 716 652 2000
info.usa@moog.com

www.moog.com/industrial

Moog Literature Name
DIGITAL CONTROL VALVES WITH CANOPEN INTERFACE - Firmware
Rev. B, 08.12.21, Id. CDS60745-en

WHAT MOVES YOUR WORLD

MOOG