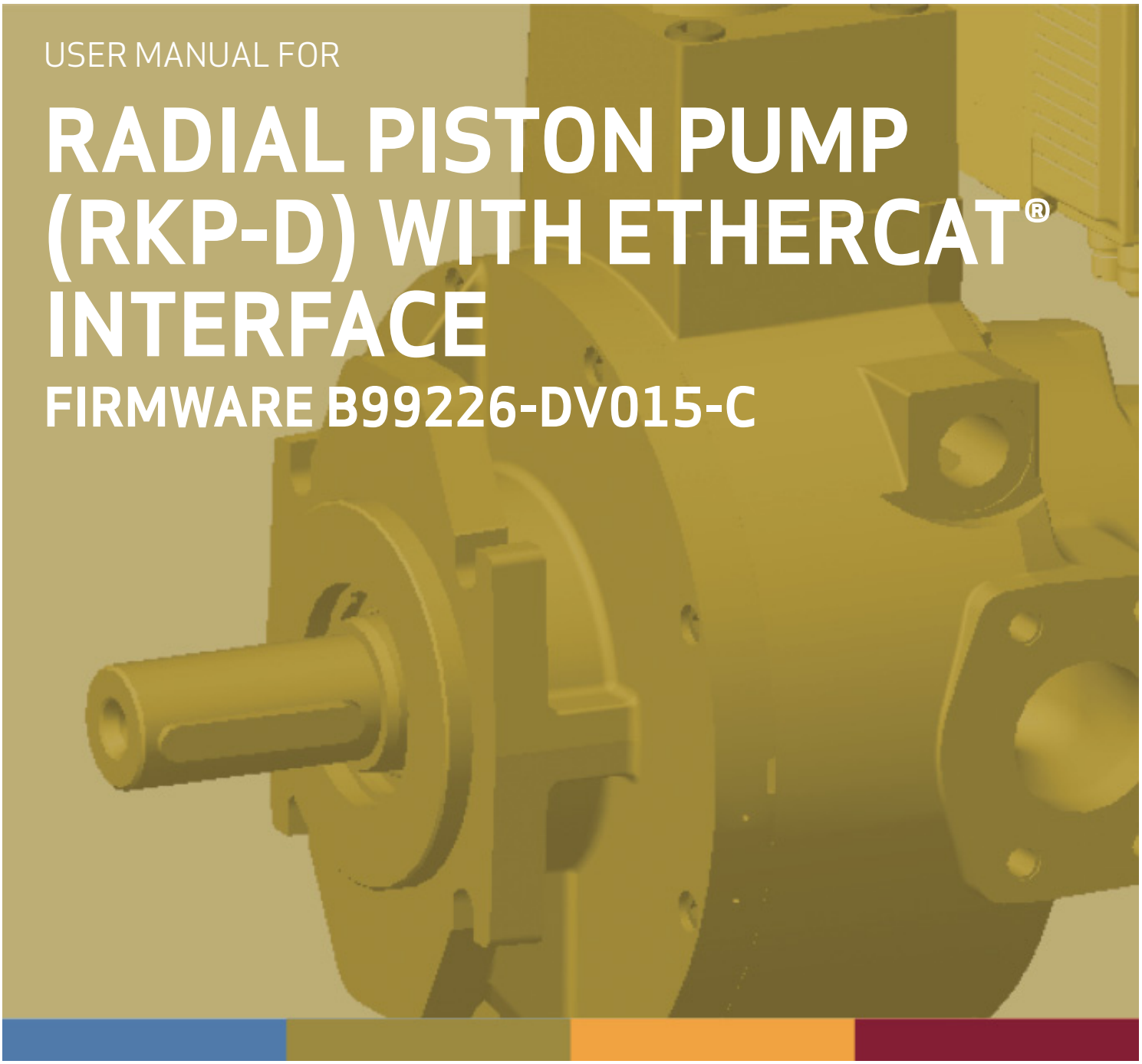
USER MANUAL FOR

RADIAL PISTON PUMP (RKP-D) WITH ETHERCAT® INTERFACE

FIRMWARE B99226-DV015-C



Rev. -, October 2013

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

Copyright

Hanns-Klemm-Straße 28
71034 Boeblingen
Germany
Telephone: +49 7031 622-0
Fax: +49 7031 622-191
E-mail: sales@moog.com
Internet: http://www.moog.com/Industrial

Table of contents

List of tables

List of figures

For your notes.

1 General information

1.1 About this manual

This document describes the EtherCAT® field bus interface of the Moog servo valve on the radial piston pump. If the Moog servo valve is mentioned in this document then it means the servo valve together with the pump.

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

All parameters follow the common EtherCAT communication profile IEC 61158-x-12 and the device specific CANopen profile "Device profile fluid power technology proportional valves and hydrostatic transmissions", CiA 408 released by the CAN in Automation (CiA) organisation.

This manual is part of the set of documentation available for the servo valve.
⇨ Chapter "1.3 Further documentation for the servo valve", page 3

This document is not a replacement for the CANopen standards as listed in the references.
⇨ Chapter "1.4 References", page 4

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

1.1.1 Reservation of changes and validity

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

1.1.2 Completeness

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

1.1.3 Place of storage

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

1.1.4 Warranty and liability

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

1.1.5 Typographical conventions

DANGER

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

WARNING

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

CAUTION

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

1.3 Further documentation for the servo valve

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

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

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

1.4 References

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

ETG	EtherCAT Technology Group Ostendstraße 196 DE-90482 Nuremberg http://www.ethercat.org
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 Nuremberg http://www.can-cia.org
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V. Lyoner Strasse 18 60528 Frankfurt/Main http://www.vdma.org

1.4.1 EtherCAT® field bus

The EtherCAT® field bus interface provides a 100Base-TX full duplex Real Time Ethernet connection to the servo valves using standard EtherCAT frames according to IEEE 802.3.

ISO/IEC 8802.3	Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications
IEC 61784 Part 2	Additional profiles for ISO/IEC 8802.3 based communication networks in real-time applications
IEC 61784 Part 5 Appendix A	Installing profiles for communication networks
IEC 61076-2-101, Amendment 1	M12 Connector
IEC 61158-3-12	EtherCAT Data-link service definition
IEC 61158-4-12	EtherCAT Data-link protocol specification
IEC 61158-5-12	EtherCAT Application layer service definition
IEC 61158-6-12	EtherCAT Application layer protocol specification
CiA 102	Physical layer for industrial applications

1.4.2 Device Profile

VDMA Profile Fluid Power	Device profile for Proportional Valves and Hydrostatic Transmissions VDMA Profile Fluid Power Technology Version 1.5
CiA 408 or Device Profile Fluid Power	CiA 408 Device profile for fluid power technology proportional valves and hydrostatic transmissions, Version 1.5e

1.5 Definitions

1.5.1 Internal resolution (iR)

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

1.5.2 Volume flow direction

A positive stroke ring demand value will result in a volume flow from connection A to the connection B of the pump.

1.5.3 Position and spool position

Position and spool position always refer to the stroke ring position. Other positions are named explicitly.

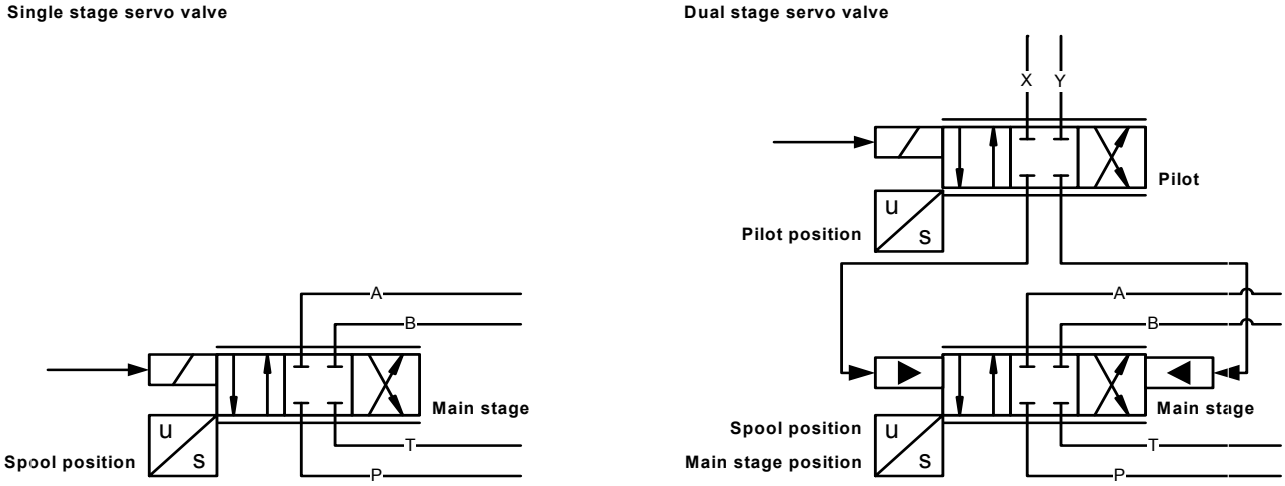


Figure 1: Servo valve position and stage names

1.6 Abbreviations

Abbreviation	Explanation
AC	Alternating Current
CAN	Controller Area Network
CANopen	ISO/OSI Layer 7 protocol, specified by CAN in Automation (CiA)
CiA	CAN in Automation
COB-ID	Communication Object Identifier
CoE	CANopen over EtherCAT protocol
CSMA/CD	Carrier sense multiple access with collision detection
DC	Direct Current
DCV	Moog Digital Control Valve
DSM	Device State Machine
DSP	Digital Signal Processor
DSV	Device specific value
EDS	Electronics Datasheet, containing a description of the CANopen object dictionary
EEPROM	Electrically erasable programmable read-only memory
EMCY	CANopen Emergency protocol

Table 1: Abbreviations (part 1 of 3)

Abbreviation	Explanation
EoE	Ethernet over EtherCAT
ESC	EtherCAT Slave Controller
ESI	EtherCAT Slave Information / EtherCAT XML file
ESM	EtherCAT network state machine / Application layer status machine
ETG	EtherCAT Technology Group
EtherCAT	Ethernet for Control and Automation Technology
FMMU	Field bus memory management unit
FoE	File over EtherCAT protocol
FPRD	Configured Address Physical Read
FPRW	Configured Address Physical Read Write
FPWR	Configured Address Physical Write
FRMW	Configured Address Physical Read Multiple Write
I	Integral element
IEEE	Institute of Electrical and Electronics Engineers
Ipv4	Internet Protocol Version 4
Ipv6	Internet Protocol Version 6
iR	Internal resolution defined by CiA 408
ISO	International Engineering Consortium
LED	Light Emitting Diode
LRD	Logical Read
LRW	Logical Read Write
LVDT	Linear Variable Differential Transformer used to measure the valves spool position
LWR	Logical Write
NMT	Network management according CANopen
NS	Network Status
OD	Object Dictionary
OSI	Open Systems Interconnection
P	Proportional gain element
PD	Proportional derivative element
PDO	Process Data Object
PE	Protective earth / Electrical grounding
Phy	Physical Layer
PPT1	Proportional first order lag element
RKP-D	Radial Piston Pump - Digital
ro	Read only
rw	Read write
RxPDO	Receive Process Data Object
RxPDO remote	Receive Process Data Object remote
RxSDO	Receive Service Data Object
SDO	Service Data Object
SDO Info	Service Data Object Information
SM	Synchronisation Manager
SoE	Servo profile over EtherCAT
TCP	Transmission Control Protocol
TR	State transmission of the valve application state machine
TxPDO	Transmit Process Data Object
TxPDO remote	Transmit Process Data Object remote

Table 1: Abbreviations (part 2 of 3)

Abbreviation	Explanation
TxSDO	Transmit Service Data Object
UDP	User Datagram Protocol
URL	Uniform Resource Locator / Internet address
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V.
WD	Watchdog
WKC or WC	Working Counter
wo	Write only
Xn	Physical connector n for electrical connection

Table 1: Abbreviations (part 3 of 3)

1.7 Trademarks

Moog and Moog Authentic Repair® are registered trademarks of Moog Inc. and its subsidiaries.
EtherCAT® is registered trademark and patented technology licensed by Beckhoff Automation GmbH, Germany.

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.

For your notes.

2 Access over EtherCAT®

2.1 Introduction

EtherCAT (Ethernet for Control and Automation Technology) is a real-time Ethernet technology and is particularly suited to communication between control systems and peripheral devices like I/O systems, drives, sensors and actuators. EtherCAT was developed in 2003 by Beckhoff Automation GmbH and propagated as an open standard. To further develop the technology, the user association "EtherCAT Technology Group" (ETG) was established.

With the EtherCAT technology, the non-real-time capability of other Ethernet solutions is overcome. The Ethernet packet is no longer received, then interpreted before the process data will be sent to the next slaves. The slave reads the data addressed to it, while the telegram passes through the device. Similarly, input data is inserted while the telegram passes through. During this process, the telegrams are only delayed by a few nanoseconds. The last slave in the segment sends the already completely processed telegram back to the first slave. This then sends the telegram back to the network master control as a reply telegram. This results in a logical ring structure for communication. As Fast Ethernet works with full duplex, this results in a ring structure also physically.

2.2 Device profiles

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

EtherCAT has inherited a lot from CANopen in its protocol definitions. Thus, the entire protocol for configuration and analysis of the servo valve (SDO and emergency) is identical to CiA 408. Thus there is no need for a special EtherCAT device profile for servo valves.

2.3 EtherCAT® slave reference model

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

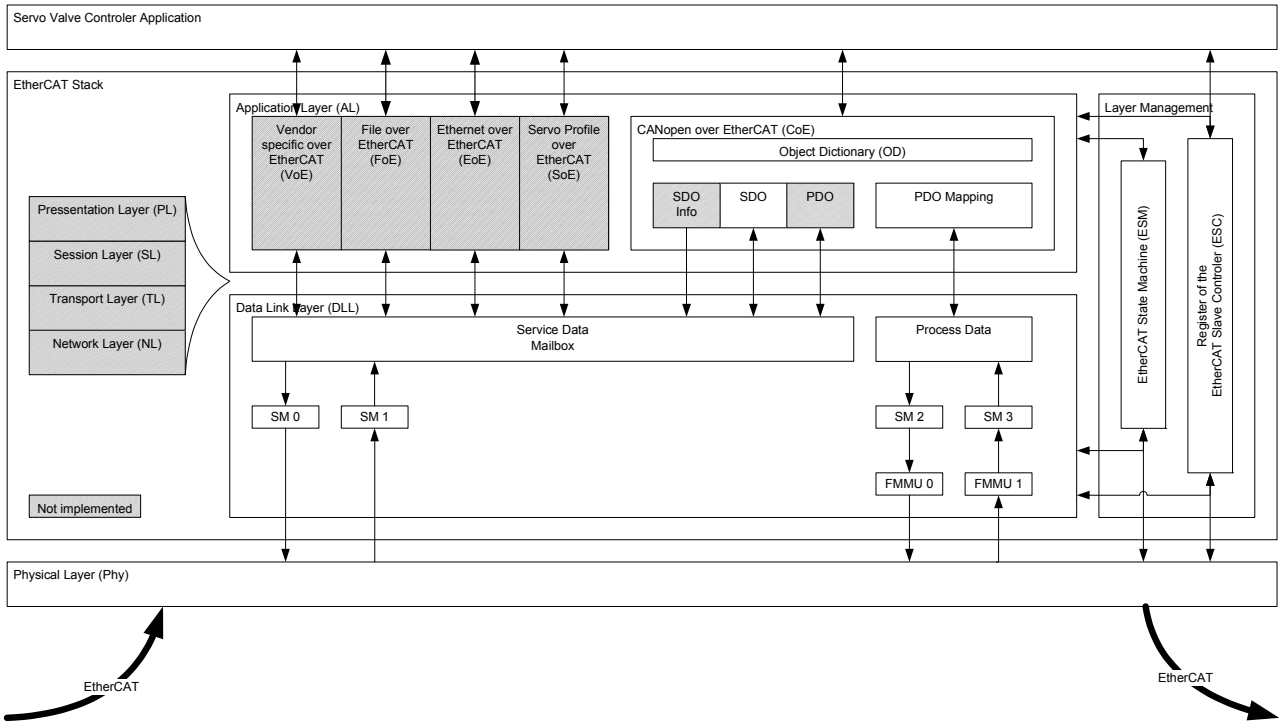


Figure 2: EtherCAT field bus communication layers

2.4 EtherCAT® protocol

The process data optimized EtherCAT protocol will be transported directly in a standard Ethernet frame. The Ethernet frame has a header and data. In the header it is defined which kind of data in the Ethernet frame are transported. The Ethernet frame can contain several types of sub protocols e.g. Ipv4, Ipv6, ARP, EtherCAT, etc. Each protocol has its own ethertype. The ethertype of the EtherCAT protocol is 0x88A4. In the absence of real-time capability, the data may be transferred over the ordinary UDP e.g. when using standard internet as communication channel.

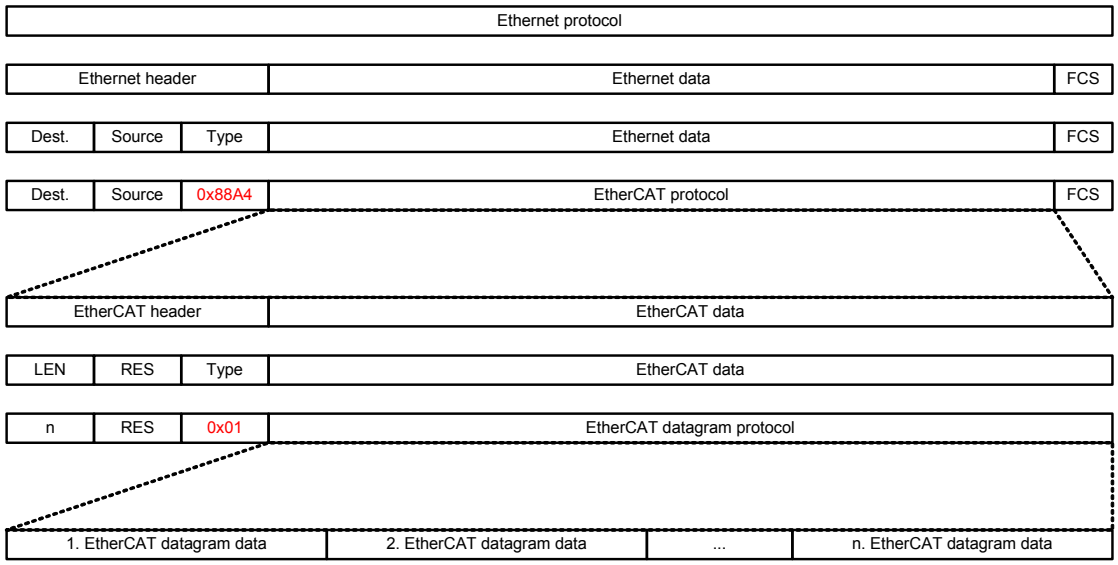


Figure 3: EtherCAT protocol

Only the EtherCAT sub-protocol for device communication (EtherCAT Type 0x01) is used for the communication between master and slaves.

2.5 Datagram protocol

Different read and write commands can be executed with the datagram protocol. According to the IEC 61158-4-12, EtherCAT also supports commands that enable reading and writing at the same time with just one telegram. While the incoming telegram passes through the EtherCAT slave controller (ESC) bit by bit, the input data addressed to the slave are read and subsequently new output data are written to the telegram passing through.

A register means a part of the EtherCAT slave controller (ESC) memory that is defined as interface memory between master and slave.

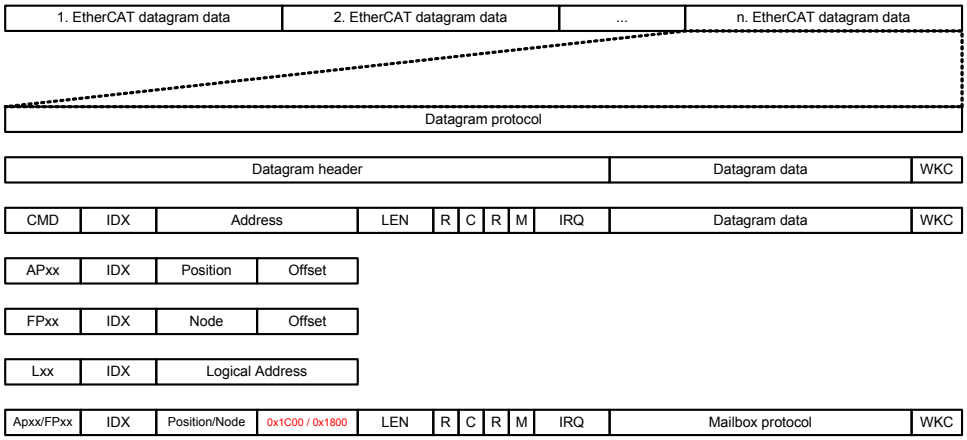


Figure 4: Datagram protocol

For addressing the slave, four different addressing modes are available. One logical addressing and three device-related addressings. For logical addressing, data of several devices can be read and written with one telegram. This addressing mode is used for real-time communication. For device-related addressing, a position-related address (auto increment address) or a configurable address (node address) can be used. The configurable address can either be allocated by the network master itself (configured station address) or manually via the network master (configured station alias).

The configured station alias node address is not supported by the servo valve hardware (ESC10 chip).

2.5.1 Device position address (auto increment address)

The datagram holds the position address of the addressed slave as a negative value. Each slave increments the address. The slave which reads the address equal zero is addressed and will execute the appropriate command at receive. Position addressing should only be used during start up of the EtherCAT system to scan the field bus and later only occasionally to detect newly attached slaves. Using position addressing is problematic if loops are closed temporarily due to link problems. Position addresses are shifted in this case and e.g. a mapping of error register values to devices becomes impossible, thus the faulty link cannot be localized

2.5.2 Node address (configured station address)

The configured station address is assigned by the master during start up (assigning with auto increment address commands) and cannot be changed by the EtherCAT slave. The configured station address is stored in the ESC register 0x0010:0x0011 (Configured Station Address).

2.5.3 Node address (configured station alias)

The configured station alias node address is not supported by the servo valve hardware (ESC10 chip).

2.5.4 Broadcast addressing

Each EtherCAT slave is addressed. Broadcast addressing is used e.g. for initialization of all slaves and for checking the status of all slaves if they are expected to be identical.

2.5.5 Logical addressing

All devices read from and write to the same logical 4 GByte address space (32 bit address field within the EtherCAT datagram). A slave uses a mapping unit field bus memory management unit (FMMU) to map data from the logical process data image to its local address space. During start up the master configures the FMMUs of each slave. The slave knows which parts of the logical process data image have to be mapped to which local address space using the configuration information of the FMMUs.

2.5.6 Datagram commands

These commands are used to transmit the process data object (PDO) and mailbox messages. The master node configuration specifies which commands are used.

Command	Description
Auto Increment Physical Read (APRD) (0x01)	Physical Addressing Each slave node is addressed via its physical position within the network segment. ⇨ Chapter "2.5 Datagram protocol", page 11
Auto Increment Physical Write (APWR) (0x02)	
Auto Increment Physical Read Write (APRW) (0x03)	
Auto Increment Physical Read Multiple Write (ARMW) (0x0D)	
Configured Address Physical Read (FPRD) (0x04)	Configured Addressing Each slave node is addressed via a configured node address within the network segment. ⇨ Chapter "2.5.2 Node address (configured station address)", page 12
Configured Address Physical Write (FPWR) (0x05)	
Configured Address Physical ReadWrite (FPRW) (0x06)	
Configured Address Physical Read Multiple Write (FRMW) (0x0E)	
Logical Read (LRD) (0x0A)	Logical Addressing In each slave, local address spaces can be mapped to global logical address spaces. These commands thus address all EtherCAT slaves for whom mapping for the selected logical address space is configured. ⇨ Chapter "2.5.5 Logical addressing", page 13 ⇨ Chapter "2.12 Field bus memory management unit (FMMU)", page 26
Logical Write (LWR) (0x0B)	
Logical Read Write (LRW) (0x0C)	
Broadcast Read (BRD) (0x07)	Broadcast Addressing All slave nodes are addressed automatically. ⇨ Chapter "2.5.4 Broadcast addressing", page 12
Broadcast Write (BWR) (0x08)	
Broadcast Read Write (BRW) (0x09)	

2.5.7 Working counter (WKC)

The EtherCAT datagram ends with a 2 byte working counter (WKC). Each datagram has an expected WKC value, which is calculated by the master before sending the telegram. This way the master can control whether the telegram was processed correctly.

We distinguish between different telegrams:

- Write telegram without reading
- Read telegram without writing
- Read-write telegram within one telegram

Following successful processing of a read telegram or write telegram by the slave, the WKC in the telegram is incremented by one and sent to the next slave.

Following successfully processed read-write-telegram within one telegram by the slave, the WKC in the telegram is incremented:

- by one for a successful write,
- by two for a successful read.

That means: by three for a successful read-write.

2.6 Mailbox protocol

The mailbox protocol is transferred by the datagram protocol.
⇨ Chapter "2.5 Datagram protocol", page 11

The mailbox protocol provides an abstraction layer with which existing non-real-time capable communication and field bus protocols can be integrated into the EtherCAT protocol. According to the IEC 61158-4-12, EtherCAT supports six types of mailbox sub protocols. These types are:

- Mailbox Error Protocol (0x00) for diagnosis of mailbox communication.
- Ethernet over EtherCAT (EoE) (0x02) for tunneling Ethernet protocols via EtherCAT (not used).
- CANopen over EtherCAT (CoE) (0x03) for tunneling the CANopen protocol via EtherCAT.
- File over EtherCAT (FoE) (0x04) a simple protocol for file transfer (not used).
- Servo profile over EtherCAT (SoE) (0x05) for tunneling the Sercos protocol via EtherCAT (not used).
- Vendor over EtherCAT (VoE) (0x0F) for tunneling manufacturer-specific protocols (not used).

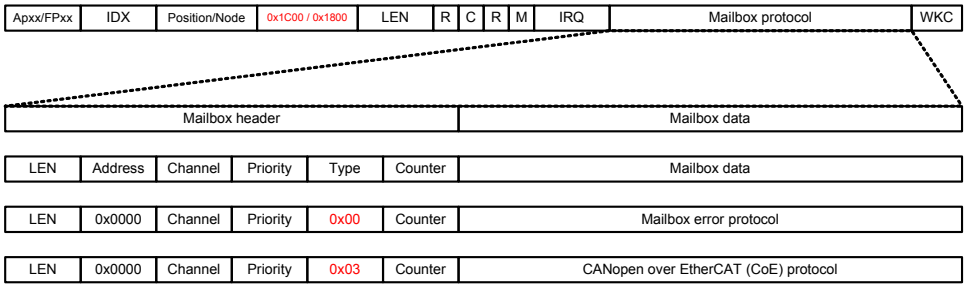


Figure 5: Mailbox protocol

The address field in the mailbox header is not considered and only plays a role for direct transfer of the mailbox protocol within the scope of the EtherCAT Automation Protocol.

2.6.1 Mailbox error protocol (0x00)

The mailbox error protocol is for diagnosing the mailbox. It reports configuration errors of the mailbox. The error protocol can be analyzed by the EtherCAT master.
⇨ Chapter "8 Diagnostics", page 157

2.6.2 CANopen over EtherCAT® (CoE) protocol (0x03)

CANopen over EtherCAT (CoE) is used for tunneling individual sub protocols of the CANopen protocol. The servo valve supports asynchronous transfer for the service data objects (SDO) and the emergency messages. The CANopen protocol will be described in more detail in the following chapter.
⇨ Chapter "2.7 CANopen protocol", page 15

2.7 CANopen protocol

CANopen is based on the CAN protocol. The CAN protocol defines the physical transfer and addressing of the telegrams. The different CANopen protocols are distinguished via the CAN ID. For each protocol, a number sequence with the number of maximum possible slaves (127 slaves) is reserved. With this it is possible to address a slave via the CAN ID and, with the same CAN ID, to select a specific protocol interface/channel (PDO, SDOs) of this slave.

For EtherCAT, the slave is addressed by the datagram addressing. CANopen protocol selection for EtherCAT takes place via the CoE command. The following CANopen protocols can be tunneled via EtherCAT:

- Emergency (EMCY) protocol (0x01)
- Transmit service data object (TxSDO) protocol (0x02)
- Receive service data object (RxSDO) protocol (0x03)
- Transmit process data object (TxPDO) protocol (0x04) (not used)
- Receive process data object (RxPDO) protocol (0x05) (not used)
- Transmit process data object remote (TxPDO remote) protocol (0x06) (not used)
- Receive process data object remote (TxPDO remote) protocol (0x07) (not used)
- Service data object Information (SDO Info) protocol (0x08) (not used)

For PDO communication the normal EtherCAT datagram protocol is used. Therefore the mailbox PDO protocols are not needed.

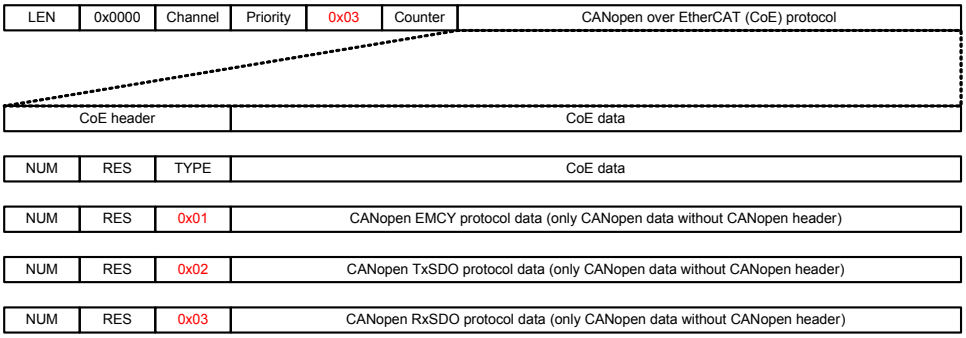


Figure 6: CANopen protocol

2.7.1 Emergency (EMCY) protocol (0x01)

The Emergency protocol is a high priority message triggered by an error event in the servo valve. The CANopen communication profile (CiA 301) defines the emergency error codes.
⇨ Chapter "8 Diagnostics", page 157

2.7.2 Transmit/receive service data object (SDO) protocol (0x02/0x03)

Service data objects are used to configure the cyclic communication parameters and the application parameters.

2.8 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.

2.8.1 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.

2.8.2 Parameter and their attributes

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

Block name							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default

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

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

Column name	**Meaning**
Block name	Describes the family of the object. If the object does not belong to a block, the object name is taken as block name.
Object name	Defined name of the object.
Index	16 bit index that addresses the entry in the object dictionary. In case of a simple variable this references the value of this variable directly. In case of records and arrays, the index addresses the whole data structure. Then the 8 bit sub-index allows access to individual elements in the structure.
Sub-index	If the object is defined as a record or array, the sub-index defines an element in the structure.
Parameter name	Defined name of the parameter.
Data type	Data type of the parameter. INTn Integer with n bits FLOAT32 Floating point with 32 bit char Character (ASC II) STRING String of characters UINTn Unsigned integer with n bits DOMAIN Application specific data block
Access	Access permission for the parameter. rw Read and write allowed wo Write only ro Read only
Persistence	Defines whether the parameter can be saved in non-volatile memory. If the persistence is set to "Y", the saved value stays in memory even after the device is turned off. Parameters not marked as persistent ("N") lose their settings after the device is turned off. The parameters with the access type "read only" are marked with "-". This means that the parameter cannot be changed by the user.
Value range	Allowed value range for the object.

Table 2: Field bus independent attributes (part 1 of 2)

Column name	Meaning
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 171
Specification	Related (field bus) standard defining the parameter. Possible entries: CiA 301 Parameters correspond to CiA 301 (CANopen). CiA 408 Parameters correspond to CiA 408. Moog DCV Moog defined parameters for digital control valves.
PDO mapping	If set to "Y", the parameter can be mapped into a PDO. If set to "N", the parameter cannot be mapped into a PDO.
Short name	Unique short name.

Table 2: Field bus independent attributes (part 2 of 2)

WARNING

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

2.8.3 Units and prefix parameter

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

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

Table 3: Unit representation

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

Table 4: Prefix representation

2.9 CANopen object dictionary (OD)

All CANopen objects are summarized in the object dictionary. The object dictionary is the link between the application and the CANopen communication unit in the CANopen device model. Each entry in the object dictionary represents one object and is marked by a 16 bit index. An index can contain up to 256 sub-indices for the individual parameters of the objects. The classification of the object dictionary is defined in the CiA 301.

Index	Object	Reference
0x0000	Not used	
0x0001…0x001F	Data types	CiA 301
0x0020…0x003F	Complex data types (not used)	CiA 301
0x0040…0x005F	Manufacturer-specific complex data types (not used)	
0x0040…0x025F	Device profile specific data types (not used)	CiA 408
0x0260…0x03FF	Reserved for further use	
0x0400…0x0FFF	Reserved for further use	
0x1000…0x1FFF	Communication profile area	CiA 301 / IEC 61158-5-12
0x2000…0x5FFF	Manufacturer-specific area	Moog DCV
0x6000…0x67FF	Standardized profile area 1st logical device	CiA 408
0x6800…0x9FFF	Standardized profile area 2nd…8th logical device (not used)	CiA 301
0xA000…0xAFFF	Standardized network variable area (not used)	
0xB000…0xBFFF	Standardized system variable area (not used)	
0xC000…0xFFFF	Reserved for further use	

Table 5: Structure of the CANopen object dictionary (OD)

2.10 Process data object (PDO) communication

2.10.1 Process data object mapping (PDO mapping)

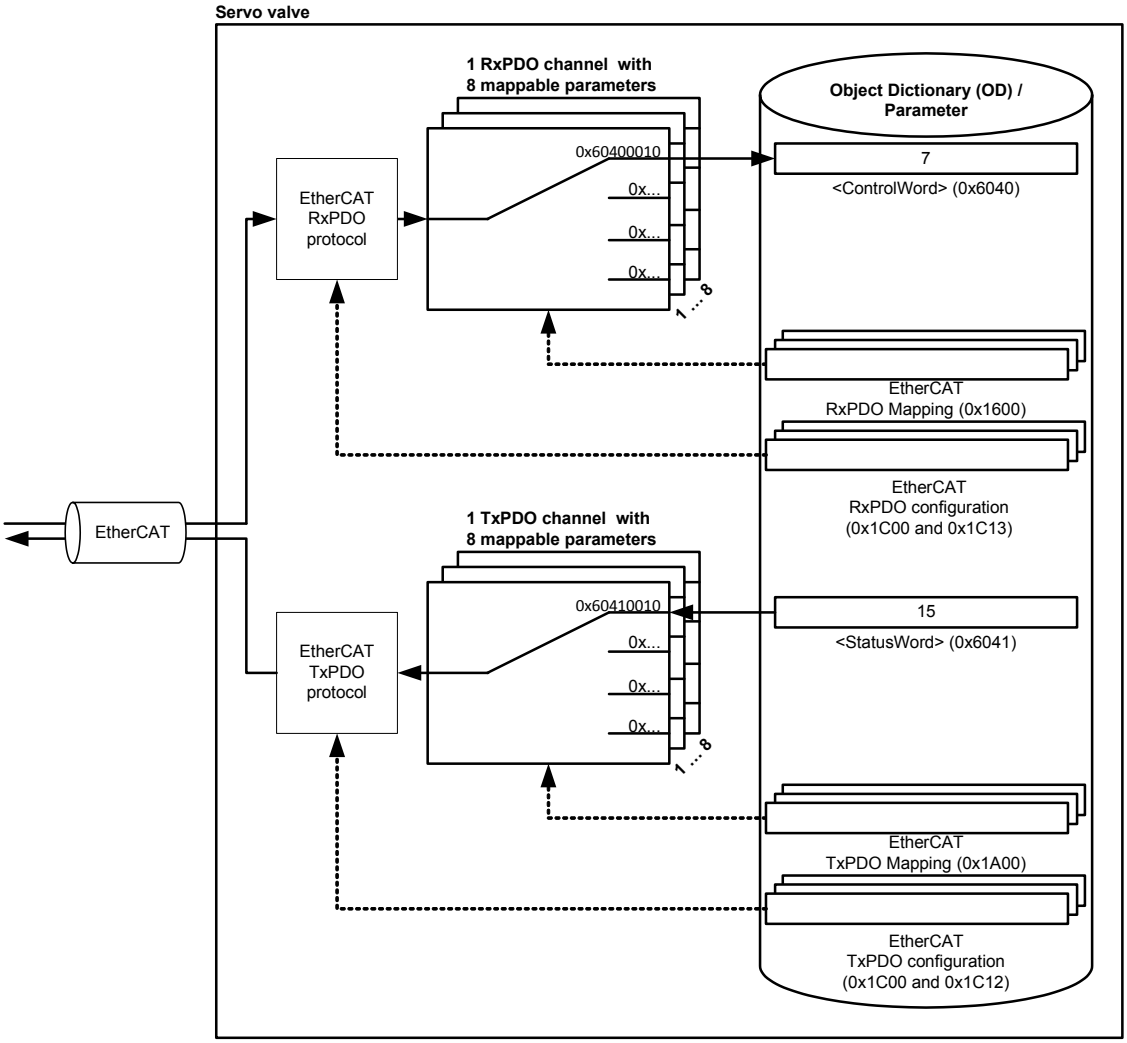


Figure 7: Process data object mapping (PDO mapping)

The object 0x1600 represents the RxPDO mapping and the object 0x1A00 represent the TxPDO mapping. Sub-index 0x00 contains the number of valid mapping entries within the mapping object. Sub-index from 0x01 to 0x08 contains the references to the mapped application parameters. The application parameters are referenced by their index, sub-index and length. The length contains the length of the application parameter in bit. This may be used to verify the mapping.

Parameter addressing				
Byte	**3**	**2**	**1**	**0**
Description	Index MSB	Index LSB	Sub-index	Parameter bit length: 0x10

The following procedure shall be used for re-mapping, which may take place during the EtherCAT 'INIT' state or in the 'PRE-OPERATIONAL' state (no PDO communication in these states):

1. Disable the mapping by setting the parameter with sub-index 0x00 of the object 0x1600 or object 0x1A00 to the value 0x00.
2. Modify the mapping by changing the parameters with sub-index 0x01…0x08 of the object 0x1600 or object 0x1A00.
3. Enable the mapping by setting the parameter with sub-index 0x00 of the object 0x1600 or object 0x1A00 to the number of mapped objects.

2.10.1.1 Object 0x1600: RxPDO mapping

With the parameter <RPdo_NumberOfMappedApplicParaInPdo> (0x1600) the number of real-time application parameters to be received can be set. To map the application parameter itself, its index, sub-index and length must be combined to a 32 bit number and written to one of the eight possible parameters <RPdo1_ApplicPara1…8> (0x1600) within the PDO object.

EtherCAT							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1600	0	RPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	3
0x1600	1	RPdo1_ApplicPara1	UINT32	rw	Y	UINT32	0x60410010
0x1600	2	RPdo1_ApplicPara2	UINT32	rw	Y	UINT32	0x63000110
0x1600	3	RPdo1_ApplicPara3	UINT32	rw	Y	UINT32	0x63800110
0x1600	4	RPdo1_ApplicPara4	UINT32	rw	Y	UINT32	0x00000000
0x1600	5	RPdo1_ApplicPara5	UINT32	rw	Y	UINT32	0x00000000
0x1600	6	RPdo1_ApplicPara6	UINT32	rw	Y	UINT32	0x00000000
0x1600	7	RPdo1_ApplicPara7	UINT32	rw	Y	UINT32	0x00000000
0x1600	8	RPdo1_ApplicPara8	UINT32	rw	Y	UINT32	0x00000000

Default configuration 3 according to CiA 408:

The following three parameters from the object dictionary are mapped as process data parameters by default:

- Device state machine (DSM) Control Word <ControlWord> (0x6040) (sub-index 0x00)
  ⇨ Chapter "5.2 Device state machine (DSM)", page 40
- Spool position setpoint value <SplSetpoint> (0x6300) (sub-index 0x01)
  ⇨ Chapter "6.2.3 Spool (stroke ring) position setpoint value path", page 53
- Pressure setpoint value <PrsSetpoint> (0x6380) (sub-index 0x01)
  ⇨ Chapter "6.2.4 Pressure setpoint value path", page 55

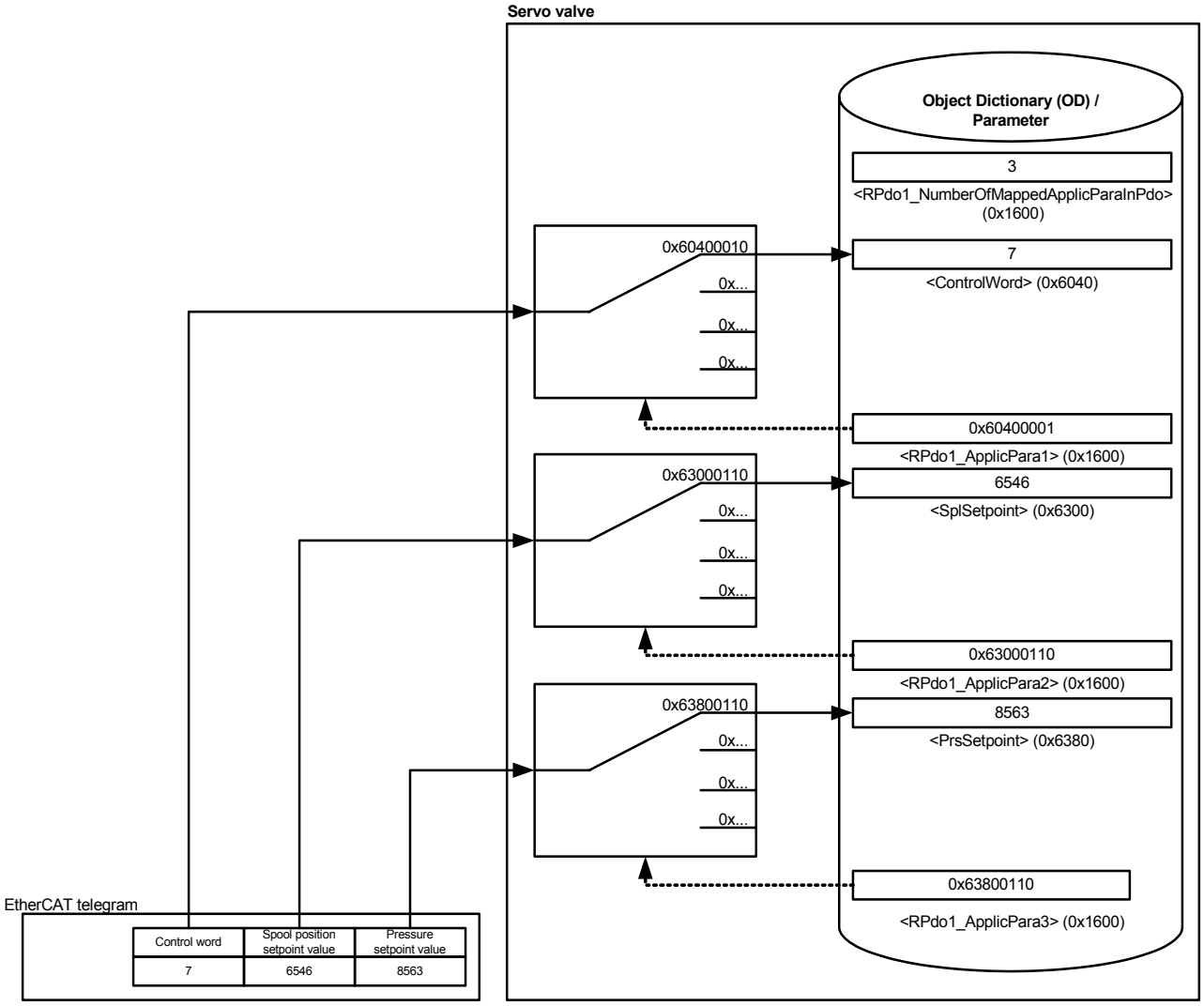


Figure 8: RxPDO mapping

The number of parameters to be mapped is entered in the first sub-index of the RxPDO mapping object (index 0x1600). As first PDO parameter, the <ControlWord> parameter with index 0x6040 and sub-index 0x00 is referenced in the second sub-index of the RxPDO mapping object (index 0x1600). The length of the <ControlWord> (0x6040) parameter is specified as 0x10 (16 bit). The same procedure is performed for the next two mapped parameters.

2.10.1.2 Object 0x1A00: TxPDO mapping

With the parameter <TPdo_NumberOfMappedApplicParaInPdo> (0x1A00) the number of real-time application parameters to be transmitted can be set. To map the application parameter itself, its index, sub-index and length must be combined to a 32 bit number and written to one of the eight parameters <TPdo1_ApplicPara1…8> (0x1A00) within the PDO object.

EtherCAT							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1A00	0	TPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	3
0x1A00	1	TPdo1_ApplicPara1	UINT32	rw	Y	UINT32	0x60410010
0x1A00	2	TPdo1_ApplicPara2	UINT32	rw	Y	UINT32	0x63000110
0x1A00	3	TPdo1_ApplicPara3	UINT32	rw	Y	UINT32	0x63800110
0x1A00	4	TPdo1_ApplicPara4	UINT32	rw	Y	UINT32	0x00000000
0x1A00	5	TPdo1_ApplicPara5	UINT32	rw	Y	UINT32	0x00000000
0x1A00	6	TPdo1_ApplicPara6	UINT32	rw	Y	UINT32	0x00000000
0x1A00	7	TPdo1_ApplicPara7	UINT32	rw	Y	UINT32	0x00000000
0x1A00	8	TPdo1_ApplicPara8	UINT32	rw	Y	UINT32	0x00000000

Default configuration 3 according to CiA 408:

The following three parameters from the object dictionary are mapped as process data parameters by default:

- Device state machine (DSM) Status Word <StatusWord> (0x6041) (sub-index 0x00)
  ⇨ Chapter "5.2 Device state machine (DSM)", page 40
- Spool position actual value <SplActualValue> (0x6301) (sub-index 0x01)
  ⇨ Chapter "7.3 Stroke ring (spool) position controller", page 96
- Pressure actual value <PrsActualValue> (0x6381) (sub-index 0x01)

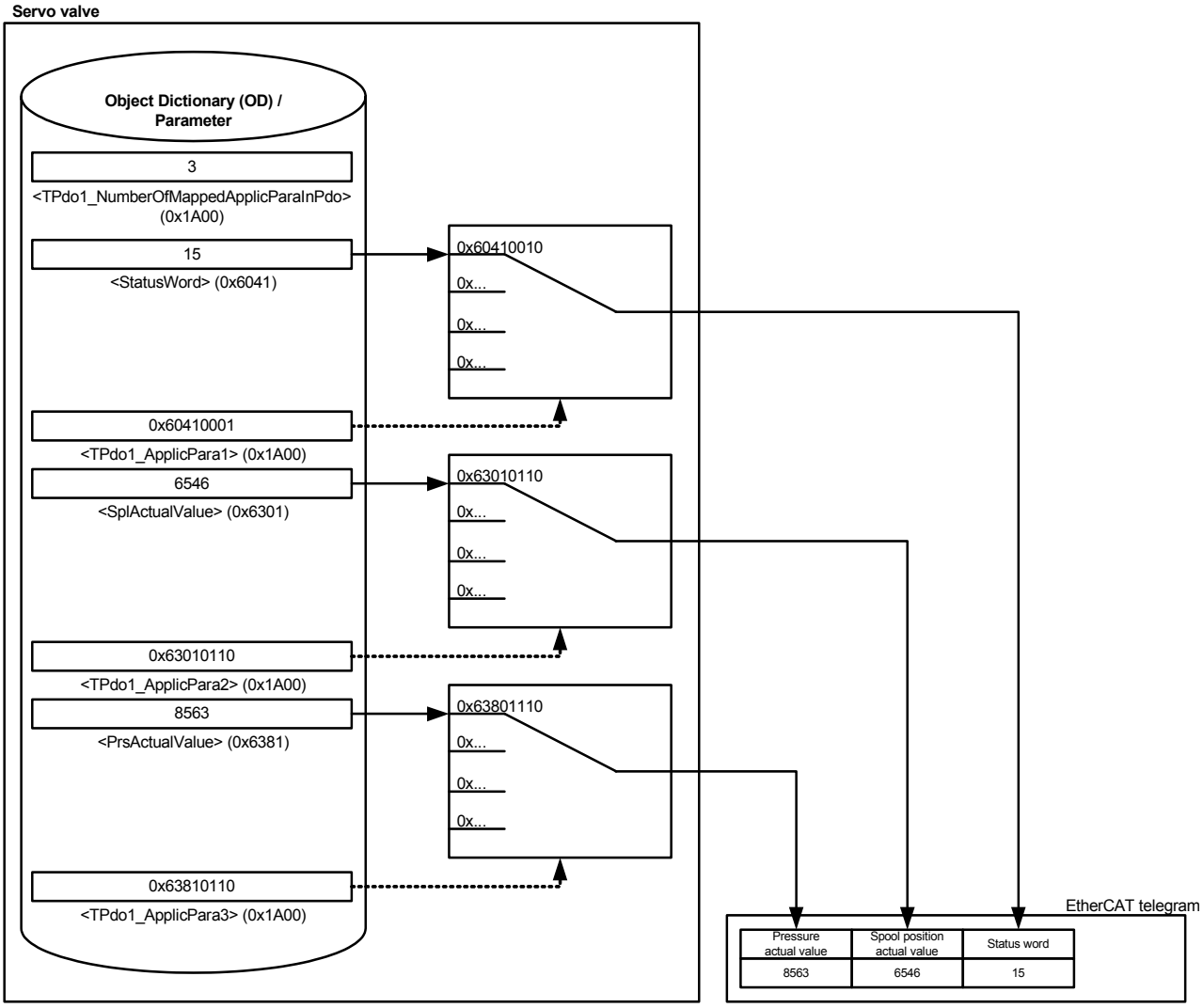


Figure 9: TxPDO mapping

The number of parameters to be mapped is entered in the first sub-index of the TxPDO mapping object (index 0x1A00). As first PDO parameter, the <StatusWord> parameter with index 0x6041 and sub-index 0x00 is referenced in the second sub-index of the TxPDO mapping object (index 0x1A00). The length of the <StatusWord> (0x6041) parameter is specified as 0x10 (16 bit). The same procedure is performed for the next two mapped parameters.

2.10.2 PDO watchdog (SM watchdog)

The PDO watchdog is implemented as a synchronization manager watchdog (SM watchdog). The SM watchdog monitors the PDO communication between EtherCAT master and servo valve. If for example there is, due to a disconnection, no EtherCAT communication with the servo valve for longer than the specified SM watchdog period time, the watchdog acts and freezes the nominal setpoint value. At the same time an error message is generated (emergency error code 0x8231). For the SM watchdog, a fault reaction can be configured via the SDO 0x2830, sub-index 114.

⇨ Chapter "8.1 Fault reaction", page 157

To activate the watchdog the ESC register 0x0814 "SM2 Ctrl/Status", bit 6 has to be set. This must be done by the network master.

If the master configuration tool does not set this bit automatically, you have to set the bit manually.

⇨ Chapter "2.10.2.1 Turn on SM watchdog manually", page 24

The watchdog time can be configured in the ESC register 0x0420 "WD Time SM". The watchdog can be disabled by setting the watchdog time to 0x0000.

PDO communication can be delayed (depending on the master) while debugging the PLC program and thus trigger the SM watchdog.

2.10.2.1 Turn on SM watchdog manually

Some network masters use the information from the EtherCAT slave information (ESI) file to configure the watchdog bit within the "SM2 Ctrl/Status" register 0x0814. In this case the ESI file has to be modified. Therefore the bit 0x40 of the SM2 control byte has to be set in the ESI file. Afterwards the master has to re-read the ESI file.

Line to be changed in the ESI file:
<Sm StartAddress="#x1000" ControlByte="**#x24**" Enable="1">Outputs</Sm>

Change to:
<Sm StartAddress="#x1000" ControlByte="**#x64**" Enable="1">Outputs</Sm>

If the control byte was copied correctly to the ESC, bit 6 should be set in the ESC register 0x0814 "SM2 Ctrl/Status" following the network status change from Preop to Saveop.

Here a short recipe to switch the SM watchdog on or off:

1. Change SM control byte in the ESI file.
2. Reload ESI files in the master configuration tool (for safety delete EtherCAT slave devices/boxes of this manufacturer and product code in the master configuration before reloading).
3. Load ESI file into the Slave ESI EEPROM using the master configuration tool.
4. Re-start the network state machine.
5. Check ESC register 0x0814 "SM2 Ctrl/Status" bit 6.
6. Check ESC register 0x0420 "WD Time SM".

2.11 Synchronization manager (SM)

The SM allows consistent and secure data exchange between the EtherCAT master and the slave device. The SM generates interrupts to inform both sides about changes.

The SM settings in the object dictionary have read only access. These settings will be read by the master on start up from the digital servo valve and are needed to configure the process data and Mailbox communication (depending on EtherCAT master).

2.11.1 Object 0x1C00: SM communication type

EtherCAT							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1C00	0	SM_CommunicationType_NoOfChannels	UINT8	ro	-	0…4	4
0x1C00	1	SM0_Communication_Type	UINT8	ro	-	UINT8	0x01 (mailbox receive)
0x1C00	2	SM1_Communication_Type	UINT8	ro	-	UINT8	0x02 (mailbox send)
0x1C00	3	SM2_Communication_Type	UINT8	ro	-	UINT8	0x03 (process data receive)
0x1C00	4	SM3_Communication_Type	UINT8	ro	-	UINT8	0x04 (process data send)

2.11.2 Object 0x1C10: SM 0 receive PDO assignment

EtherCAT							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1C10	0	SM0_MBX_Receive_NoOfAssignedPDOs	UINT8	ro	-	0…4	0

2.11.3 Object 0x1C11: SM 1 transmit PDO assignment

EtherCAT							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1C11	0	SM1_MBX_Send_NoOfAssignedPDOs	UINT8	ro	-	0…4	0

2.11.4 Object 0x1C12: SM 2 receive PDO assignment

EtherCAT							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1C12	0	SM2_NoOfAssignedReceivePDOs	UINT8	ro	-	0…4	1
0x1C12	1	SM2_receive_Pdo1_Assignment	UINT16	ro	-	UINT16	0x1600
0x1C12	2	SM2_receive_Pdo2_Assignment	UINT16	ro	-	UINT16	0x0000
0x1C12	3	SM2_receive_Pdo3_Assignment	UINT16	ro	-	UINT16	0x0000
0x1C12	4	SM2_receive_Pdo4_Assignment	UINT16	ro	-	UINT16	0x0000

2.11.5 Object 0x1C13: SM 3 transmit PDO assignment

EtherCAT							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x1C13	0	SM3_NoOfAssignedTransmitPDOs	UINT8	ro	-	0…4	1
0x1C13	1	SM3_transmit_Pdo1_Assignment	UINT16	ro	-	UINT16	0x1A00
0x1C13	2	SM3_transmit_Pdo2_Assignment	UINT16	ro	-	UINT16	0x0000
0x1C13	3	SM3_transmit_Pdo3_Assignment	UINT16	ro	-	UINT16	0x0000
0x1C13	4	SM3_transmit_Pdo4_Assignment	UINT16	ro	-	UINT16	0x0000

2.12 Field bus memory management unit (FMMU)

The field bus memory management unit (FMMU) converts logical addresses into physical addresses by the means of internal address mapping. Thus, FMMUs allow using logical addressing for data segments that span several slave devices: one datagram addresses data within several arbitrarily distributed ESCs. Each FMMU channel maps one continuous logical address space to one continuous physical address space of the slave. The number of supported FMMUs depends on the ESC. The access type supported by an FMMU is configurable to be either read, write, or read/write.

2.13 Distributed clocks (DC)

The distributed clock protocol ensures that the process data will be processed with a very low jitter.
As the internal clock cycle of the servo valves is less than 350 µs, with a sampling time of the master of 1 ms, relatively jitter-free collection of process data via the master can be ensured even without DC. As the mechanical time constants for typical hydraulic servo applications are larger than 1 ms, implementation of the DCs was dispensed with.

2.14 EtherCAT® network state machine (ESM)

Each network slave has an EtherCAT network slave state machine (ESM) which coordinates the start-up and behavior during normal operation and is controlled by the master. The network state machine is also called Application Layer Status Machine. In the subsequent parts the abbreviation AL stands for application layer.

The EtherCAT network state machine must not be mistaken with the device state machine.

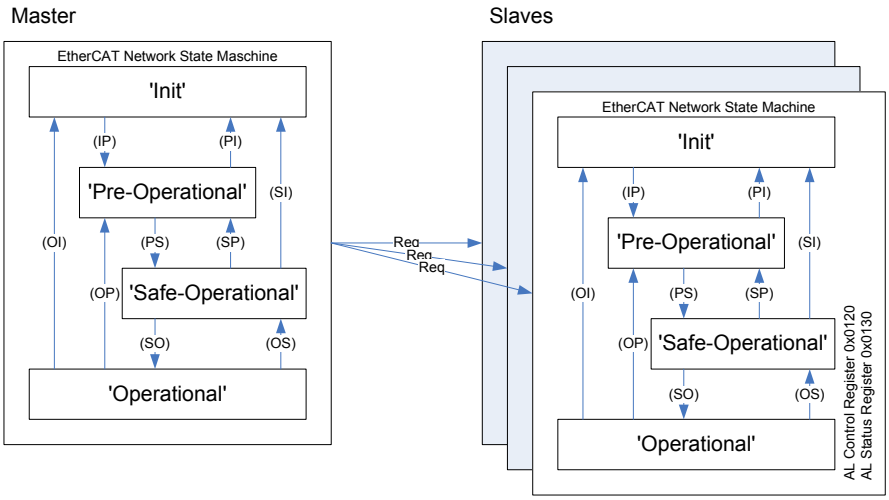


Figure 10: EtherCAT network state machine (ESM)

The EtherCAT State Machine is represented in following ESC registers:

Register Name	Register	Description
AL Status Control	0x0120	Represents the requested state of the EtherCAT network state machine.
AL Status Status	0x0130	Represents the actual state of the EtherCAT network state machine.
AL Status Code	0x0134	Not supported by the servo valve hardware (ESC10 chip).

2.14.1 State machine states

State	Description
'INIT'	Default state after power on No communication on Application (PDO / SDO)
'PRE-OPERATIONAL'	Network communication (Mailbox / SM / FMMU) initialized SDO communication is enabled PDO communication is disabled
'SAFE-OPERATIONAL'	Mailbox communication on the Application Layer SDO communication is enabled PDO communication: Only actual values from the servo valve are sent to the master PDO communication: Setpoint values from the master are not sent to the servo valve. They remain in 'SAFE-OPERATIONAL' state
'OPERATIONAL'	Mailbox communication on the Application Layer SDO communication is enabled PDO communication is enabled

Table 6: EtherCAT network state machine states

If the SM watchdog is activated and the SM watchdog detects a PDO communication fault, the EtherCAT network state machine (ESM) falls back to the 'SAFE-OPERATIONAL' state.
⇨ Chapter "2.10.2 PDO watchdog (SM watchdog)", page 24

2.15 EtherCAT® network configuration

The configuration of an EtherCAT network is done by special configuration tools. These tools need the Electronic Data Sheets (EDS) files and the EtherCAT Slave Information (ESI) files of the connected EtherCAT devices.

2.15.1 Electronic data sheet (EDS) files

Electronic Datasheets (EDS) are files which describe the capabilities of CANopen devices and are therefore crucial to CANopen. The object dictionary with all parameters is described in the EDS file. The EDS file is used for the CANopen over EtherCAT (CoE) communication. The servo valve EDS file is provided by Moog or can be downloaded from the Moog website http://www.moog.com/industrial.

The EtherCAT master refers the EDS file with in the ESI file. In the ESI file you will find the XML tag EtherCATInfo/Descriptions/Device/Mailbox/CoE/EdsFile with the EDS filename.

2.15.2 EtherCAT® slave information (ESI) files

The ESI file contains data for configuration of the servo valve. These include configuration of the EtherCAT slave controller (ESC), product identification, PDO communication, FMMU configuration as well as SM configuration. The data are stored in the ESI file in XML format. For this reason the file extension is *.xml. These data is stored permanently in the ESI EEPROM and can be overwritten by the network master. The ESI files provide the possibility for offline configuration, e.g. without available slave devices.

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

Only the master can change the ESI EEPROM content of the servo valve (depending on the slave hardware).

Offline configuration of the EtherCAT network is only possible if the ESI file is available for the EtherCAT master.

Unfortunately there is no standard procedure yet for configuring EtherCAT devices by the network master. Different network master manufacturers have implemented different ways for the configuration of EtherCAT slave interfaces. When starting up the network, the master configures the slave by writing the configuration data into the ESC registers of the slave. The configuration data required for this can originate from different sources:

- ESI file (ESC configuration, Identity object, PDO mapping, supported protocols)
- ESI EEPROM (ESC configuration, Identity object, supported protocols)
- SDO objects of the slaves (Identity object, PDO mapping)

Not all configuration data are available in all data sources. Masters of different manufacturers may only use some of these sources and may use them in differing order.

For masters expecting the PDO and SM watchdog configuration in the ESI EEPROM, it must be stored in the ESI EEPROM. Moog does not store the PDO configuration in the ESI EEPROM by default.

2.16 EtherCAT® Application Layer (AL) status code register (0x134:0x135)

The Application Layer (AL) status code register is a register in the DP-RAM of each EtherCAT slave. It is intended to show further information in case of a network error which is indicated by the error indication flag (0x130.4=1). The following table shows the AL status codes and the relevant states.

Code	Description	Network state or transition
0x0000	No error	Any
0x0001	Unspecified error	Any
0x0011	Invalid requested state change	'INIT'→'SAFE-OPERATIONAL', 'INIT'→'OPERATIONAL', 'PRE-OPERATIONAL'→'OPERATIONAL'
0x0012	Unknown requested state	Any
0x0013	Bootstrap not supported	'INIT'→Bootstrap
0x0014	No valid firmware	'INIT'→'PRE-OPERATIONAL'
0x0016	Invalid mailbox configuration	'INIT'→'PRE-OPERATIONAL'
0x0017	Invalid sync manager configuration. Number of mapped parameters in PDO not correct.	'PRE-OPERATIONAL'→'SAFE-OPERATIONAL', 'SAFE-OPERATIONAL'→'OPERATIONAL'
0x0018	No valid setpoint inputs	'OPERATIONAL', 'SAFE-OPERATIONAL', 'PRE-OPERATIONAL'→'SAFE-OPERATIONAL'
0x0019	No valid actual value outputs	'OPERATIONAL', 'SAFE-OPERATIONAL'→'OPERATIONAL'
0x001A	Synchronization error	'OPERATIONAL', 'SAFE-OPERATIONAL'→'OPERATIONAL'
0x001B	Sync manager watchdog timed out	'OPERATIONAL', 'SAFE-OPERATIONAL'
0x001C	Invalid sync manager types	'OPERATIONAL', 'SAFE-OPERATIONAL', 'PRE-OPERATIONAL'→'SAFE-OPERATIONAL'
0x001D	Invalid output configuration. Number of mapped parameters in RxPDO not correct or number of bytes of the mapped parameters do not match.	'OPERATIONAL', 'SAFE-OPERATIONAL', 'PRE-OPERATIONAL'→'SAFE-OPERATIONAL'
0x001E	Invalid input configuration. Number of mapped parameters in TxPDO not correct or number of bytes of the mapped parameters do not match.	'OPERATIONAL', 'SAFE-OPERATIONAL', 'PRE-OPERATIONAL'→'SAFE-OPERATIONAL'
0x001F	Invalid watchdog configuration. Check configuration in SII and ESI file, 'ControlByte' of SM with Start Address 0x1000.	'OPERATIONAL', 'SAFE-OPERATIONAL', 'PRE-OPERATIONAL'→'SAFE-OPERATIONAL'

Table 7: AL status codes (part 1 of 2)

Code	Description	Network state or transition
0x0020	Slave needs cold start	Any
0x0021	Slave needs 'INIT'	'PRE-OPERATIONAL', 'SAFE-OPERATIONAL', 'OPERATIONAL'
0x0022	Slave needs 'PRE-OPERATIONAL'	'SAFE-OPERATIONAL', 'OPERATIONAL'
0x0023	Slave needs 'SAFE-OPERATIONAL'	'OPERATIONAL'
0x0043	Mailbox COE. Check size of mailbox in ESI file and SII.	'PRE-OPERATIONAL', 'SAFE-OPERATIONAL', 'OPERATIONAL'

Table 7: AL status codes (part 2 of 2)

For your notes.

3 Device structure

3.1 Overview

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

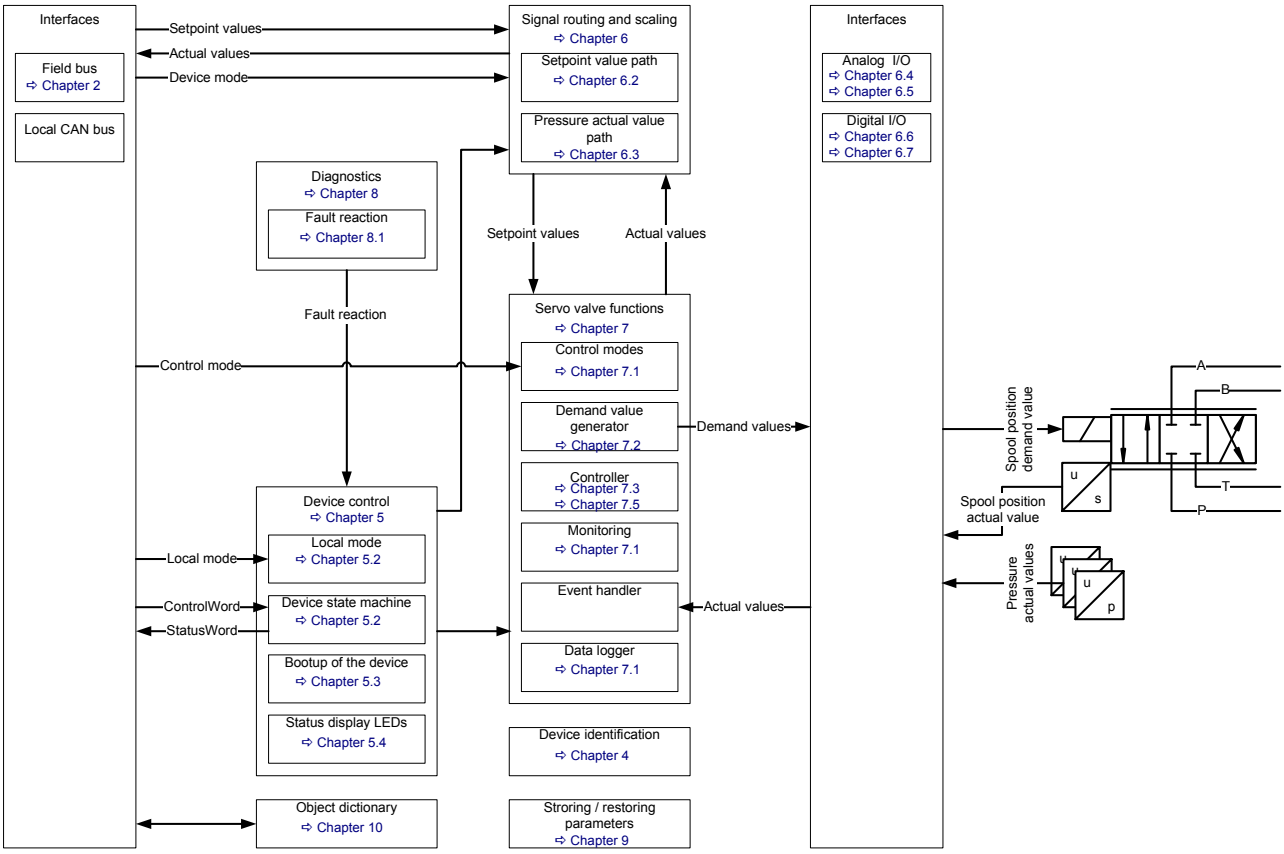


Figure 11: Device structure

3.1.1 Device identification

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

3.1.2 Device control

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

3.1.3 Signal routing and scaling

This block conditions the raw input signals for the signal processing and maps these to the internal servo valve functions. This mapping depends among other things on the <DeviceMode> (0x6042) and the device state machine state.

3.1.4 Servo valve functions

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

3.1.5 Diagnostics

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

3.1.6 Storing / restoring parameters

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

3.1.7 Object dictionary

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

3.2 Device controller structure

The following figure shows the device structure with focus on the signal flow. Depending on the <ControlMode> (0x6043) only a subset of the device structure may be used.

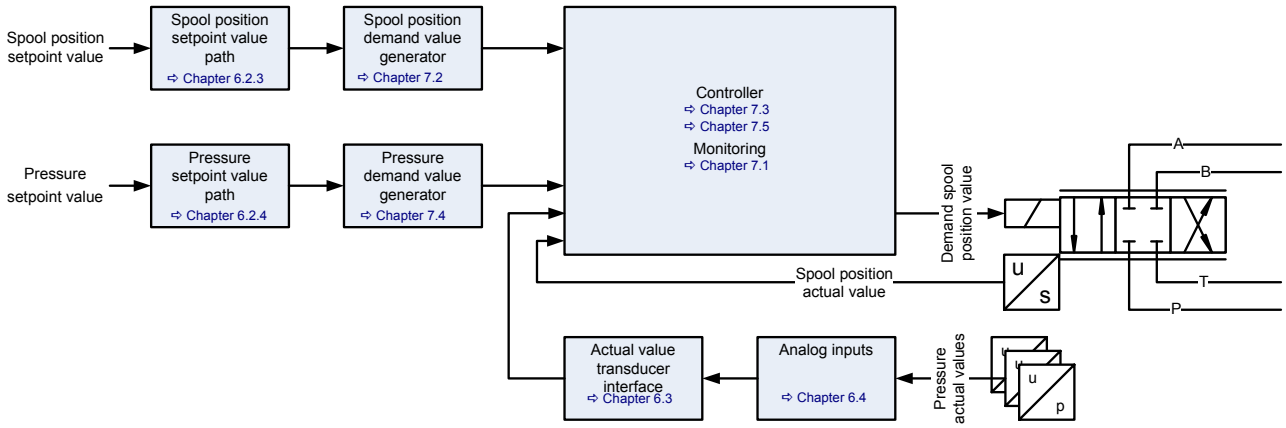


Figure 12: Device controller structure

4 Device identification

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

4.1 Objects of the CANopen communication profile defined by CiA 301

4.1.1 Object 0x1000: Device Type

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

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

Value description

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

Table 8: Possible values of parameter <DeviceType> (0x1000)

4.1.2 Object 0x1008: Manufacturer device name

This parameter indicates the name of the servo valve.

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

4.1.3 Object 0x1009: Manufacturer hardware version

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

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

4.1.4 Object 0x100A: Manufacturer software version

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x100A	0	ManufacturerSoftwareVersion	STRING	ro	-	None	DSV

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	Persistence	Value range	Default
0x1018	1	VendorId	UINT32	ro	-	UINT32	40
0x1018	2	ProductCode	UINT32	ro	-	UINT32	0
0x1018	3	RevisionNumber	UINT32	ro	-	UINT32	0
0x1018	4	SerialNumber	UINT32	ro	-	UINT32	0

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

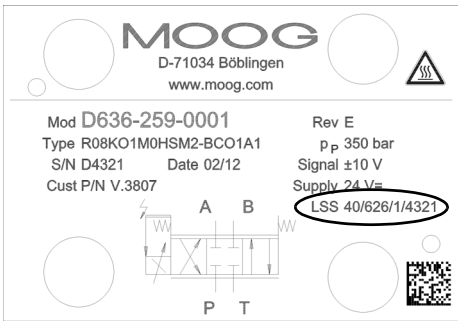


Figure 13: Name plate of the device with identification object address

Value description

Sub-index	Parameter	Meaning
1	<VendorId>	Unique vendor ID, 0x28 reserved for Moog.
2	<ProductCode>	Product code of the Moog servo valve. Each valve model number has its own CANopen product code.
3	<RevisionNumber>	Revision number of the Moog servo valve.
4	<SerialNumber>	Serial number of the Moog servo valve (digits without leading character) as on the name plate. ⇨ Chapter "4.2.3 Object 0x6052: Serial number", page 35

Table 9: Possible values of parameter Identity object (0x1018)

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

4.2 Objects defined by Device Profile Fluid Power

4.2.1 Object 0x6050: Version

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

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

4.2.2 Object 0x6051: Code number

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

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

4.2.3 Object 0x6052: Serial number

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

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

4.2.4 Object 0x6053: Description

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

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

4.2.5 Object 0x6054: Model description

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

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

4.2.6 Object 0x6055: Model URL

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

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

4.2.7 Object 0x6056: Parameter set code

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

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

4.2.8 Object 0x6057: Vendor name

This parameter indicates the name of the device vendor.

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

4.2.9 Object 0x605F: Capability

This object provides information on the capabilities of the used device, e.g. the supported control modes.

Device							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x605F	0	Capability	UINT32	ro	-	UINT32	0x3F000000

Value description

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

Table 10: Possible values of parameter <Capability> (0x605F)

Bit	**Description**	**Default**
0…15	Reserved	
16	Drive capability	0
17	Supports control mode drive open loop movement	0
18	Supports control mode drive velocity control	0
19	Supports control mode drive force control	0
20	Supports control mode drive position control	0
21	Supports control mode drive flow control	0
22, 23	Reserved	
24	Servo valve or pump capability	1
25	Supports control mode spool position open loop	1
26	Supports control mode spool position closed loop	1
27	Supports control mode pressure open loop	1
28	Supports control mode pressure closed loop	1
29	Supports control mode p/Q closed loop	1
30	Reserved	0
31	Supports modular device according CiA 301	0

Table 11: Bit values of parameter <Capability> (0x605F)

5 Device control

The device control contains a device state machine (DSM) which activates or deactivates the servo valve. The states of the DSM can be changed externally by the <ControlWord> (0x6040) which can be set via bus using a PDO or locally by the parameter value <LocalControlWord> (0x4040). It is possible to configure a power on delay for the communication. The status LEDs display the current network and device states.

5.1 Local mode

The source of the control word, acting on the device state machine, is defined by the parameter <Local> (0x604F) as shown in the following figure.

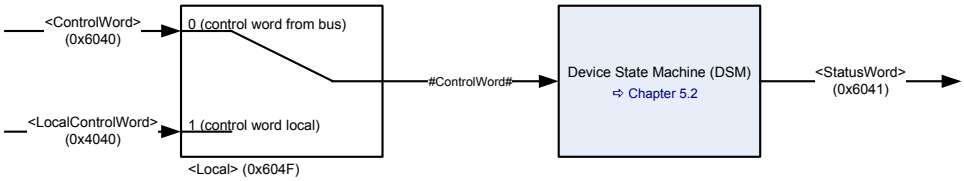


Figure 14: Local mode

5.1.1 Object 0x604F: Local

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

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

Value description

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

Table 12: Possible values of parameter <Local> (0x604F)

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

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

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46

5.1.2 Object 0x6040: Control word

The functionality of the <LocalControlWord> (0x4040) and the <ControlWord> needs to be distinguished. If the parameter <Local> (0x604F) is set, the DSM input #ControlWord# comes from the local parameter <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

<ControlWord>				
Bit	**Description**	**<Control Mode> (0x6043) is set to 1…4**	**<Control Mode> (0x6043) is set to 5**	**Specification**
0	Bit Disabled (D)			CiA 408
1	Bit Hold (H)			
2	Bit Active (M)			
3	Bit Reset Faults (R)			
4…7	Reserved			
8	<ControlMode> (0x6043) specific	Reserved	Enable pressure controller	CiA 408
9	Slave mode enable			DS 408
10	Leakage compensation enable			DS 408
11	Power limitation enable			RKP-D specific
12	Reserved			
13	Hold pressure enable			RKP-D specific
14	Hold pressure forced			RKP-D specific
15	Ramp stop			Moog DCV

Table 13: Possible values of parameter <ControlWord> (0x6040)

Bits 0, 1, 2, 3: 'DISABLED', 'HOLD', 'ACTIVE', 'RESET FAULTS'
The lower four bits within the control word represent the device state machine's (DSM) control command.
⇨ Chapter "5.2.2.1 DSM state transitions caused by the control word", page 43

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

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

⇨ Chapter "7.1 Control modes", page 82

Bit 9: Slave mode enable
This bit is used to enable the slave mode of the pump.

Bit 10: Leakage compensation enable
This bit is used to enable/disable the leakage compensation.

Bit 11: Power limitation enable
Enables/disables the power limitation function.

Bit 13: Hold pressure enable
Enables/disables the local holding pressure switchover function.

Bit 14: Hold pressure forced
Enables/disables externally forcing of the holding pressure switchover.

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

Spool position demand value generator: ⇨ Chapter "7.2.5 Ramp", page 89
Pressure demand value generator: ⇨ Chapter "7.4.5 Ramp", page 104

5.1.3 Object 0x4040: Local control word

Parameter description: ⇨ Chapter "5.1.2 Object 0x6040: Control word", page 38

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

Value description

Same values as <ControlWord> (0x6040)
⇨ Table 13, page 38

5.1.4 Object 0x403F: Local control word default

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

Device							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x403F	0	LocalControlWordDefault	UINT16	rw	Y	UINT16	0x0107

5.2 Device state machine (DSM)

The device state machine (DSM) describes the states of the servo valve and the transitions between them. Any state represents a certain internal and external behavior. State changes result from DSM input and other events (for example switching on the supply voltage or on the appearance of a device fault). The current device state can be read by means of the <StatusWord> (0x6041) (bits 0…3 of the status word indicate the device condition).

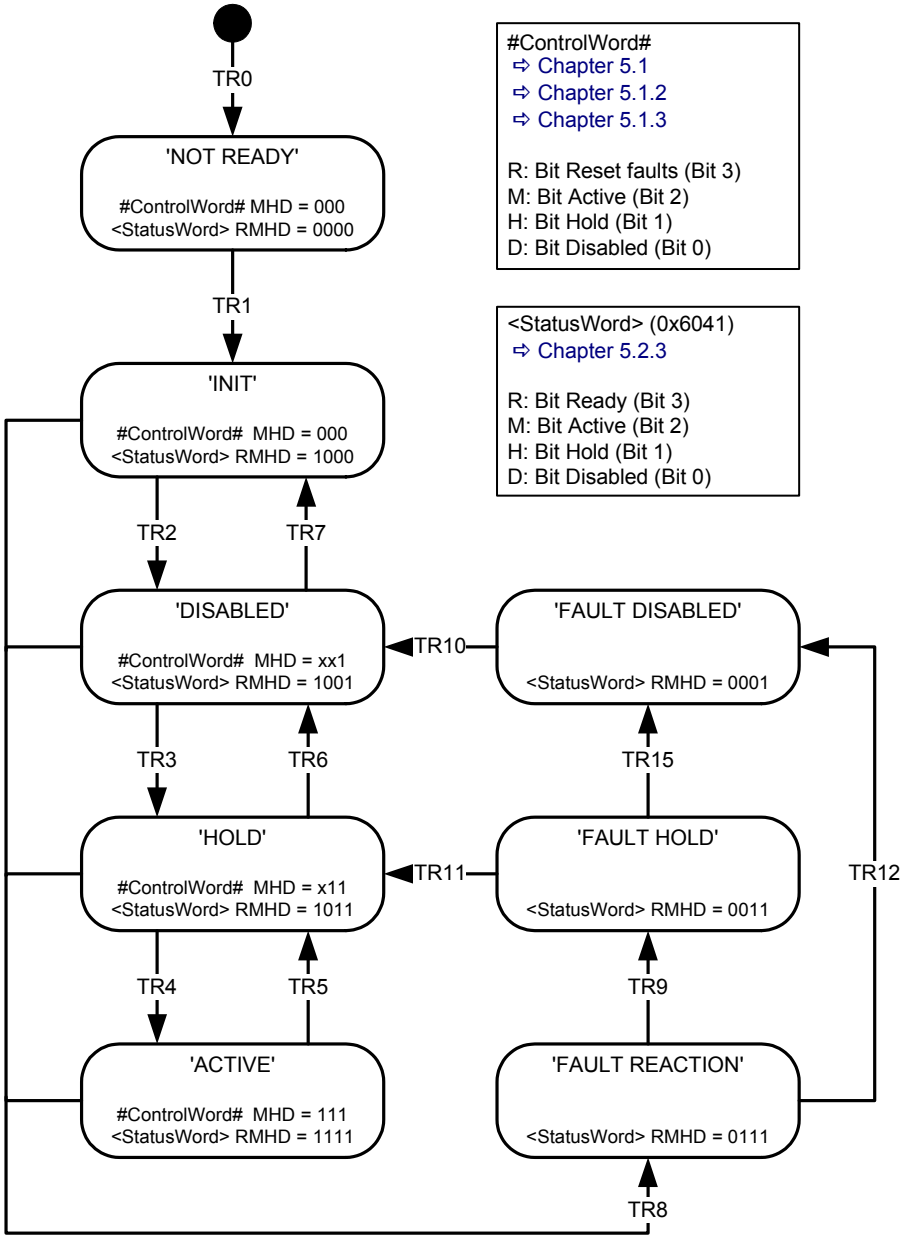


Figure 15: Device state machine

5.2.1 DSM states

The <StatusWord> (0x6041) indicates the DSM state. The following DSM states are possible:

'NOT_READY':

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

'INIT':

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

'DISABLED':

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

'HOLD':

- Device parameters can be set.
- Actual values are available.
- The selected <ControlMode> (0x6043) is active.
  ⇨ Chapter "7.1 Control modes", page 82
- The setpoint values from the bus or from the analog input according to the chosen <DeviceMode> (0x6042) are not effective.
  ⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 52
- The spool closed loop is active and the spool goes to the new setpoint, namely the 'spool hold setpoint' as specified by <SplHoldSetPoint> (0x6314). When the pressure controller is active, the pressure set-point changes to the <PrsHoldSetPoint> (0x6394).
  ⇨ Chapter "6.2.3.3 Object 0x6314: Hold setpoint", page 54
  ⇨ Chapter "6.2.4.3 Object 0x6394: Hold setpoint", page 56

'ACTIVE':

- Device parameters can be set.
- Actual values are available.
- The setpoint values from the bus or from the analog input according to the chosen <DeviceMode> (0x6042) are active.

'FAULT 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 82
- The setpoint values from the bus or from the analog input according to the chosen <DeviceMode> (0x6042) are not effective.
  ⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 52
- The pre-defined, electrically controlled hold spool position setpoint value <SplHoldSetPoint> (0x6314) or the hold pressure setpoint value <PrsHoldSetPoint> (0x6394) is active.
  ⇨ Chapter "6.2.3.3 Object 0x6314: Hold setpoint", page 54
  ⇨ Chapter "6.2.4.3 Object 0x6394: Hold setpoint", page 56

'FAULT_REACTION':

- This state is assumed when the device detects an error.
- A fault dependent vendor specific action is executed.
  ⇨ Chapter "8.1.3 Fault reaction type", page 162
- The resulting fault state depends on the vendor specific <FaultReactionType> (0x2830).
- Important condition for transitions 9, 12, 13:
  The RMHD bits of the #ControlWord# do not increase the state of the DSM.

Coming from	Meaning
'DISABLED'	'FAULT_DISABLED'
'HOLD'	'FAULT_HOLD' 'FAULT_DISABLED'
'ACTIVE'	'FAULT_HOLD' 'FAULT_DISABLED'
'FAULT_HOLD'	'FAULT_HOLD' 'FAULT_DISABLED'
'FAULT_DISABLED'	'FAULT_DISABLED'

The default state after power on if <LocalMode> (0x604F) is set to 1 (Control Word Local) and enable signal on is defined by the parameter <LocalControlModeDefault> (0x403F).
⇨ Chapter "5.1.4 Object 0x403F: Local control word default", page 39

WARNING

The word "failsafe" means not a personnel safety. If a personnel safe servo valve is needed, some additional electrical and hydraulic parts are necessary.

5.2.2 State transitions

State transitions are caused by

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

5.2.2.1 DSM state transitions caused by the control word

The following table lists the transitions depending on the #ControlWord#.

The device control commands, which cause a state transition, are formed by the four low-order bits of the #ControlWord#.
⇨ Chapter "5.1 Local mode", page 37

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

		Control word bit								Comments/Conditions
		7	6	5	4	3	2	1	0	
Transition (TR)	**Control Word**					**R**	**M**	**H**	**D**	
TR2	Activate 'DISABLED'	x	x	x	x	x	x	x	1	
TR3	Activate 'HOLD'	x	x	x	x	x	x	1	1	Depending on enable signal ⇨ Chapter "5.2.2.4 Enable behavior", page 44
TR4	Activate 'ACTIVE'	x	x	x	x	x	1	1	1	Depending on enable signal ⇨ Chapter "5.2.2.4 Enable behavior", page 44
TR5	Deactivate 'ACTIVE	x	x	x	x	x	0	x	X	
TR6	Deactivate 'HOLD'	x	x	x	x	x	0	0	X	
TR7	Deactivate 'DISABLED'	x	x	x	x	x	0	0	0	
TR10	Reset 'FAULT_DISABLED'	x	x	x	x	0	0	0	1	This transition is executed if the reset bit changes from 0 to 1 (rising edge) or the enable signal toggles from 0 to 1. ⇨ Chapter "5.2.2.4 Enable behavior", page 44 Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 45
		change to								
		x	x	x	x	1	0	0	1	
TR11	Reset 'FAULT_HOLD'	x	x	x	x	0	0	1	1	This transition is executed if the reset bit changes from 0 to 1 (rising edge) or the enable signal toggles from 0 to 1. ⇨ Chapter "5.2.2.4 Enable behavior", page 44 Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 45
		change to								
		x	x	x	x	1	0	1	1	
TR15	'FAULT_HOLD' to 'FAULT_DISABLED'	x	x	x	x	x	0	0	1	

5.2.2.2 DSM state transitions caused by the enable signal

The following events will lead automatically to state changes.

Transition (TR)	Description	Comments/Conditions
TR2	Activate 'DISABLED'	Depending on ControlWord RMHD ≥ x001
TR3	Activate 'HOLD'	Depending on ControlWord RMHD ≥ x011
TR4	Activate 'ACTIVE'	Depending on ControlWord RMHD ≥ x111
TR5	Deactivate 'ACTIVE'	
TR6	Deactivate 'HOLD'	
TR7	Deactivate 'DISABLED'	
TR9	Transition from 'FAULT_REACTION' to 'FAULT_HOLD'	Depending on the enable behavior
TR10	Reset 'FAULT_DISABLED'	Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 45
TR11	Reset 'FAULT_HOLD'	Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 45
TR15	'FAULT_HOLD' to 'FAULT_DISABLED'	Depending on the enable behavior

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

5.2.2.3 DSM state transitions caused by internal events

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

Transition (TR)	Description	Comments/Conditions
TR0	Power up	
TR1	Device init successful.	Initialization of device parameters with stored values. Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 45
TR8	A fault was detected. On entering 'FAULT_REACTION' state an emergency message is sent out.	If state is 'DISABLED' or 'FAULT_DISABLED', state transitions to 'FAULT_HOLD' will be redirected to 'FAULT DISABLED'. Behavior of error output pin: ⇨ Chapter "5.2.2.5 Error output pin", page 45
TR9	Transition from 'FAULT_REACTION' to 'FAULT_HOLD' (fault reaction successful).	Depending on enable signal and the enable behavior. ⇨ Chapter "5.2.2.4 Enable behavior", page 44
TR12	Transition from 'FAULT_REACTION' to 'FAULT_DISABLED' (fault reaction successful).	Depending on enable signal and the enable behavior. ⇨ Chapter "5.2.2.4 Enable behavior", page 44

5.2.2.4 Enable behavior

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

⇨ Chapter "6.6.1 Digital input 0 (enable signal)", page 76

5.2.2.4.1 DSM state transitions depending on the enable signal

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

Servo valves with the type designator 13 in the order code of D, F and H:
If the enable signal is switched off (0 V), the spool position is forced to its spring centered failsafe position.

WARNING

The word "failsafe" means not a personnel safety. If a personnel safe servo valve is needed, some additional electrical and hydraulic parts are necessary.

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

Servo valves with the type designation number 13 in the order code C, G, E, J, S and T:
If the enable signal is turned off (0 V), the spool position setpoint is changed to the electrical spool hold position <SplHoldSetPoint> (0x6314) or <PrsHoldSetPoint> (0x6394).

⇨ Chapter "5.2.1 DSM states", page 41
⇨ Chapter "6.2.3 Spool (stroke ring) position setpoint value path", page 53
⇨ Chapter "6.2.4 Pressure setpoint value path", page 55

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

5.2.2.4.2 Fault confirmation with the enable signal

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

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

5.2.2.5 Error output pin

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

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

To enable this behavior on the digital output 1, the parameter <DigitalOutputType1> (0x2420) must be set to 2.
⇨ Chapter "6.7.2 Object 0x5E41: Digital output type", page 77

5.2.3 Object 0x6041: Status word

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

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

Value description

<StatusWord>				
Bit	**<StatusWord> bit**	**<Control Mode> (0x6043) is set to 1…4**	**<Control Mode> (0x6043) is set to 5**	**Specification**
0	Bit Disabled (D)			CiA 408
1	Bit Hold (H)			
2	Bit Active (M)			
3	Bit Ready (R)			
4	Indicates that bit <Local> (0x604F) is set			CiA 408
5…7	Reserved			
8	<ControlMode> (0x6043) specific	Reserved	Pressure controller effective	CiA 408
9	Ramp running			CiA 408
10	Limit touched (c)			CiA 408
11	Control deviation			CiA 408
12…14	Reserved			
15	Ramp frozen			Moog DCV

Table 14: Possible values of parameter <StatusWord> (0x6041)

Bits 0, 1, 2, 3: 'DISABLED', 'HOLD', 'ACTIVE', 'READY'
These bits indicate the state of the device state machine (DSM).
⇨ Chapter "5.2 Device state machine (DSM)", page 40

Bit 4: Indicates that bit <Local> (0x604F) is set
The <LocalControlWord> (0x4040) is the active control word.
⇨ Chapter "5.1 Local mode", page 37

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

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

⇨ Chapter "7.1 Control modes", page 82

This bit is only active if the <ControlMode> (0x6043) is set to 5 (p/Q control).

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

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

Spool position demand value generator: ⇨ Chapter "7.2.5 Ramp", page 89
Pressure demand value generator: ⇨ Chapter "7.4.5 Ramp", page 104

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

Spool position demand value generator: ⇨ Chapter "7.2.3 Limit function", page 87
Pressure demand value generator: ⇨ Chapter "7.4.3 Limit function", page 102

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 "7.12 Monitoring", page 133

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

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

Spool position demand value generator: ⇨ Chapter "7.2.5 Ramp", page 89
Pressure demand value generator: ⇨ Chapter "7.4.5 Ramp", page 104

5.3 Bootup of the device

The bootup sequence can by 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	Persistence	Value range	Default
0x200F	0	PowerOnDelay	UINT8	rw	Y	0…10	None

5.4 Status display LEDs

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

Figure 16: Status display LEDs

5.4.1 Module status LED «MS»

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

⇨ Chapter "5.2 Device state machine (DSM)", page 40

Module status LED «MS»	Description	Valve State Machine (status word) (according to Device Profile Fluid Power)
Off		No supply power.
Green blinking	'INIT' or 'DISABLED'	Servo valve standby mode.
Green	'HOLD' or 'ACTIVE'	Normal operation.
Red blinking	'FAULT_DISABLED' or 'FAULT_HOLD'	Recoverable error. Fault reactions 'FAULT', 'FAULT_HOLD': ⇨ Chapter "8.1.3 Fault reaction type", page 162
Red	'NOT_READY'	Unrecoverable error. Fault reaction 'NOT_READY': ⇨ Chapter "8.1.3 Fault reaction type", page 162

5.4.2 Network link/activity LED «L/A in» and «L/A out»

The network Link/Activity LEDs «L/A in» and «L/A out» display the state of the physical connection.

LEDs «L/A in» and «LA out»	Network Link/Activity State	Link	Activity
On	Physical connection established. No data transfer.	Yes	No
Flickering	Physical connection established and data transfer.	Yes	Yes
Off	No physical connection.	No	No

Attention, the LEDs «L/A in» and «LA out» are blinking fast to indicate an incomplete physical connection, e.g. caused by a cable break of a single wire. This can be confused with the flickering state. To distinguish between these two states, stop the network master to avoid network traffic. If the LED «L/A in» / «LA out» is still blinking very fast, please check the cabling.

5.4.3 Network Run LED «RUN»

The network Run LED «RUN» displays the state of the communication.

LED «RUN»	Network Link/Activity State
Off	Device is in state 'INIT'.
Blinking	Device is in state 'PRE-OPERATIONAL'.
Single flash	Device is in state 'SAFE-OPERATIONAL'.
On	Device is in state 'OPERATIONAL'.

5.4.4 Network status LED «NS in» and «NS out» on previous valves

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

Network status LED «L/A in» or «LA out»	NMT state machine (ESM)	Description
Off		No power supply or not connected.
Orange	'INIT'	Link up.
Green blinking	'PRE-OPERATIONAL' 'SAFE-OPERATIONAL'	Connected. SDO communication is possible.
Green	'OPERATIONAL'	Connected. SDO and PDO communication are possible.
Red		A network major error has occurred.

For your notes.

6 Signal routing and scaling

The term "spool position" used in this manual stands for the stroke ring position of the RKP-D.

6.1 Signal routing structure

The following picture shows the structure of the signal routing for the setpoint values and the physical actual values of the servo valve depending on the <ControlMode> (0x6043) used. The blocks with gray backgrounds are described in detail in this chapter.

⇨ Chapter "7.1 Control modes", page 82

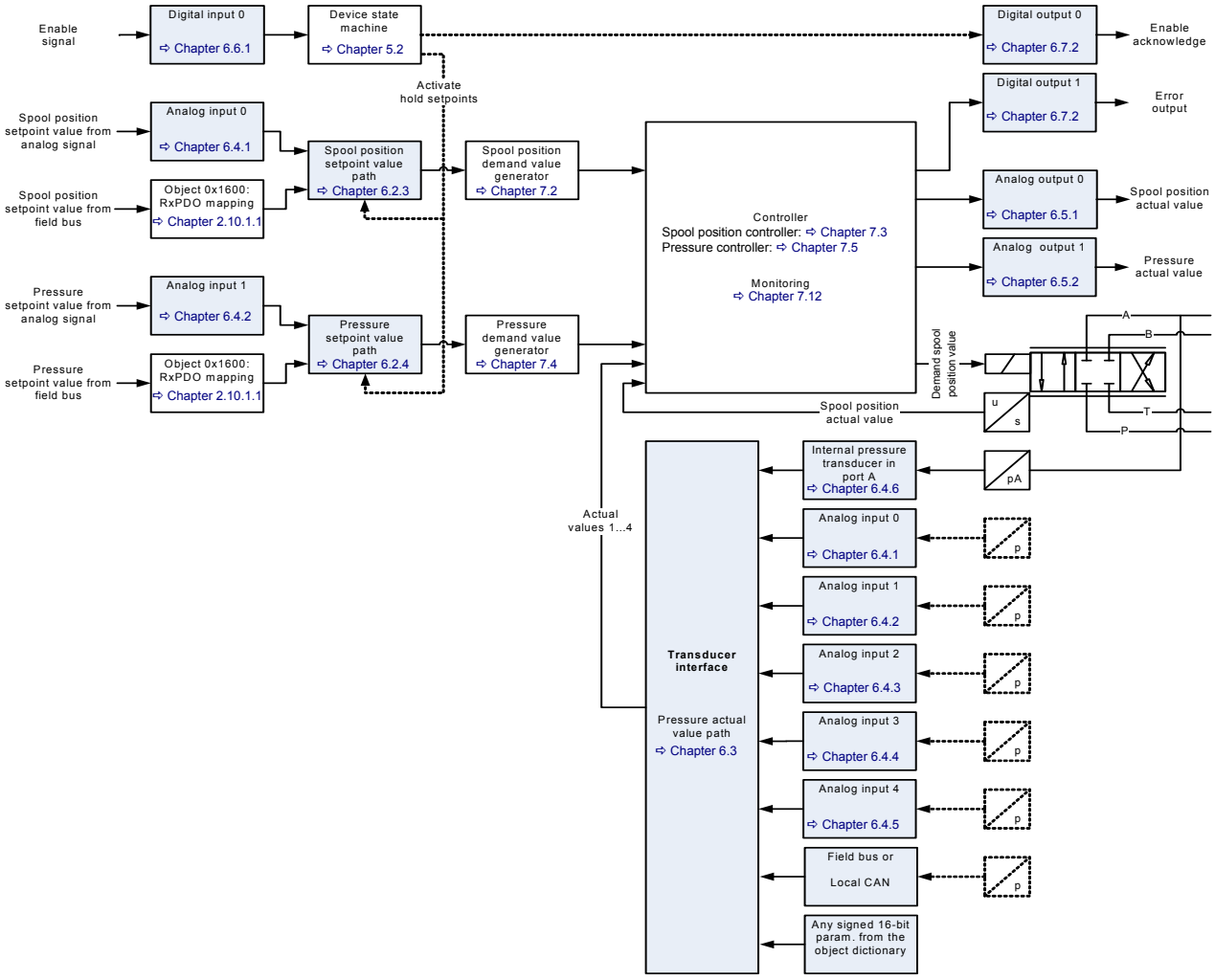


Figure 17: Signal routing

6.2 Setpoint value path

The Setpoint values for pressure and spool position can be received via the field bus or via the analog inputs.

- The spool position setpoint value received via bus is provided by the parameter <SplSetpoint> (0x6300).
- The pressure setpoint value received via bus is provided by the parameter <PrsSetpoint> (0x6380).
- The spool position setpoint value from local source is coming from the analog input 0 <ActualValue0> (0x3204).
- The pressure setpoint value from local source is coming from the analog input 1 <ActualValue1> (0x320C).

Which setpoint is in effect depends on the parameter <DeviceMode> (0x6042) and the <StatusWord> (0x6041):

- <DeviceMode> (0x6042) is set to 1 (setpoint input via bus), <StatusWord> (0x6041) is 0111_b ('ACTIVE'): The setpoint value received via field bus is forwarded to the demand value generator.
- <DeviceMode> (0x6042) is set to 2 (setpoint input locally), <StatusWord> (0x6041) is 0111_b ('ACTIVE'): The setpoint value coming from the analog input is forwarded to the demand value generator.
- <StatusWord> (0x6041) is 0011_b ('HOLD'):
  The hold setpoint values are taken as setpoint values. This hold setpoint is in effect regardless if the device mode <DeviceMode> (0x6042) is set to 1 (setpoint input via bus) or <DeviceMode> (0x6042) is set to 2 (setpoint input locally).
  The spool position hold setpoint value is stored in the parameter <SplHoldSetpoint> (0x6314). The pressure hold setpoint value is stored in the parameter <PrsHoldSetpoint> (0x6394).

6.2.1 Object 0x6042: Device mode

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

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

Value description

<DeviceMode>	**Type of analog input**
0	Reserved
1	Setpoint input via the bus
2	Setpoint input locally
All other	Reserved

Table 15: Possible values of parameter <DeviceMode> (0x6042)

The effective <DeviceMode> (0x6042) after power up is defined by the parameter <DeviceModeDefault> (0x4042).
⇨ Chapter "6.2.2 Object 0x4042: Device mode default", page 53

6.2.2 Object 0x4042: Device mode default

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

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

6.2.3 Spool (stroke ring) position setpoint value path

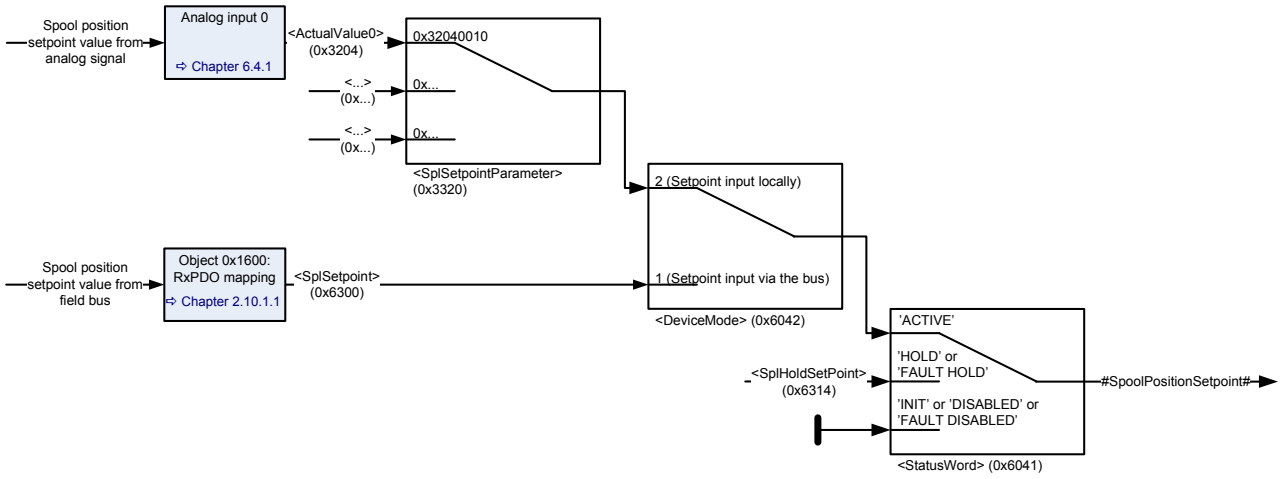


Figure 18: Spool position setpoint value path

6.2.3.1 Object 0x6300: Setpoint

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

1 Control spool position closed loop
2 Control spool position open loop
5 p/Q closed loop

The setpoint value <SplSetpoint> (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 "5.2.3 Object 0x6041: Status word", page 46
⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 52

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

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

6.2.3.2 Object 0x3320: Setpoint parameter

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

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

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46
⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 52

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

Value description

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

Table 16: Possible values of parameter <SplSetpointParameter> (0x3320)

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

6.2.3.3 Object 0x6314: Hold setpoint

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

1 Control position closed loop
2 Control position open loop
5 p/Q closed loop

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 82

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

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46

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

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

6.2.4 Pressure setpoint value path

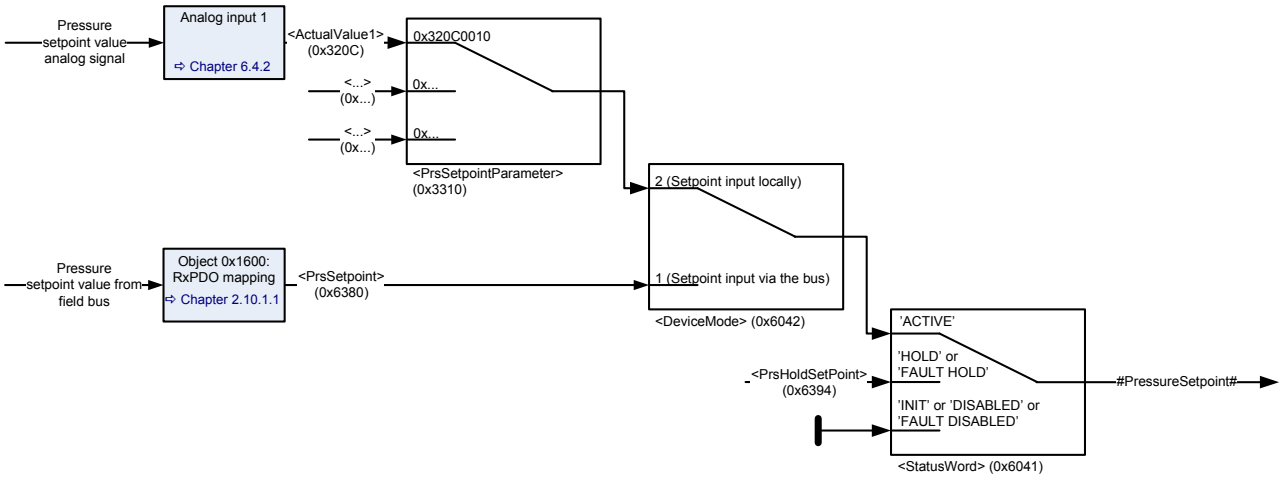


Figure 19: Pressure setpoint value path

6.2.4.1 Object 0x6380: Setpoint

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

3 Control pressure closed loop
4 Control pressure open loop
5 p/Q closed loop

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 82

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

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46
⇨ Chapter "6.2.1 Object 0x6042: Device mode", page 52

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

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

6.2.4.2 Object 0x3310: Setpoint parameter

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

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

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

Value description

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

Table 17: Possible values of parameter <PrsSetpointParameter> (0x3310)

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

6.2.4.3 Object 0x6394: Hold setpoint

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

3 Control pressure closed loop
4 Control pressure open loop
5 p/Q closed loop

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 82

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

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46

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

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

6.3 Actual value transducer interface

There are six available inputs which can be used as interface for the actual pressure value to be forwarded to the controller:

- Analog input 0…4
- Internal pressure sensor
- For special application any parameter can be mapped to an interface e.g. from a field bus.

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

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

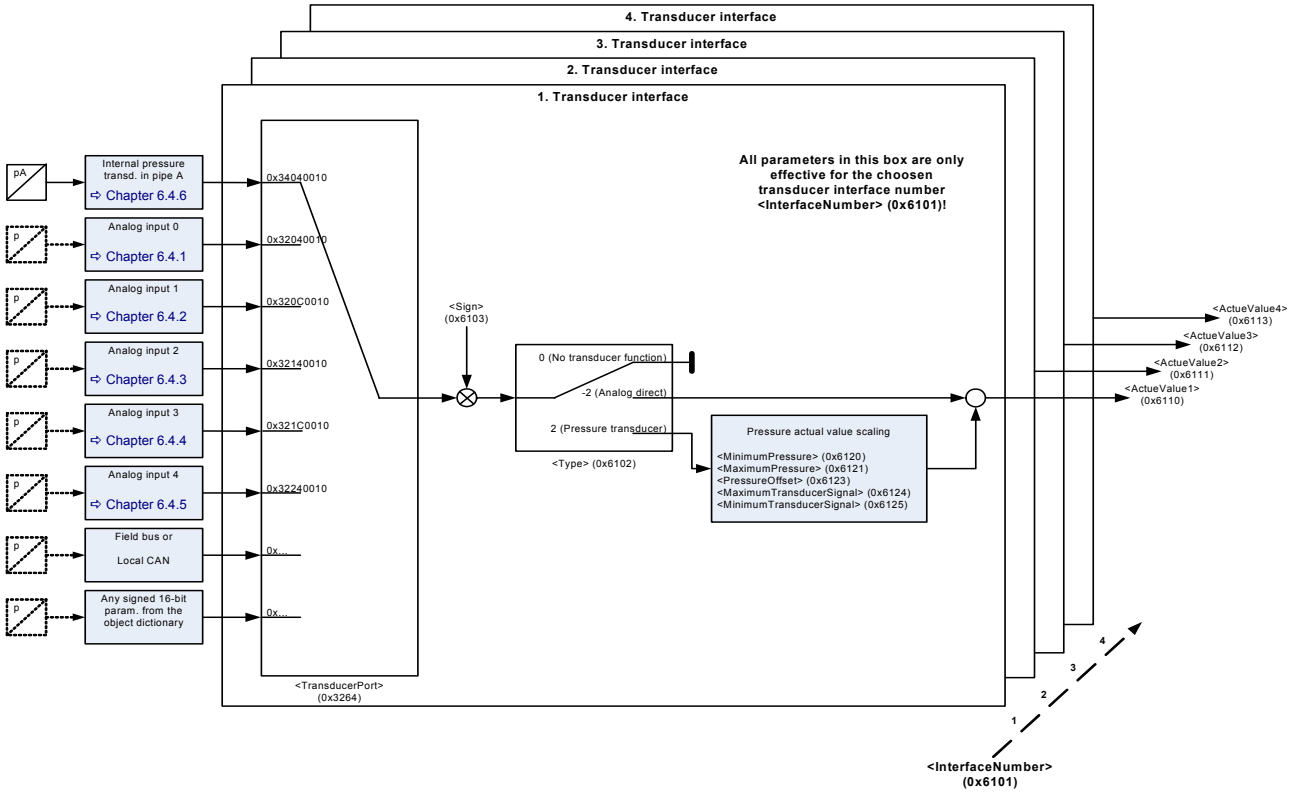


Figure 20: Actual value path

Per default, analog input 0 and 1 are used for the analog setpoint values. Therefore the inputs 2…4 are normally used for actual values.
⇨ Chapter "6.4 Analog inputs", page 68

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	Object Name
0x6100	Max interface number
0x6101	Interface number
0x6102	Type
0x6103	Sign
0x6104	Actual value
0x3264	Transducer port
0x6120	Minimum pressure
0x6121	Maximum pressure
0x6123	Pressure offset
0x6124	Maximum transducer signal
0x6125	Minimum transducer signal

6.3.1 Object 0x6110: Actual value 1

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

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

6.3.2 Object 0x6111: Actual value 2

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

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

6.3.3 Object 0x6112: Actual value 3

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

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

6.3.4 Object 0x6113: Actual value 4

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

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

6.3.5 Transducer interface definition

The assignment of an actual value source to an interface is done by setting the parameter <TransducerPort> (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).

6.3.5.1 Object 0x6100: 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	Persis-tence	Value range	Default
0x6100	0	MaxInterfaceNumber	UINT8	ro	-	UINT8	4

6.3.5.2 Object 0x6101: Interface number

This parameter defines the actual referenced interface.

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

Value description

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

Table 18: Possible values of parameter <InterfaceNumber> (0x6101)

6.3.5.3 Object 0x6102: 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	Persis-tence	Value range	Default
0x6102	0	Type	INT8	rw	N	INT8	None

Value description

<Type>	Description
0 (no transducer function)	Interface deactivated.
2 (pressure transducer)	Pressure sensor actual value conditioning active. ⇨ Chapter "6.3.6 Pressure actual value scaling", page 62
5 (general transducer)	General input. Scaling with min/max values and offset.
–2 (analog direct)	No further scaling active.
All other values	Not used.

Table 19: Possible values of parameter <Type> (0x6102)

6.3.5.4 Object 0x6103: 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	Persis-tence	Value range	Default
0x6103	0	Sign	INT8	rw	N	–1 or +1	None

Value description

<Type>	Description
–1	Negative
1	Positive
All other values	Reserved

Table 20: Possible values of parameter <Sign> (0x6103)

6.3.5.5 Object 0x6104: 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	Persis-tence	Value range	Default
0x6104	0	ActualValue	INT16	ro	-	INT16	None

6.3.5.6 Object 0x3264: Transducer port

This parameter defines the transducer port where the actual physical values are coming from. The port is defined by a parameter index, sub-index and length.
This parameter setting is effective for the interface selected by <InterfaceNumber> (0x6101).

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3264	0	TransducerPort	UINT32	rw	N	UINT32	None

Value description

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

Table 21: Possible values of parameter <TransducerPort> (0x3264)

There are 6 analog inputs (0…4 and the internal pressure transducer). Each input has its actual value parameter. Each actual value can be assigned to an interface by the <TransducerPort> (0x6104) parameter as described in the examples below.

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

6.3.5.7 Object 0x3270: Servo valve transducer structure

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3270	0	ValveTransducerStructure	DOMAIN	rw	Y	None	

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

6.3.6 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 "6.3.5.3 Object 0x6102: Type", page 60

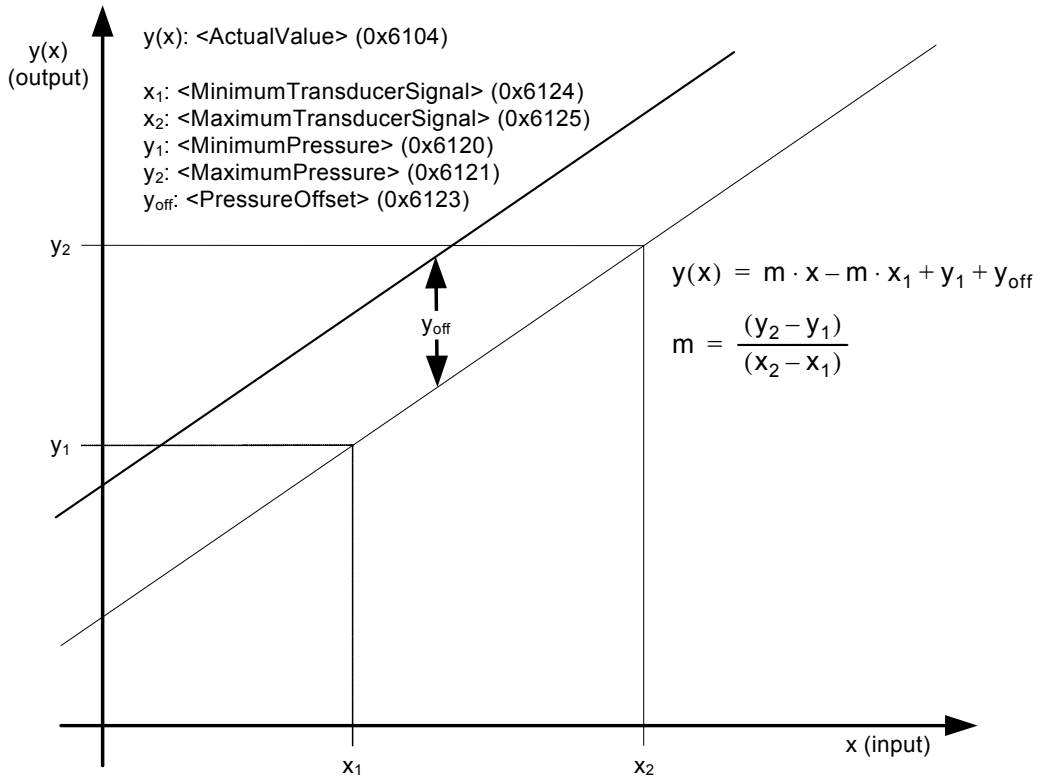


Figure 21: Pressure actual value scaling

6.3.6.1 Object 0x6120: Minimum pressure

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6120	1	MinimumPressure	INT16	rw	N	INT16	None

6.3.6.2 Object 0x6121: Maximum pressure

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6121	1	MaximumPressure	INT16	rw	N	INT16	16384

6.3.6.3 Object 0x6124: Minimum transducer signal

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6124	1	MinimumTransducerSignal	INT16	rw	N	INT16	None

6.3.6.4 Object 0x6125: Maximum transducer signal

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6125	1	MaximumTransducerSignal	INT16	rw	N	INT16	None

6.3.6.5 Object 0x6123: Pressure offset

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	Persis-tence	Value range	Default
0x6123	1	PressureOffset	INT16	rw	N	INT16	None

6.3.7 General input scaling

The general input is used to scale an INT32 input or an INT16 input. Example: an external pressure transducer with CAN interface is mapped to the receive PDO <Integer32> (0x0004). To activate the general input scaling, the interface type needs to be configured to "general input". This is done by writing the value 5 to the parameter <Type> (0x6102).

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

⇨ Chapter "6.3.5.3 Object 0x6102: Type", page 60

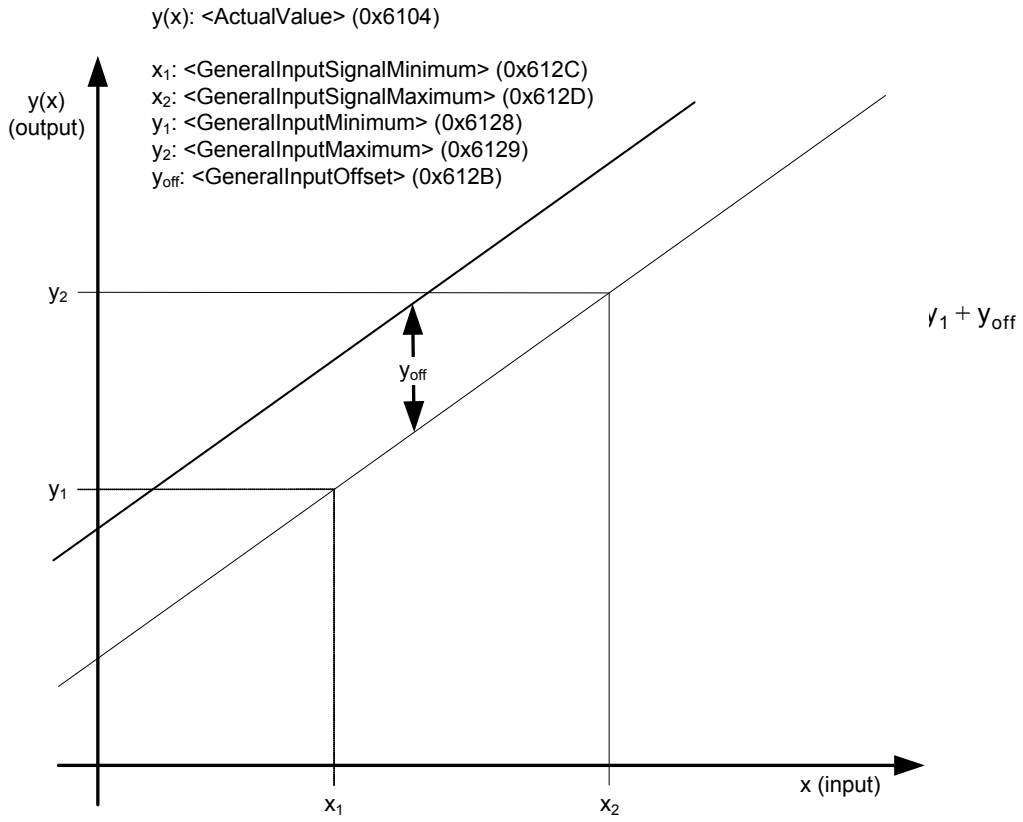


Figure 22: General input scaling

6.3.7.1 Object 0x6128: General input minimum

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

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

6.3.7.2 Object 0x6129: General input maximum

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

Valve_ActualValueConditioning							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6129	1	GeneralInputMaximum	INT16	rw	N	INT16	16384

6.3.7.3 Object 0x612C: General input signal minimum

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

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

6.3.7.4 Object 0x612D: General input signal maximum

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

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

6.3.7.5 Object 0x612B: General input 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	GeneralInputOffset	INT16	rw	N	INT16	None

6.3.8 Parameterization examples

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

6.3.8.1 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).

6.3.8.2 Example 1: Enable/disable transducer interface

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

6.3.8.3 Example 2: Change sign of the transducer signal

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

6.3.8.4 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 <dums16> (0x0003) should be used as input:

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

   Build the address value in the following manner:

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

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

6.3.8.5 Example 4: Adjust transducer interface with 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 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 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 0x00030010 in the parameter <TransducerPort> (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 <PressureOffset> (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: <MinimumTransducerSignal> (0x6124)
   x_2: <MaximumTransducerSignal> (0x6125)
   y_1: <MinimumPressure> (0x6120)
   y_2: <MaximumPressure> (0x6121)

6.4 Analog inputs

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

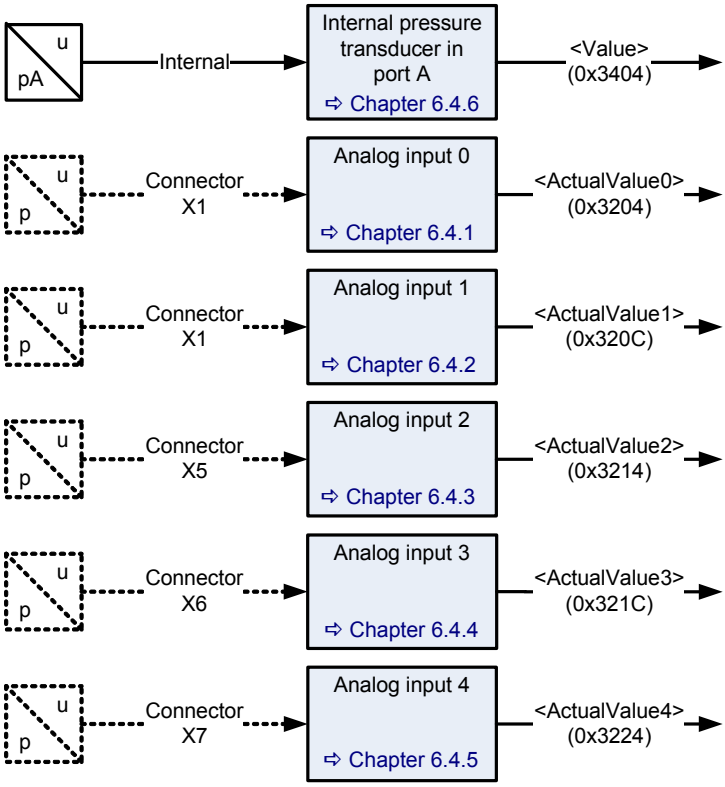


Figure 23: Analog inputs

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

The analog input 0 is intended for the spool position setpoint value and the analog input 1 is intended for the pressure setpoint value. The analog inputs 0 and 1 will only be routed to the demand value generators, if the <DeviceMode> (0x6042) is set to 2 (setpoint input locally). The setpoint values are provided by the field bus, if the <DeviceMode> (0x6042) is set to 1 (setpoint input via bus). In this case the analog inputs 0 and 1 can be used as additional inputs for external transducers.

⇨ Chapter "6.2.3 Spool (stroke ring) position setpoint value path", page 53

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

⇨ Chapter "6.3.5 Transducer interface definition", page 59

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

6.4.1 Analog input 0

6.4.1.1 Object 0x3200: Input type

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

AnalogInput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3200	0	InputType	INT8	rw	Y	INT8	1

Value description

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

Table 22: Possible values of parameter <InputType> (0x3200)

WARNING

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

6.4.1.2 Object 0x3204: Actual value

Actual value of the analog input 0.

AnalogInput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3204	0	ActualValue0	INT16	ro	-	INT16	None

6.4.2 Analog input 1

6.4.2.1 Object 0x3208: Input type

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

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

Value description

⇨ Table 22, page 69

6.4.2.2 Object 0x320C: Actual value

Actual value of the analog input 1.

AnalogInput1							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x320C	0	ActualValue1	INT16	ro	-	INT16	None

6.4.3 Analog input 2

6.4.3.1 Object 0x3210: Input type

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

AnalogInput2							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3210	0	InputType	INT8	rw	Y	INT8	2

Value description

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

Table 23: Possible values of parameter <InputType> (0x3210)

WARNING

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

6.4.3.2 Object 0x3214: Actual value

Actual value of the analog input 2.

AnalogInput2							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3214	0	ActualValue2	INT16	ro	-	INT16	None

Value description

⇨ Table 23, page 70

6.4.4 Analog input 3

6.4.4.1 Object 0x3218: Input type

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

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

Value description

⇨ Table 23, page 70

6.4.4.2 Object 0x321C: Actual value

Actual value of the analog input 3.

AnalogInput3							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x321C	0	ActualValue3	INT16	ro	-	INT16	None

6.4.5 Analog input 4

6.4.5.1 Object 0x3220: Input type

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

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

Value description

⇨ Table 23, page 70

6.4.5.2 Object 0x3224: Actual value

Actual value of the analog input 4.

AnalogInput4							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3224	0	ActualValue4	INT16	ro	-	INT16	None

6.4.6 Internal pressure transducer input

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

⇨ Chapter "6.3.5 Transducer interface definition", page 59

6.4.6.1 Object 0x3404: Actual value

Actual value of the internal pressure transducer input.

PressureTransducer							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3404	0	ActualValue	INT16	ro	-	INT16	None

6.5 Analog outputs

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

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

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

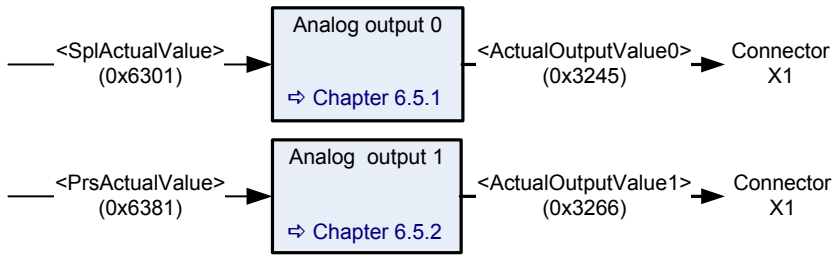


Figure 24: Analog outputs

⇨ Document CA63420-001 "User Manual Electrical Interfaces"

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

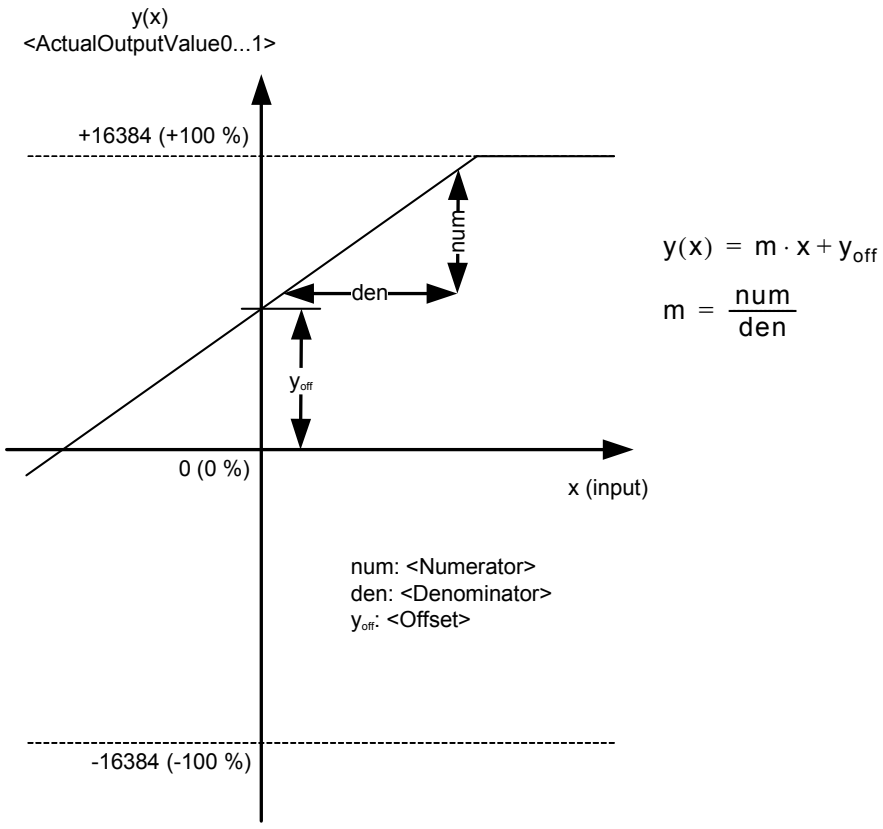


Figure 25: Analog output scaling

6.5.1 Analog output 0

6.5.1.1 Object 0x3244: Scaling

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3244	1	Numerator	INT16	rw	Y	INT16	32760
0x3244	2	Denominator	INT16	rw	Y	INT16	32760
0x3244	3	Offset	INT16	rw	Y	INT16	0

6.5.1.2 Object 0x3245: Actual value

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3245	0	ActualOutputValue0	INT16	ro	-	Depending on <Type> (0x3243)	None

6.5.1.3 Object 0x3240: Parameter

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

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3240	0	Parameter	UINT32	rw	Y	UINT32	0x63010110

Value description

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

Table 24: Possible values of parameter <Parameter> (0x3240)

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

6.5.1.4 Object 0x3243: Type

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

Value description

<Type>	<ActualOutputValue0> (0x3245) range	Output signal range
0	–16384…16384	4…20mA
1	0…16384	4…20mA

Table 25: Possible values of parameter <Type> (0x3243)

6.5.2 Analog output 1

6.5.2.1 Object 0x3265: Scaling

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3265	1	Numerator	INT16	rw	Y	INT16	32760
0x3265	2	Denominator	INT16	rw	Y	INT16	32760
0x3265	3	Offset	INT16	rw	Y	INT16	0

6.5.2.2 Object 0x3266: Actual value

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3266	0	ActualOutputValue1	INT16	ro	-	Depending on <Type> (0x3263)	None

6.5.2.3 Object 0x3260: Parameter

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

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3260	0	Parameter	UINT32	rw	Y	UINT32	0x63810110

Value description

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

Table 26: Possible values of parameter <Parameter> (0x3260)

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

6.5.2.4 Object 0x3263: Type

AnalogOutput0							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3263	0	Type	UINT8	ro	-	0…1	0

Value description

<Type>	<ActualOutputValue1> (0x3266) range	Output signal range
0	–16384…16384	4…20 mA
1	0…16384	4…20 mA

Table 27: Possible values of parameter <Type> (0x3263)

6.6 Digital inputs

The servo valve has one digital input.

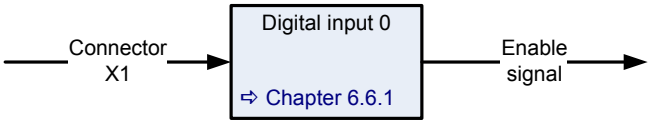


Figure 26: Digital inputs

⇨ Document CA63420-001 "User Manual Electrical Interfaces"

6.6.1 Digital input 0 (enable signal)

The digital enable signal incorporates the following functions:

- Control the device state machine (DSM).
  ⇨ Chapter "5.2.2.4.1 DSM state transitions depending on the enable signal", page 45
- Fault confirmation by generating a rising edge on the digital enable signal.
  ⇨ Chapter "5.2.2.4.2 Fault confirmation with the enable signal", page 45

6.7 Digital outputs

The following digital outputs are available for the servo valves if a 11+PE connector for X1 is used. For Q-valves the digital output 0 is available on pin 8. p/Q-valves are using this pin for the analog output (actual pressure). The digital output 1 is available on pin 11.

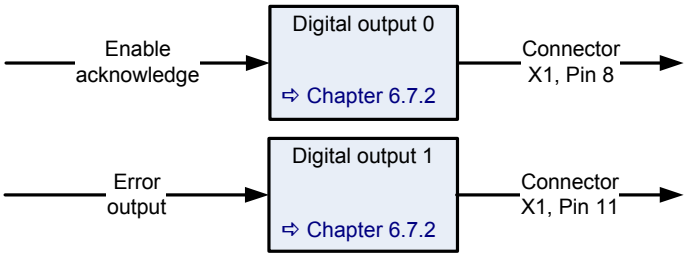


Figure 27: Digital outputs in the default configuration

⇨ Document CA63420-001 "User Manual Electrical Interfaces"

6.7.1 Object 0x5E42: Digital output value

The <DigitalOutputValue> controls the state of the digital outputs in case the <DigitalOutputType> = 0.

ValveDigitalOutputType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5E42	1	DigitalOutputValue0	INT8	rw	-	0…1	0
0x5E42	2	DigitalOutputValue1	INT8	rw	-	0…1	0

6.7.2 Object 0x5E41: Digital output type

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

ValveDigitalOutputType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x5E41	1	DigitalOutputType0	INT8	ro	-	0…4	3
0x5E41	2	DigitalOutputType1	INT8	ro	-	0…4	2

Value description

<DigitalOutput1Type>	Description
0	The servo valve's digital output 1 is controlled by the parameter <DigitalOutputValue> (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 "7.12.3 Failsafe monitoring", page 137
2	Error output pin. The servo valve's digital output acts according to the Device Profile Fluid Power. This means it is controlled by the device state machine (fault indication). In this case the digital output is used to indicate fault states (negative logic). ⇨ Chapter "5.2.2.5 Error output pin", page 45
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 "7.12 Monitoring", page 133

Table 28: Possible values of parameter <DigitalOutputType> (0x5E41)

WARNING

The word "failsafe" means not a personnel safety. If a personnel safe servo valve is needed, some additional electrical and hydraulic parts are necessary.

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

6.7.2.1 Object 0x2420: Digital output 1 type

The <DigitalOutput1Type> (0x2420) is a copy of the <DigitalOutputType1> (0x5E41) and is used to be compatible to old firmware versions.

ValveDigitalOutputType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x2420	0	DigitalOutput1Type	INT8	ro	-	0…4	2

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	Persis-tence	Value range	Default
0x0002	0	Integer08	INT8	rw	N	INT8	None

6.8.2 Object 0x0003: Signed two byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	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	Persis-tence	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	Persis-tence	Value range	Default
0x0005	0	Unsigned08	UINT8	rw	N	UINT8	None

6.8.5 Object 0x0006: Unsigned two byte integer

DataType							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	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	Persis-tence	Value range	Default
0x0007	0	Unsigned32	UINT32	rw	N	UINT32	None

6.8.7 Object 0x0008: Float32

Eventhandler							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x0008	0	Float32	FLOAT32	rw	N	FLOAT32	None

6.8.8 Object 0x0009: Visible string

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

For your notes.

7 Servo valve (pump) functions

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

- Controller (spool position and pressure controller)
- Monitoring functions
- Command signal conditioning (spool position and pressure demand value generator)

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

⇨ Chapter "7.1 Control modes", page 82

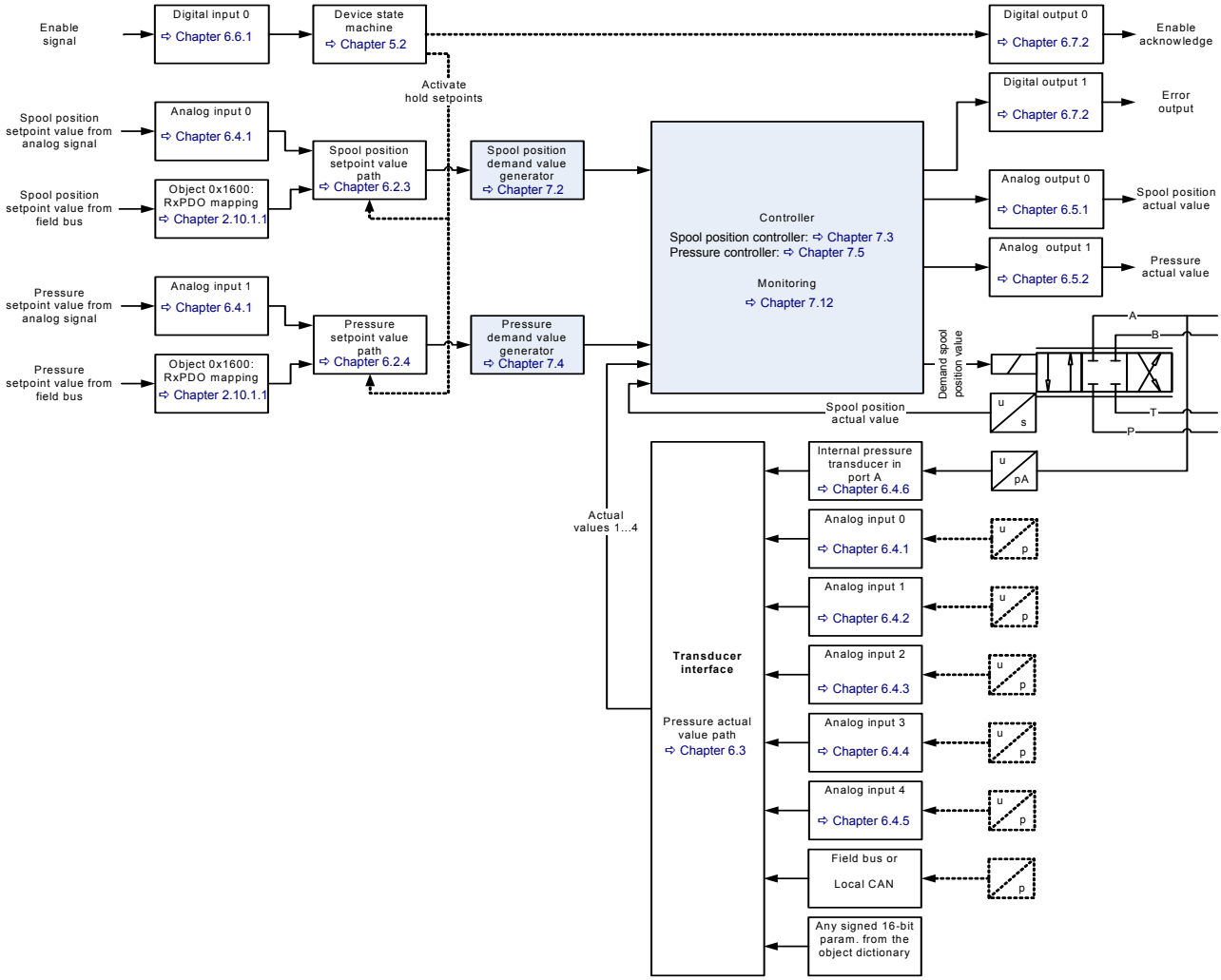


Figure 28: Servo valve controller and command signal conditioning

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

- Spool position controller (configured by Moog)
  ⇨ Chapter "7.3 Stroke ring (spool) position controller", page 96
- Pressure controller (configured by user)
  ⇨ Chapter "7.5 Pressure controller", page 110
- Pressure demand signal polarity
- Spool position (Q) / pressure (P) switchover

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

7.1 Control modes

The servo valve can be run in the following control modes. The control mode of the servo valve is set with the parameter <ControlMode> (0x6043).

<ControlMode>	Meaning
1	Spool position control open loop Used for tests ⇨ Chapter "7.1.3 Spool position control open loop", page 83
2	Spool position control closed loop Spool position control ⇨ Chapter "7.1.4 Spool position control closed loop", page 84
3	Pressure control open loop Used for tests. Behaves like a closed loop Q control. ⇨ Chapter "7.1.5 Pressure control open loop", page 84
4	Pressure control closed loop Pressure / force control ⇨ Chapter "7.1.6 Pressure control closed loop", page 85
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 85

Table 29: Control mode values

7.1.1 Object 0x6043: Control mode

This parameter selects the servo valve control mode.

The available control modes are defined by the <Capability> (0x605F) parameter.
⇨ Chapter "4.2.9 Object 0x605F: Capability", page 36

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

Value description

<ControlMode>	Meaning
1	Spool position control open loop
2	Spool position control closed loop
3	Pressure control open loop
4	Pressure control closed loop
5	p/Q control

Table 30: Possible values of parameter <ControlMode> (0x6043)

Power limitation

The control modes 2, 3, 4 and 5 work parallel to the power limitation.

If the power limitation is active (enabled with control word bit 11), then the lowest controller output is forwarded to the pump.

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	Persis-tence	Value range	Default
0x4043	0	ControlModeDefault	INT8	rw	Y	1…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).

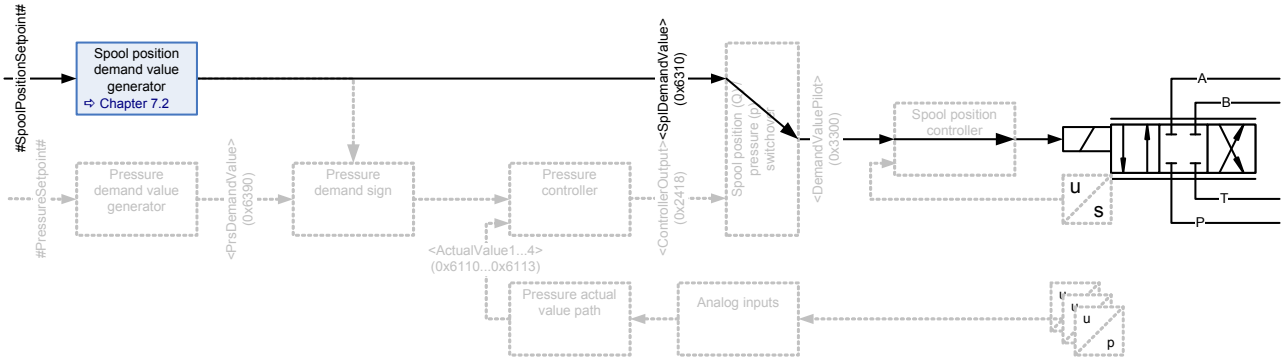


Figure 29: Spool position control open loop

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 82
⇨ Chapter "7.2.1 Object 0x6310: Demand value", page 86

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

The signal #SpoolpositionSetpoint# is an internal signal only. It links the signal from the spool position setpoint value path to the spool position demand value generator.
⇨ Chapter "6.2.3 Spool (stroke ring) position setpoint value path", page 53

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).

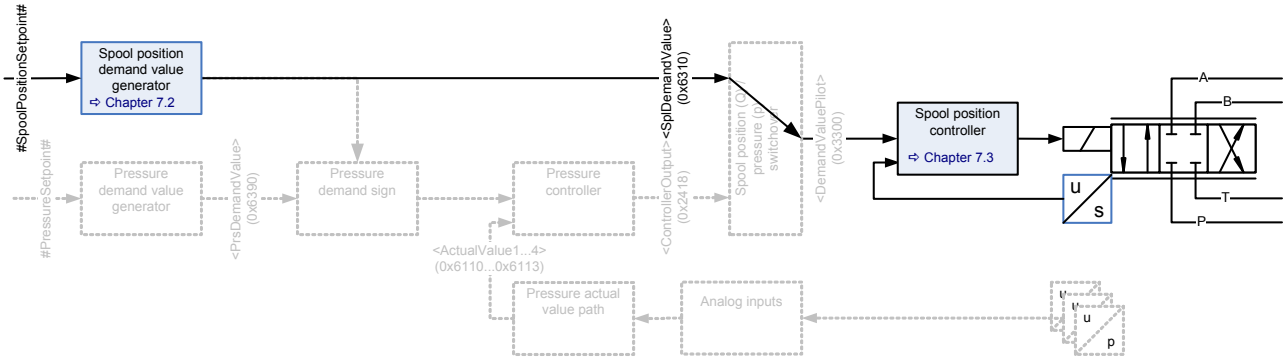


Figure 30: Spool position control closed loop

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 82
⇨ Chapter "7.2.1 Object 0x6310: Demand value", page 86

The signal #SpoolpositionSetpoint# is an internal signal only. It links the signal from the spool position setpoint value path to the spool position demand value generator.
⇨ Chapter "6.2.3 Spool (stroke ring) position setpoint value path", page 53

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).

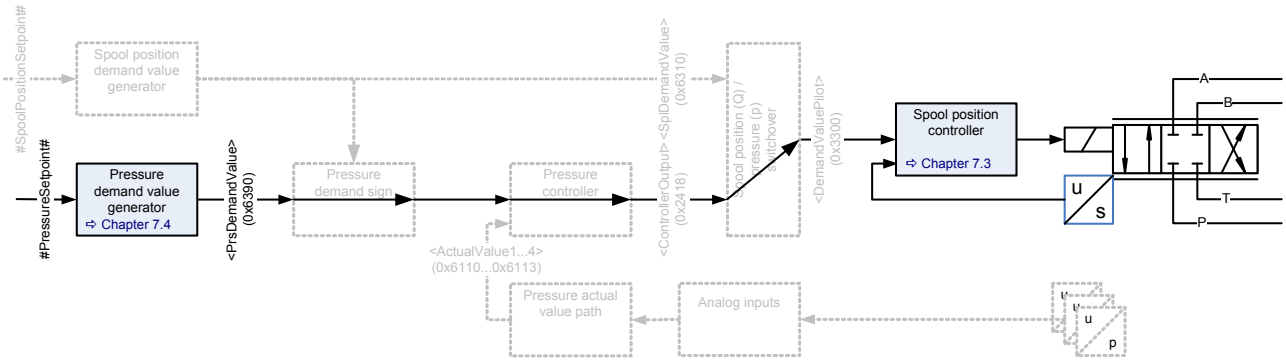


Figure 31: Pressure control open loop

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 82
⇨ Chapter "7.2.1 Object 0x6310: Demand value", page 86

The behavior is the same as in the control mode "Spool position closed loop" except it uses the pressure setpoint value as the input signal.

The signal #PressureSetpoint# is an internal signal only. It links the signal from the pressure setpoint value path to the pressure demand value generator.
⇨ Chapter "6.2.4 Pressure setpoint value path", page 55

WARNING

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).

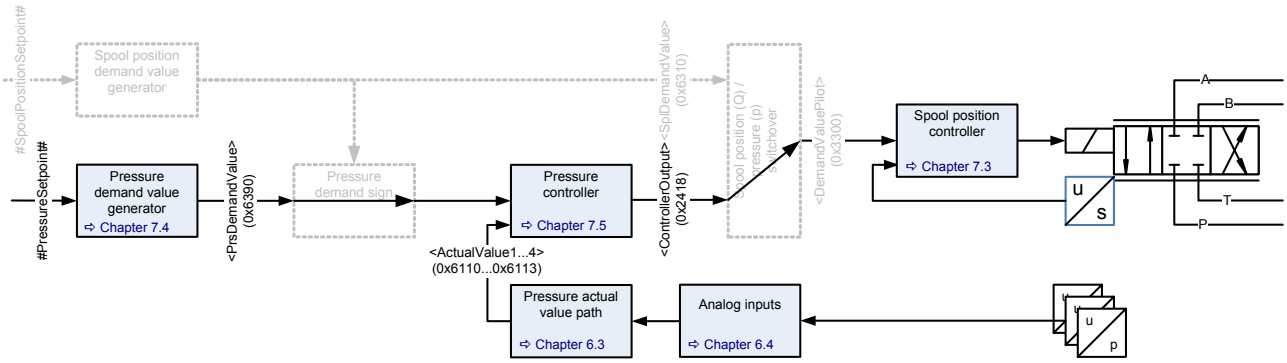


Figure 32: Pressure control closed loop

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 82
⇨ Chapter "7.4.1 Object 0x6390: Demand value", page 101
⇨ Chapter "6.3 Actual value transducer interface", page 57

The signal #PressureSetpoint# is an internal signal only. It links the signal from the pressure set-point value path to the pressure demand value generator.
⇨ Chapter "6.2.4 Pressure setpoint value path", page 55

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).

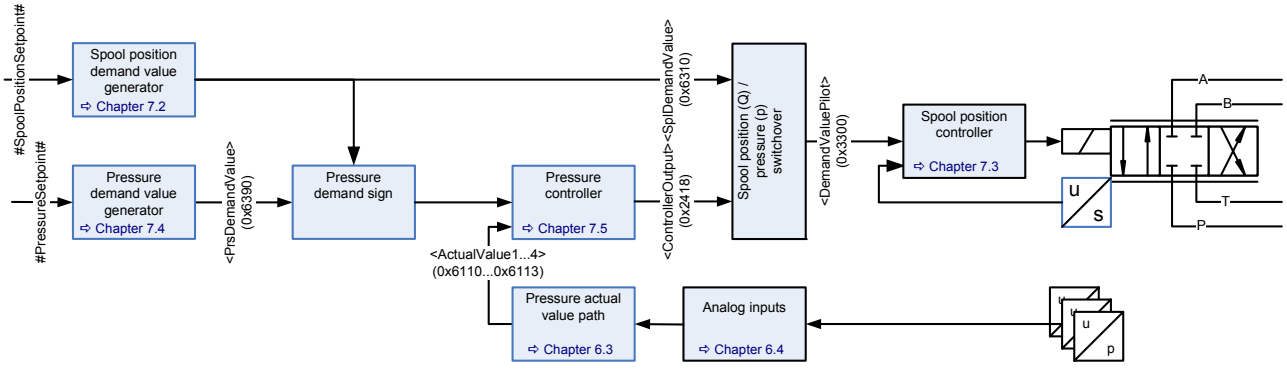


Figure 33: p/Q control closed loop

⇨ Chapter "7.2.1 Object 0x6310: Demand value", page 86
⇨ Chapter "7.4.1 Object 0x6390: Demand value", page 101
⇨ Chapter "6.3 Actual value transducer interface", page 57

The signal #PressureSetpoint# is an internal signal only. It links the signal from the pressure set-point value path to the pressure demand value generator.
⇨ Chapter "6.2.4 Pressure setpoint value path", page 55

7.2 Spool position setpoint conditioning / demand value generator

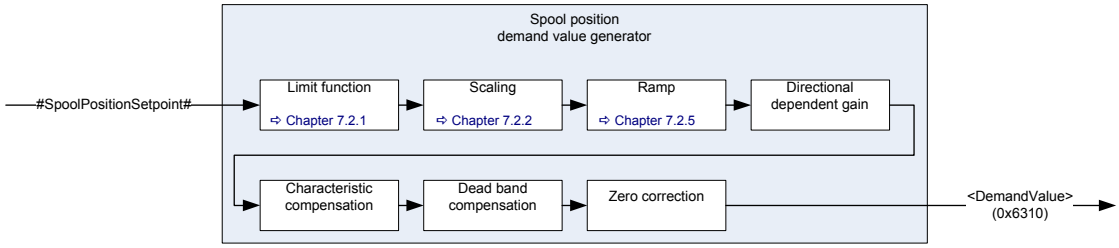


Figure 34: Spool position demand value generator

The internal signal #SpoolPositionSetpoint# is used to link the spool position setpoint value to the spool position demand value generator.
⇨ Chapter "6.2.3 Spool (stroke ring) position setpoint value path", page 53

7.2.1 Object 0x6310: Demand value

The demand value is generated from the #SpoolPositionSetpoint# by means of the functions in the demand value generator and forwarded to the spool position controller.

ValvePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6310	1	SplDemandValue	INT16	ro	-	INT16	None
0x6310	2	Unit	UINT8	ro	-	UINT8	0
0x6310	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.2.2 Object 0x6311: Reference value

The reference value is the value that corresponds to 100 % of the input signal. This means that a 100 % input signal is equal to 16384 increments and a –100 % input signal is equal to –16384 increments. This parameter depends on the controller hardware. It can be used by the field bus master to scale the setpoint values.

ValvePositionControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6311	1	SplReferenceValue	INT16	ro	-	INT16	16384
0x6311	2	Unit	UINT8	ro	-	UINT8	0
0x6311	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.2.3 Limit function

This function limits the value range of the spool position input signal. The limit is defined by setting the upper and lower limit.

Bit 10 of the status word indicates whether the input signal is being limited or not.
⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46

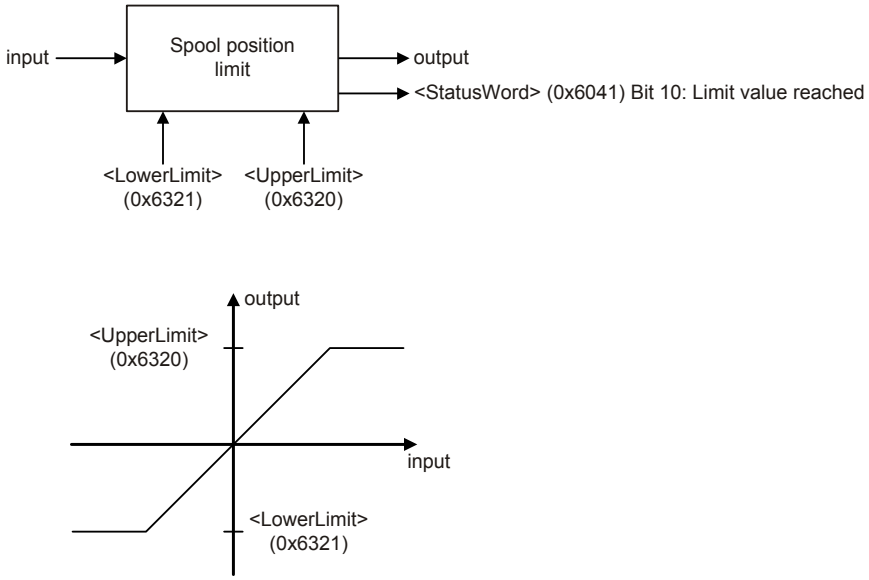


Figure 35: Limit function

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: Upper Limit

ValvePositionControl_DemandValueGenerator_Limit							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6320	1	UpperLimit	INT16	rw	Y	<LowerLimit> (0x6321)…32767	32760
0x6320	2	Unit	UINT8	ro	-	UINT8	0
0x6320	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.2.3.2 Object 0x6321: Lower Limit

ValvePositionControl_DemandValueGenerator_Limit							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6321	1	LowerLimit	INT16	rw	Y	–32768… <UpperLimit> (0x6320)	–32760
0x6321	2	Unit	UINT8	ro	-	UINT8	0
0x6321	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.2.4 Scaling

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

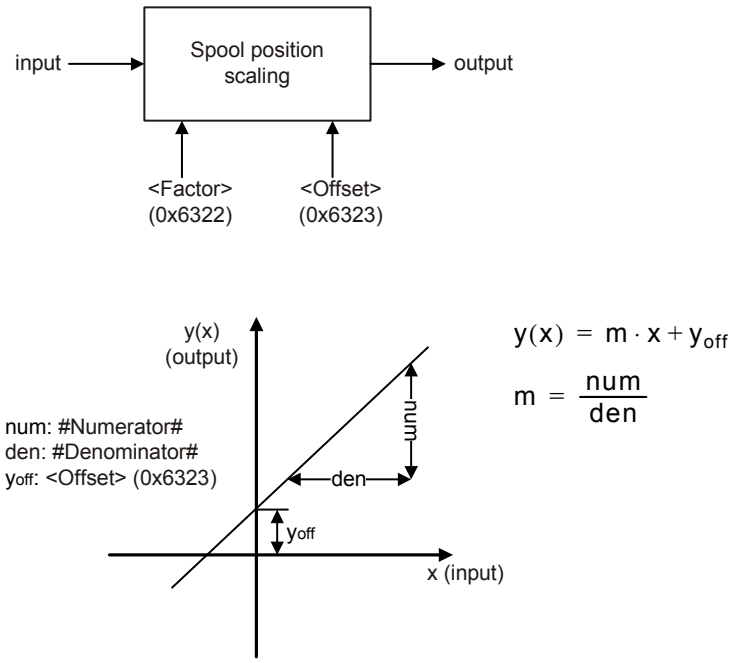


Figure 36: Scaling function

7.2.4.1 Object 0x6322: Factor

This parameter is a slope factor by which the input is multiplied. It is defined by two signed integer values, the numerator (upper 16 bits of the parameter) and the denominator (lower 16 bits of the parameter).

ValvePositionControl_DemandValueGenerator_Scaling							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6322	0	Factor	UINT32	rw	Y	UINT32	0x00010001

Value description

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

Table 31: Data structure of the slope factor

7.2.4.2 Object 0x6323: Offset

This parameter is the offset of the linear output function.

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

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.2.5 Ramp

The ramp function limits the slew rate of the input signal. The <Type> (0x6330) parameter is used to activate a one-quadrant, two-quadrant or four-quadrant ramp or to deactivate the ramp function.

If the ramp function is running or the ramp function is stopped by the bit 15 (ramp stop) of the #ControlWord# signal the following <StatusWord> (0x6041) bits are set:

<StatusWord> (0x6041) bit	Description
9	This bit is set if the following conditions are true: Spool position and/or pressure ramp function is active and spool position and/or pressure function is running and #ControlWord# bit 15 is set to false.
15	This bit is set if the following conditions are true: Spool position and/or pressure ramp function is active and #ControlWord# bit 15 is set to true.

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46
⇨ Chapter "5.1.1 Object 0x604F: Local", page 37
⇨ Chapter "5.1.2 Object 0x6040: Control word", page 38
⇨ Chapter "5.1.3 Object 0x4040: Local control word", page 39

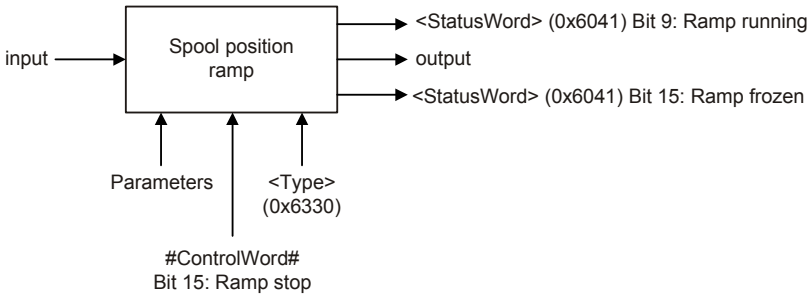


Figure 37: Ramp function

7.2.5.1 Object 0x6330: Type

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

ValvePositionControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6330	0	Type	INT8	rw	Y	0…3	0

Value description

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

Table 32: Possible values of parameter <Type> (0x6330)

7.2.5.2 One-quadrant ramp (ramp type 1)

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

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

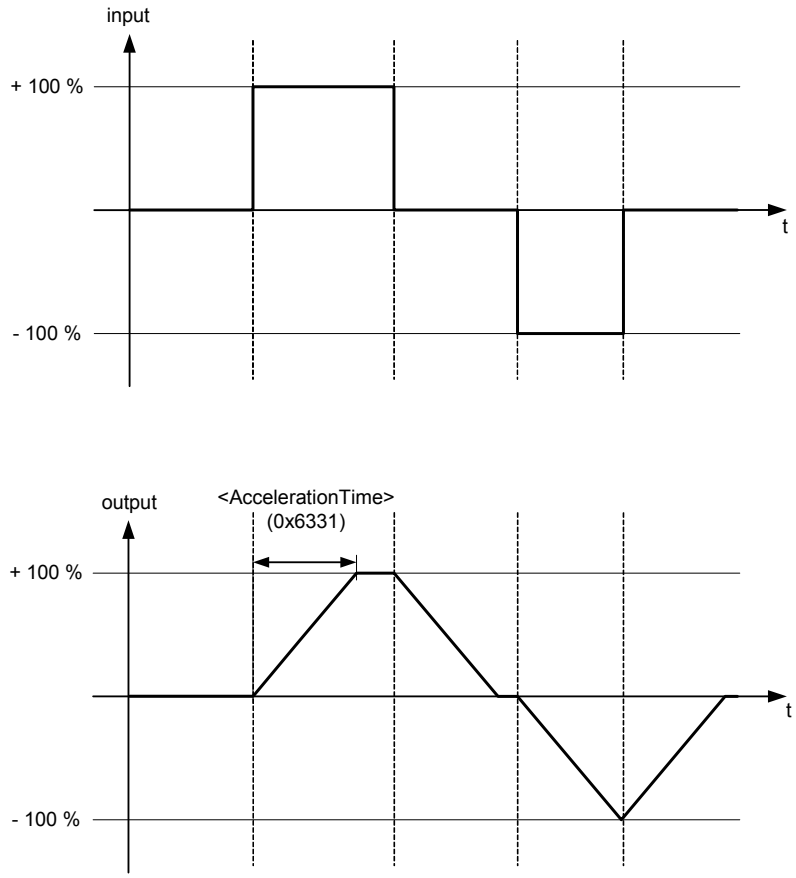


Figure 38: Ramp type 1

7.2.5.2.1 Object 0x6331: Acceleration time

This parameter defines the output signal's maximum rate of change. The acceleration time corresponds to the time the signal needs for a change from 0 to 100 % as shown in ⇨ Figure 38, page 90. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

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

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.2.5.3 Two-quadrant ramp (ramp type 2)

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

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

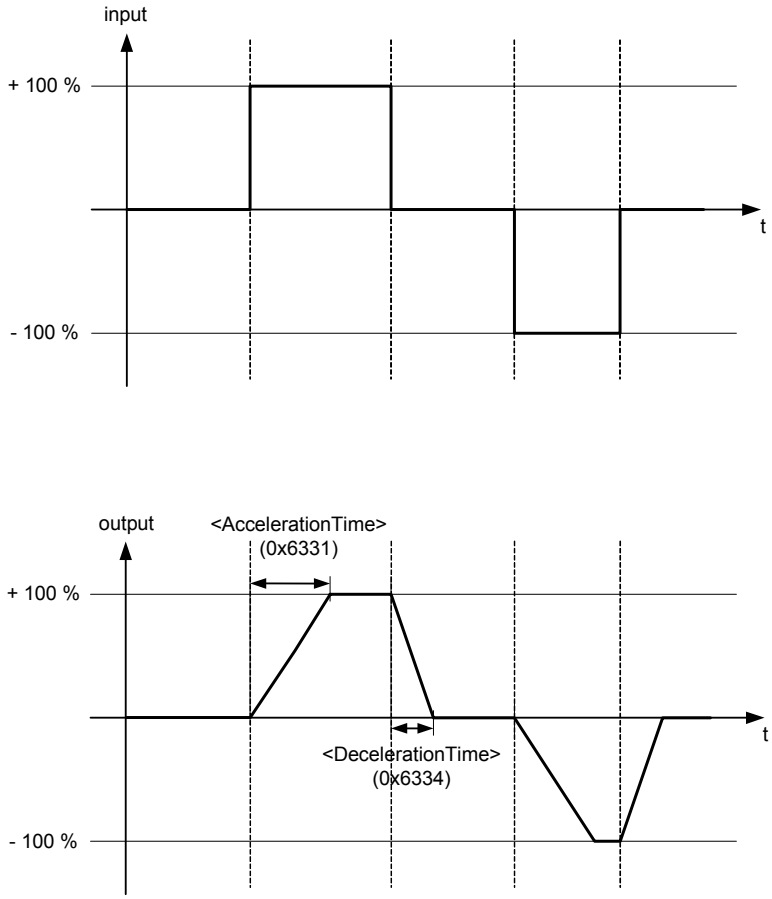


Figure 39: Ramp type 2

7.2.5.3.1 Object 0x6331: Acceleration time

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

⇨ Chapter "7.2.5.2.1 Object 0x6331: Acceleration time", page 91

7.2.5.3.2 Object 0x6334: Deceleration time

This parameter defines the output signal's maximum rate of change. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6334	1	DecelerationTime	UINT16	rw	Y	UINT16	0
0x6334	2	Unit	UINT8	ro	-	UINT8	3
0x6334	3	DecelerationTime_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

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.

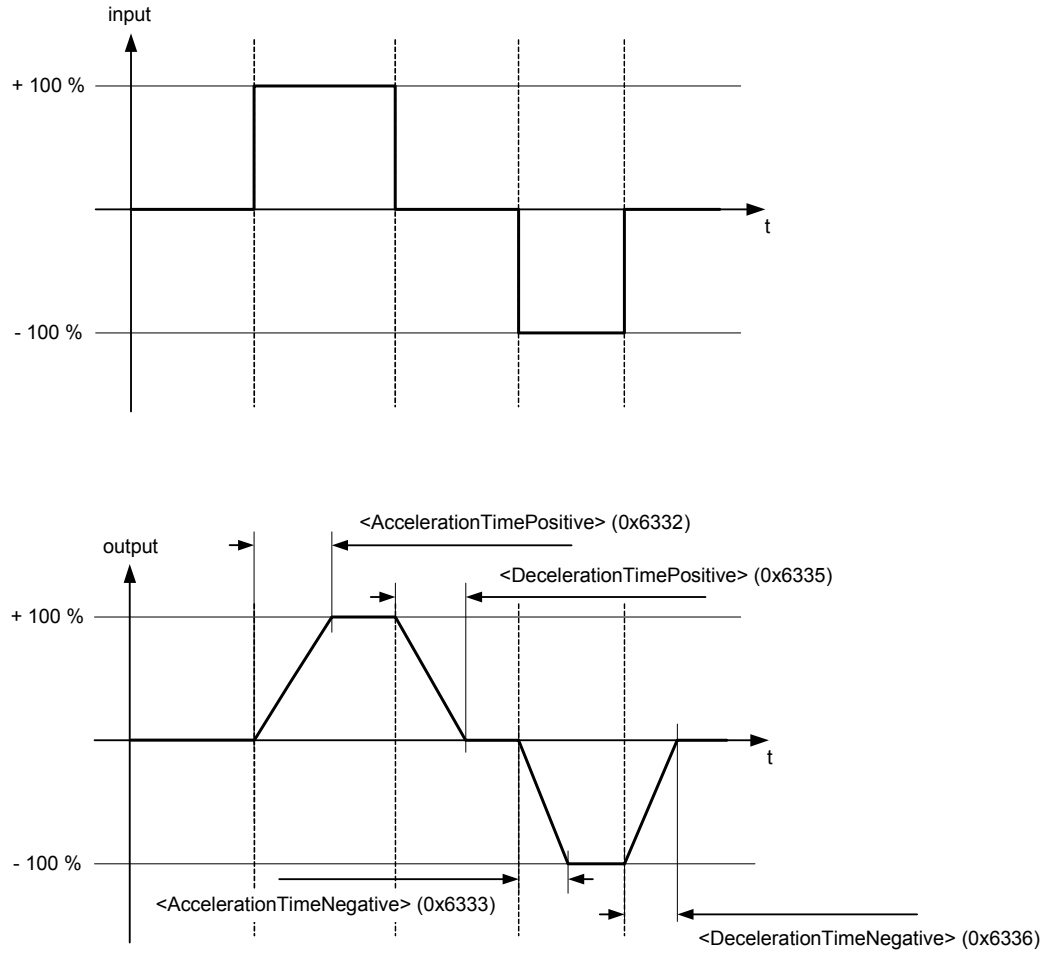


Figure 40: Ramp type 3

7.2.5.4.1 Object 0x6332: Acceleration time positive

This parameter defines the output signal's maximum rate of change on the positive side. The acceleration time corresponds to the time that the signal needs for a change from 0 to 100 %. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6332	1	AccelerationTimePositive	UINT16	rw	Y	UINT16	0
0x6332	2	Unit	UINT8	ro	-	UINT8	3
0x6332	3	AccelerationTimePositive_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.2.5.4.2 Object 0x6333: Acceleration time negative

This parameter defines the output signal's maximum rate of change on the negative side. The acceleration time corresponds to the time that the signal needs for a change from 0 to 100 %. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6333	1	AccelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x6333	2	Unit	UINT8	ro	-	UINT8	3
0x6333	3	AccelerationTimeNegative_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.2.5.4.3 Object 0x6335: Deceleration time positive

This parameter defines the output signal's maximum rate of change on the positive side. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6335	1	DecelerationTimePositive	UINT16	rw	Y	UINT16	0
0x6335	2	Unit	UINT8	ro	-	UINT8	3
0x6335	3	DecelerationTimePositive_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.2.5.4.4 Object 0x6336: Deceleration time negative

This parameter defines the output signal's maximum rate of change on the negative side. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePositionControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6336	1	DecelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x6336	2	Unit	UINT8	ro	-	UINT8	3
0x6336	3	DecelerationTimeNegative_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.2.6 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.

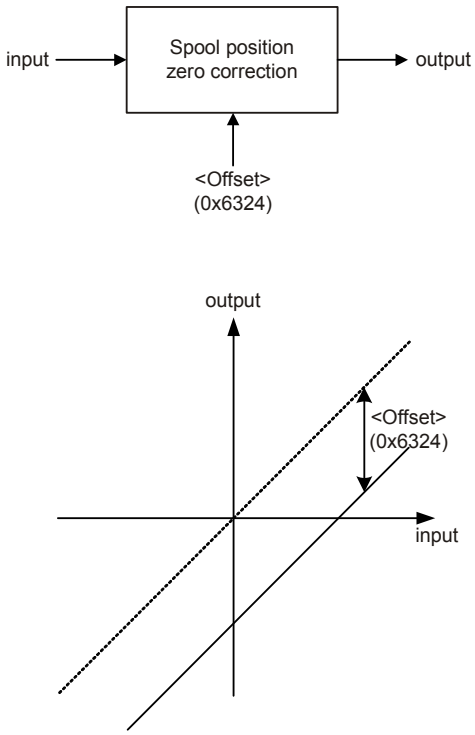


Figure 41: Zero correction

7.2.6.1 Object 0x6324: Offset

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

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.2.7 Hybrid mode correction

In the hybrid mode, a constant pump (typically not connected to the bus) and a solo pump operate to the same volume. This functionality requires the knowledge of the constant pump's flow volume. This flow volume (volume of the hybrid pump) is set with the parameter <HybridFlow> (0x2147) and is given as percentage of the solo pump maximum volume.

The hybrid adjustment is done within the demand value generator before the position controller. The transfer behavior is as follows:

$$\text{output} = \text{input} \cdot (16384 + \text{hybridflow}) - \text{hybridflow}$$

Figure 42 shows the transfer behavior for the hybrid mode and also the solo mode for comparison purposes. The calculation of the flow demand for the servo pump (servo demand) is as follows:

$$\text{servo demand} = \text{total demand} \cdot \left(1 + \frac{\text{flow constant pump}}{16384}\right) - \text{flow constant pump}$$

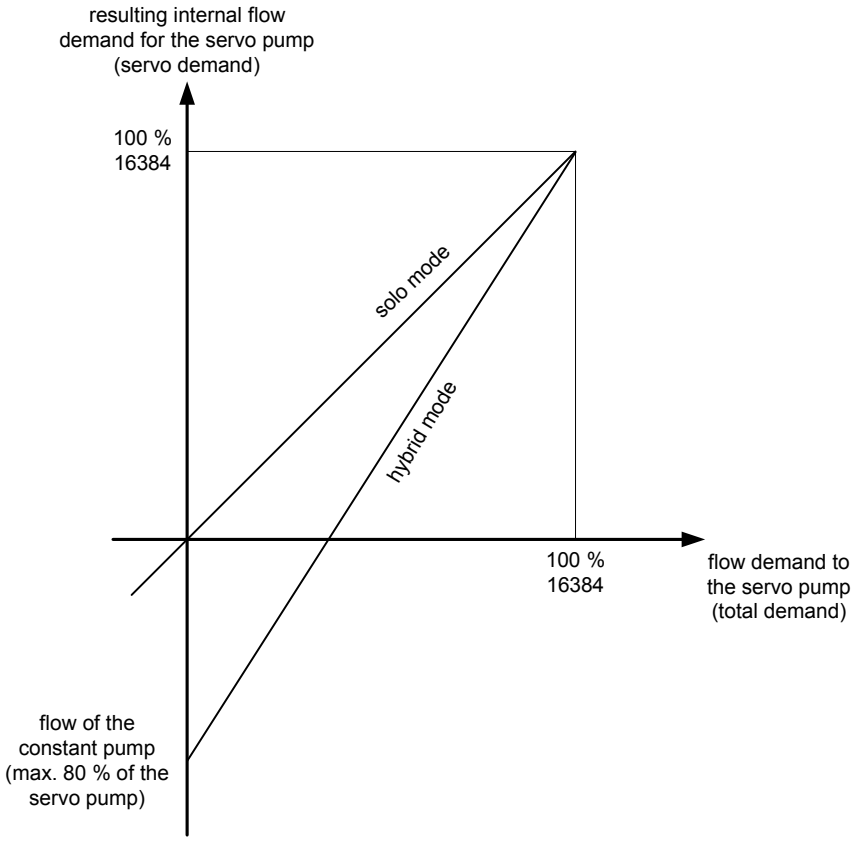


Figure 42: Transfer behavior hybrid mode / solo mode

Enabling hybrid operation

The hybrid mode is selected through the parameter <Mode> (0x2148). Writing the value 1 to the parameter activates the hybrid mode. 0 means hybrid mode is deactivated. The hybrid mode can be activated/deactivated for every parameter set by writing the corresponding value (0 or 1) to the subindexes (1…16) of 0x2148.

Description of parameter set switching:
⇨ Chapter "7.11 Analog parameter set switching", page 130

7.2.7.1 Object 0x2147: Hybrid flow

This parameter holds the flow of the hybrid pump (constant pump). The unit is relative to the nominal flow of the servo pump.

Example:
The servo pump has a nominal flow of 140 l/min. This corresponds to a set signal of 16384. The constant pump has a flow of 80 l/min. On the scale of the servo pump this corresponds to:

$$16384 \cdot \frac{80}{140} = 9362.3$$

AnalogParameterSetSwitching							
Index	Subindex	Name	Data type	Access	Persistence	Value range	Default
0x2147	0	HybridFlow	INT16	rw	Y	INT16	DSV

7.3 Stroke ring (spool) position controller

The spool position controller controls the stroke ring position. The parameters are set up by Moog during production. Only for the sake of completeness will some details be explained in this section.

The spool position controller is configured in the factory and cannot be changed by the user.

7.3.1 Pilot and main stage on RKP-D

Control structure of the stroke ring position. On the RKP-D the main stage is controlling the position of the stroke ring.

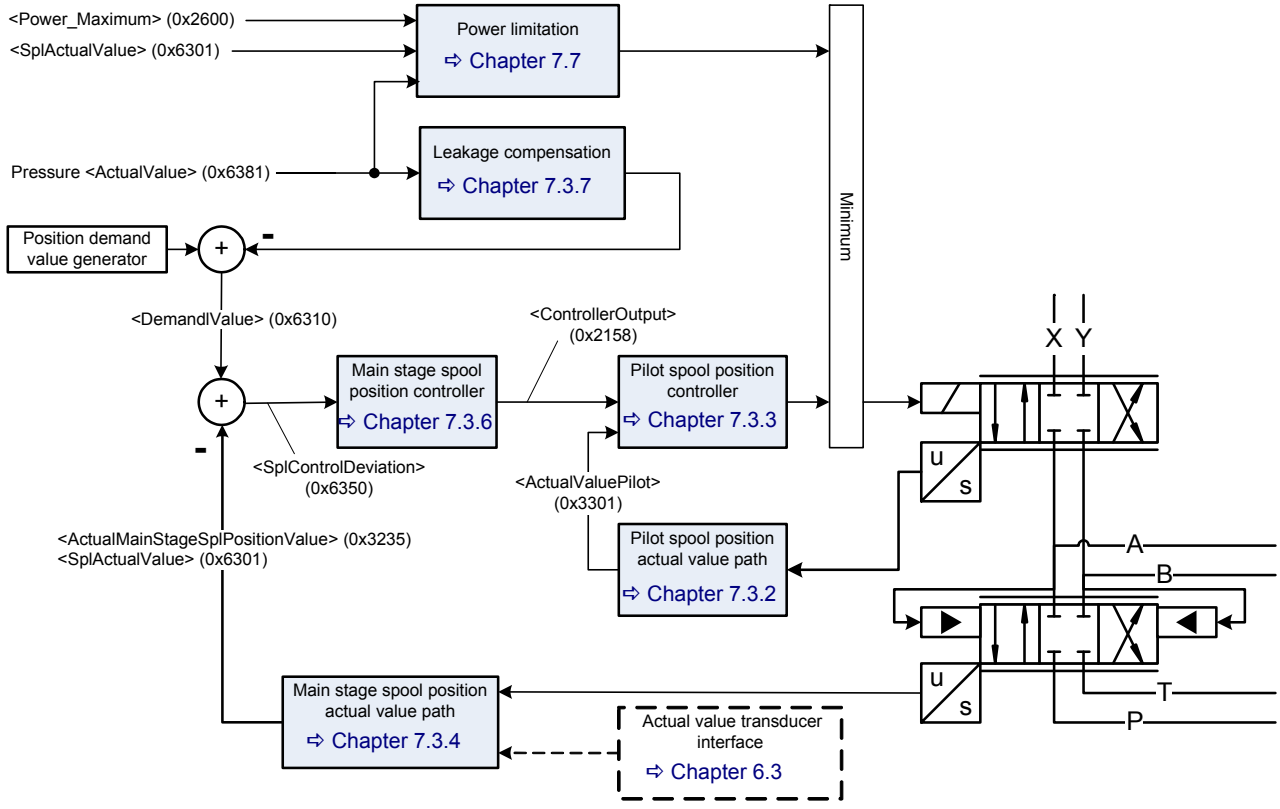


Figure 43: Position control closed loop

7.3.2 Stroke ring position actual value path

7.3.2.1 Object 0x6301: Actual value

This parameter holds the actual value of the spool position.

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

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.3.2.2 Object 0x3301: Actual value pilot

The spool position value of the pilot stage is scaled and mapped to the same signal <ActualValuePilot> (0x3301).

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

7.3.2.3 Object 0x3506: Customer Scaling Offset

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

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3506	0	CustomerScalingOffset	INT16	rw	Y	–819…819	0

7.3.3 Pilot spool position controller

The spool position controller is used to control the pilots spool position. The main stage is controlled by the main stage spool position controller.

7.3.3.1 Object 0x6350: Control deviation

The control deviation is the difference between the setpoint value and the actual value of the stroke ring position.

ValvePositionControl							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6350	1	SplControlDeviation	INT16	ro	-	INT16	None
0x6350	2	Unit	UINT8	ro	-	UINT8	0
0x6350	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.3.3.2 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	Persis-tence	Value range	Default
0x241F	0	CustomerOverallGain	FLOAT32	rw	Y	0.0…2.0	1.0

7.3.4 Stroke ring position actual value path

7.3.4.1 Object 0x3237#1…3: Customer scaling

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

$$\text{ActualMainStageSplPosVal (0x3235)} = \frac{(\text{Input} + \text{CustomerScalingOffset (0x3237\#3)}) \times \text{CustomerScalingFactorNumerator (0x3237\#1)}}{\text{CustomerScalingFactorDenominator (0x3237\#2)}}$$

ExternalLVDT							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3237	1	CustomerScalingFactorNumerator	INT16	rw	Y	INT16	16386
0x3237	2	CustomerScalingFactorDenominator	INT16	rw	Y	INT16	16386
0x3237	3	CustomerScalingOffset	INT16	rw	Y	INT16	0

7.3.4.2 Object 0x3235: Actual value

This parameter holds the the stroke ring position (main stage spool position).

ExternalLVDT							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x3235	0	ActualMainStageSplPositionValue	INT16	ro	-	INT16	None

7.3.5 Stroke ring position transducer selection

For applications with an external main stage position signal you can use a transducer interface to root any actual main stage spool postionen signal to the main stage spool position controller.

7.3.5.1 Object 0x2149: Active transducer interface main stage

This parameter holds the interface number for the main stage spool position actual value signal rooted to the controller. If <ActiveTransducerInterfaceMainStage> (0x2149) = 0 then <SplActualValue> (0x6301) is used, otherwise the output of the interface number <ActiveTransducerInterfaceMainStage> (0x2149).

ValveMainStageControl							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x2149	0	ActiveTransducerInterfaceMainStage	UINT8	rw	Y	0…8	None

7.3.6 Stroke ring position controller

7.3.6.1 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	Persis-tence	Value range	Default
0x215C	0	MainStageCustomerOverallGain	FLOAT32	rw	Y	0.0…2.0	1.0

7.3.6.2 Object 0x2158: Controller output

This parameter holds the main spool position controller output.

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

7.3.7 Leakage compensation

Background
As the pressure increases, a hydraulic system will produce increasing internal leakage that will be missing from the usable volume flow that should be normally generated by the pump. The pump electronics contain a leakage-compensation to cancel out this effect.

The leakage value set with the parameter <LeakageCompensation> (0x2120) expresses how much leakage the pump has depending on the supply pressure. The unit is %eccentricity/bar.
⇨ Chapter "7.3.7.1 Object 0x2120: Leakage compensation", page 100

Affected values (parameters)
The leakage compensation affects the demand value to the position controller as shown in the following figure.

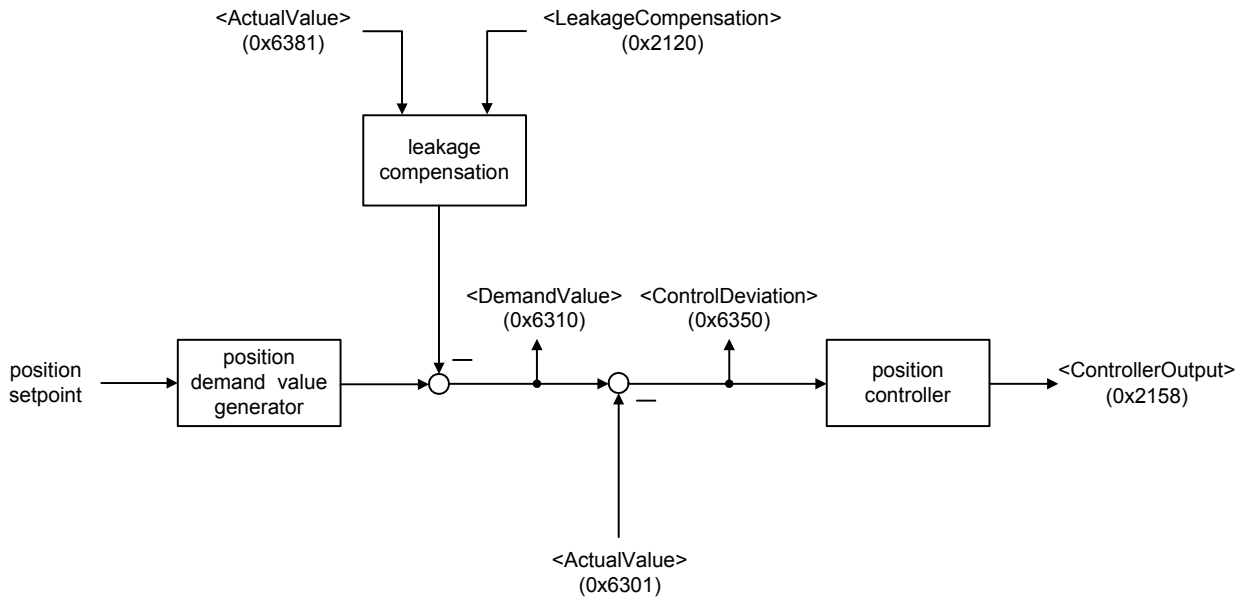


Figure 44: Leakage compensation

Enabling of the leakage compensation function
The leakage compensation function is switched on and off with the bit 10 in the control word. If bit 10 is set to true, leakage compensation is enabled.
⇨ Chapter "5.1.2 Object 0x6040: Control word", page 38

7.3.7.1 Object 0x2120: Leakage compensation

This parameter is used to set the leakage compensation value as described in the previous section. The unit of the leakage compensation is %eccentricity/bar.

PumpController							
Index	Subindex	Name	Data type	Access	Persis-tence	Value range	Default
0x2120	0	LeakageCompensation	FLOAT32	rw	Y	FLOAT32	DSV

7.4 Pressure setpoint conditioning / demand value generator

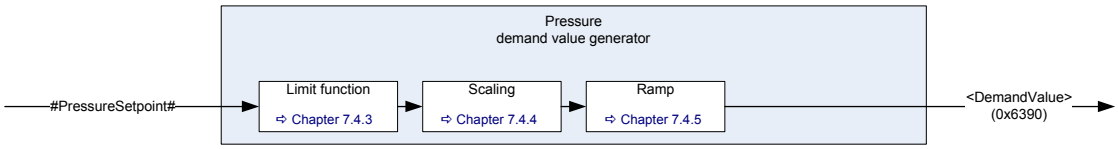


Figure 45: Pressure demand value generator

The internal signal #PressureSetpoint# is used to link the pressure setpoint value to the pressure demand value generator.
⇨ Chapter "6.2.4 Pressure setpoint value path", page 55

7.4.1 Object 0x6390: Demand value

The demand value indicated by this parameter is generated from the #PressureSetpoint# by means of the functions in the demand value generator and forwarded to the pressure controller.

ValvePressureControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6390	1	PrsDemandValue	INT16	ro	N	INT16	None
0x6390	2	Unit	UINT8	ro	-	UINT8	0
0x6390	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.4.2 Object 0x6391: Reference value

The reference value is the value that corresponds to 100 % of the input signal. This means that a 100 % pressure input signal is equal to 16384 increments and a –100 % input signal is equal to –16384 increments. This parameter depends on the controller hardware. It can be used by the field bus master to scale the setpoint values.

ValvePressureControl_DemandValueGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x6391	1	PrsReferenceValue	INT16	ro	-	INT16	16384
0x6391	2	Unit	UINT8	ro	-	UINT8	0
0x6391	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.4.3 Limit function

This function limits the value range of the input signal. The limit is defined by setting the upper limit and lower limit parameters.

Bit 10 of the status word indicates whether the input signal is being limited by this function or not.
⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46

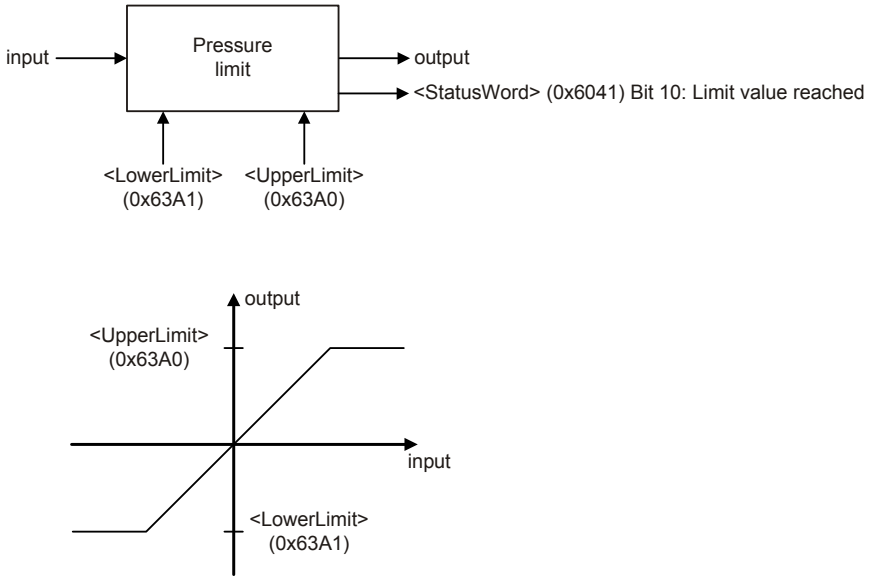


Figure 46: Limit function

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

7.4.3.1 Object 0x63A0: Upper Limit

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

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.4.3.2 Object 0x63A1: Lower Limit

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

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

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.

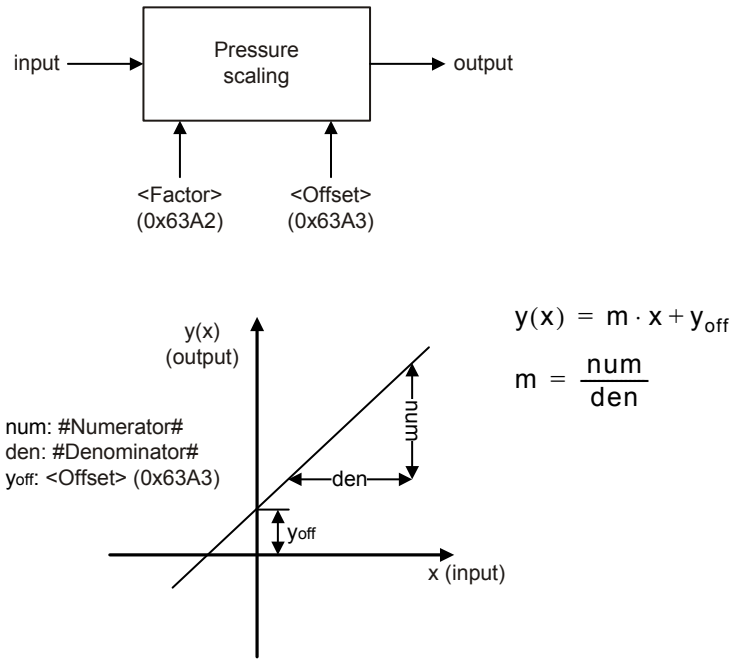


Figure 47: Scaling function

7.4.4.1 Object 0x63A2: Factor

This parameter is a slope factor by which the input is multiplied. It is defined by two signed integer values, the numerator (upper 16 bits of the parameter) and the denominator (lower 16 bits of the parameter).

ValvePressureControl_DemandValueGenerator_Scaling							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x63A2	0	Factor	UINT32	rw	Y	UINT32	0x00010001

Value description

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

Table 33: Data structure of the slope factor

7.4.4.2 Object 0x63A3: Offset

This parameter is the offset of the linear output function.

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

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.4.5 Ramp

The ramp function limits the slew rate of the input signal. The <Type> (0x63B0) parameter is used to activate a one-quadrant, two-quadrant or four-quadrant ramp or to deactivate the ramp function.

If the ramp function is running or the ramp function is stopped by the bit 15 (ramp stop) of the #ControlWord# signal the following <StatusWord> (0x6041) bits are set:

<StatusWord> (0x6041) bit	Description
9	This bit is set if the following conditions are true: Spool position and/or pressure ramp function is active and spool position and/or pressure function is running and #ControlWord# bit 15 is set to false.
15	This bit is set if the following conditions are true: Spool position and/or pressure ramp function is active and #ControlWord# bit 15 is set to true

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46
⇨ Chapter "5.1.1 Object 0x604F: Local", page 37
⇨ Chapter "5.1.2 Object 0x6040: Control word", page 38
⇨ Chapter "5.1.3 Object 0x4040: Local control word", page 39

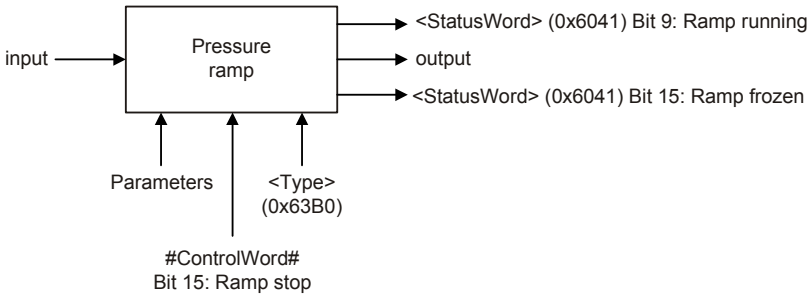


Figure 48: Ramp function

7.4.5.1 Object 0x63B0: Type

This parameter defines the progression of the ramp.

ValvePressureControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x63B0	0	Type	INT8	rw	Y	0…3	0

Value description

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

Table 34: Possible values of parameter <Type> (0x63B0)

7.4.5.2 One-quadrant ramp (ramp type 1)

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

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

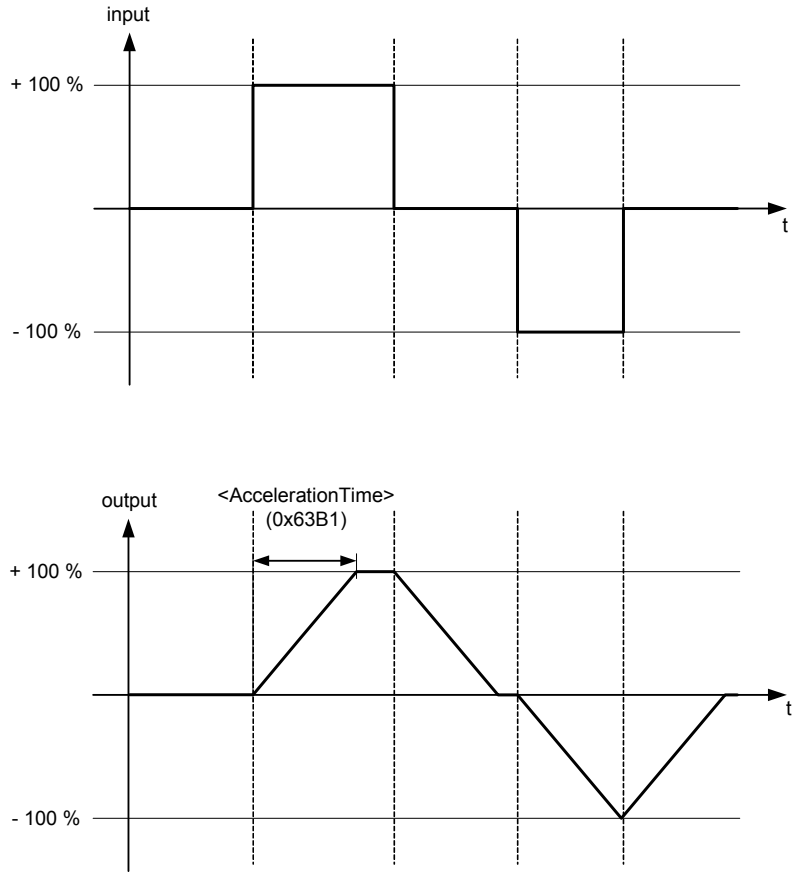


Figure 49: Ramp type 1

7.4.5.2.1 Object 0x63B1: Acceleration time

This parameter defines the output signal's maximum rate of change. The acceleration time corresponds to the time the signal needs for a change from 0 to 100 % as shown in the figure above. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x63B1	1	AccelerationTime	UINT16	rw	Y	UINT16	0
0x63B1	2	Unit	UINT8	ro	-	UINT8	3
0x63B1	3	AccelerationTime_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.4.5.3 Two-quadrant ramp (ramp type 2)

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

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

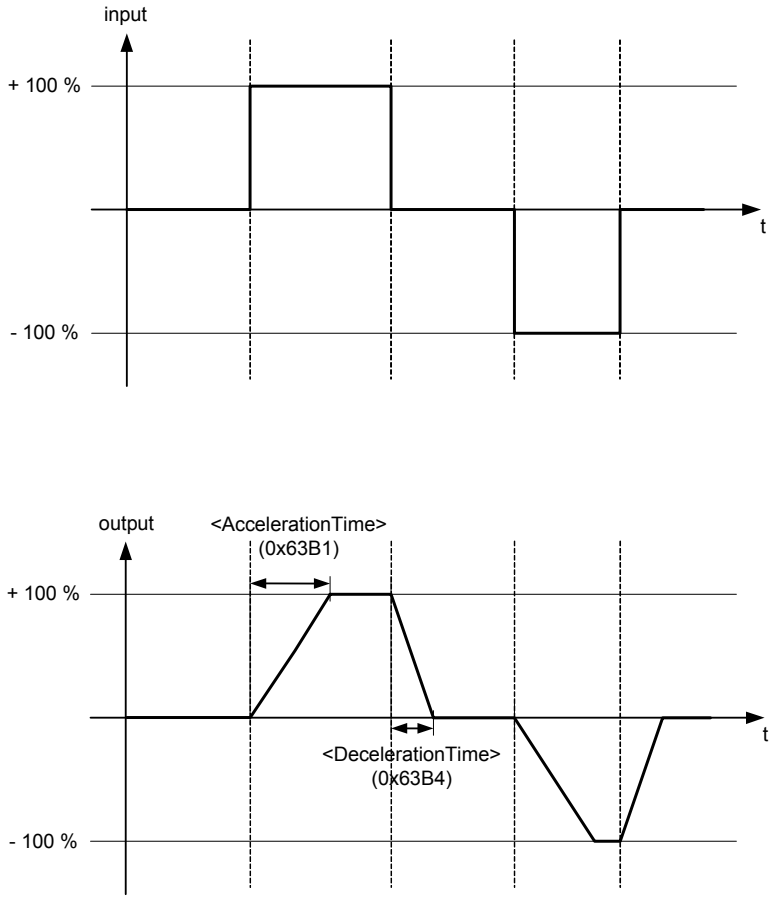


Figure 50: Ramp type 2

7.4.5.3.1 Object 0x63B1: Acceleration time

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

⇨ Chapter "7.4.5.2.1 Object 0x63B1: Acceleration time", page 106

7.4.5.3.2 Object 0x63B4: Deceleration time

This parameter defines the output signal's maximum rate of change. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x63B4	1	DecelerationTime	UINT16	rw	Y	UINT16	0
0x63B4	2	Unit	UINT8	ro	-	UINT8	3
0x63B4	3	DecelerationTime_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

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.

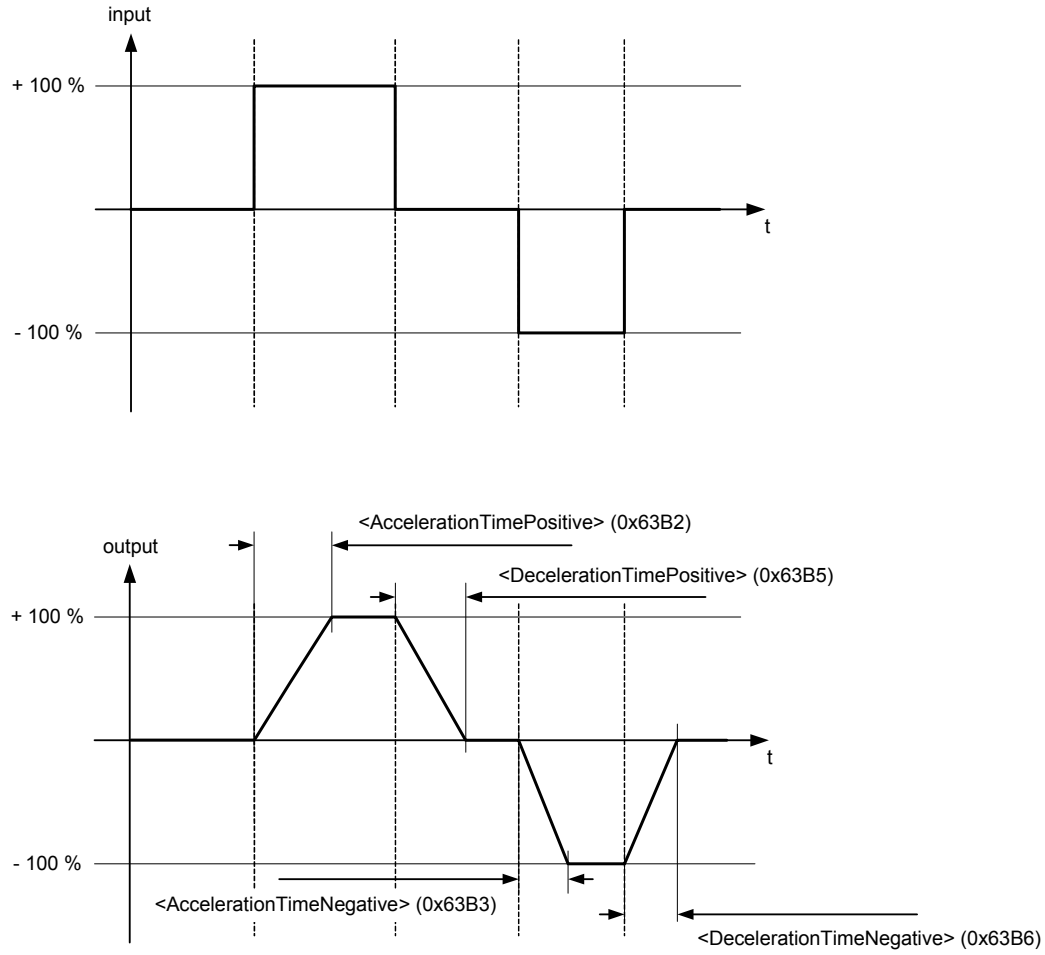


Figure 51: Ramp type 3

7.4.5.4.1 Object 0x63B2: Acceleration time positive

This parameter defines the output signal's maximum rate of change on the positive side. The acceleration time corresponds to the time that the signal needs for a change from 0 to 100 %. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x63B2	1	AccelerationTimePositive	UINT16	rw	Y	UINT16	0
0x63B2	2	Unit	UINT8	ro	-	UINT8	3
0x63B2	3	AccelerationTimePositive_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.4.5.4.2 Object 0x63B3: Acceleration time negative

This parameter defines the output signal's maximum rate of change on the negative side. The acceleration time corresponds to the time that the signal needs for a change from 0 to 100 %. The acceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x63B3	1	AccelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x63B3	2	Unit	UINT8	ro	-	UINT8	3
0x63B3	3	AccelerationTimeNegative_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.4.5.4.3 Object 0x63B5: Deceleration time positive

This parameter defines the output signal's maximum rate of change on the positive side. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x63B5	1	DecelerationTimePositive	UINT16	rw	Y	UINT16	0
0x63B5	2	Unit	UINT8	ro	-	UINT8	3
0x63B5	3	DecelerationTimePositive_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.4.5.4.4 Object 0x63B6: Deceleration time negative

This parameter defines the output signal's maximum rate of change on the negative side. The deceleration time corresponds to the time that the signal needs for a change from 100 to 0 %. The deceleration time can be specified with a resolution of 1 s, 100 ms, 10 ms or 1 ms.

ValvePressureControl_DemandValueGenerator_Ramp							
Index	Sub-index	Parameter name	Data type	Access	Persis-tence	Value range	Default
0x63B6	1	DecelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x63B6	2	Unit	UINT8	ro	-	UINT8	3
0x63B6	3	DecelerationTimeNegative_Prefix	INT8	rw	Y	–4…0	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.5 Pressure controller

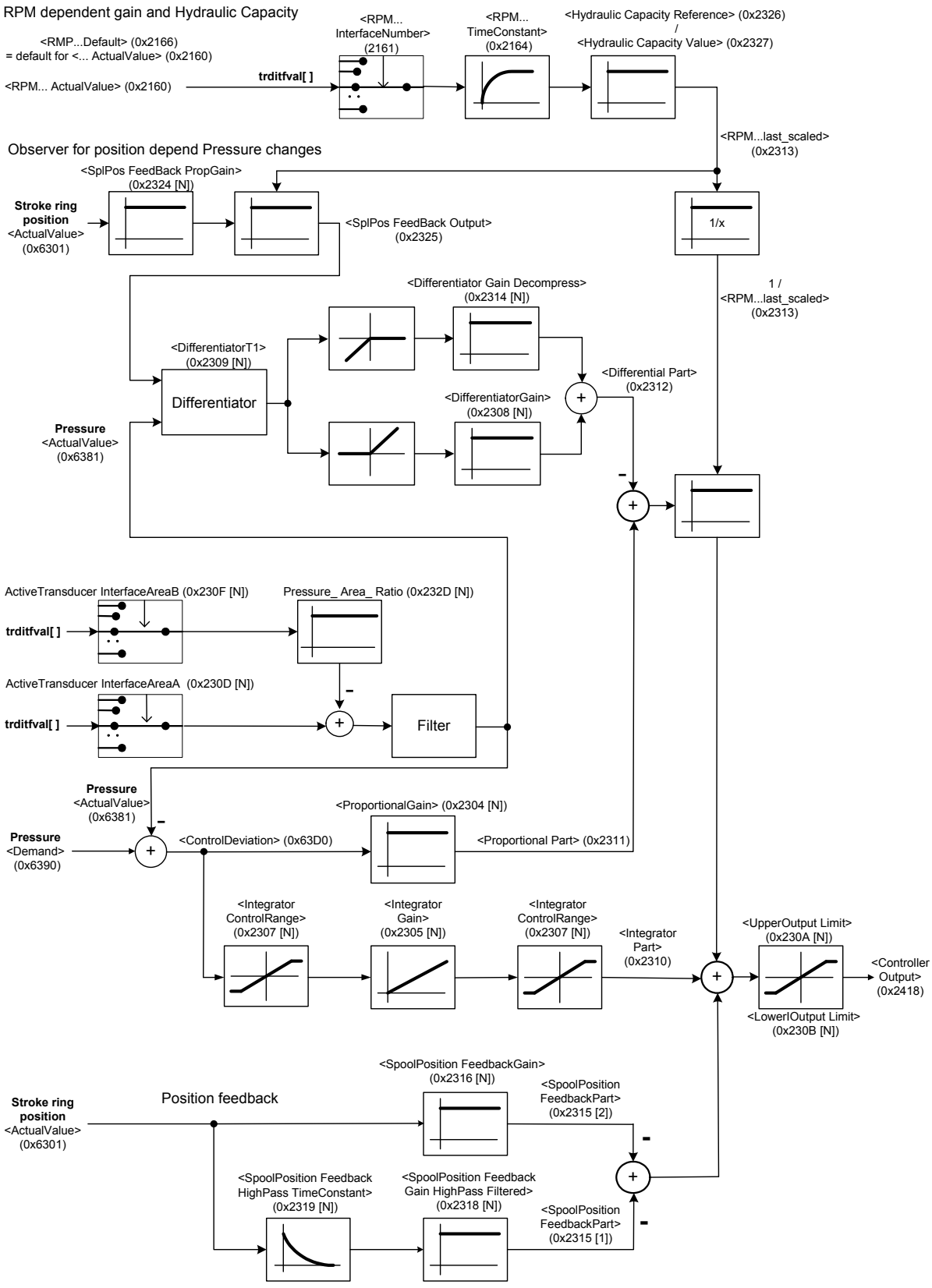


Figure 52: Pressure controller

⇨ Chapter "7.4.1 Object 0x6390: Demand value", page 101
⇨ Chapter "6.3 Actual value transducer interface", page 57

7.5.1 Objects 0x2304 / 0x2311: Proportional gain

<ProportionalGain>
The proportional gain of the pressure controller.
Usual values: 0.5…2

ValvePressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2304	1…16	ProportionalGain	FLOAT32	rw	Y	0…+inf	DSV

<ProportionalPart>
The proportional portion of the pressure controller's proportional controller unit. This parameter can be used to observe the behavior of the proportional portion.

ValvePressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2311	0	ProportionalPart	FLOAT32	ro	N	FLOAT32	None

7.5.2 Objects 0x2305 / 0x2307 / 0x2310: Integrator

<IntegratorGain>

ValvePressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2305	1…16	IntegratorGain	FLOAT32	rw	Y	0…+inf	DSV

<IntegratorControlRange>
The control range of the pressure controller's integrator. If the pressure-control deviation lies within this range, then the integrator is working with the set integrator gain <IntegratorGain>.
Usual values: 0…163

ValvePressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2307	1…16	IntegratorControlRange	INT16	rw	Y	0…32767	DSV

<IntegratorPart>
The integral proportion of the pressure controller's integrator. This parameter can be used to observe the behavior of the integrator.

ValvePressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2310	0	IntegratorPart	FLOAT32	ro	N	FLOAT32	None

7.5.3 Objects 0x230A / 0x230B: Upper/lower output limit

ValvePressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x230A	1…16	UpperOutputLimit	INT16	rw	Y	<LowerOut-putLimit>… 32767	16384
0x230B	1…16	LowerOutputLimit	INT16	rw	Y	-32768… <Upper-OutputLimit>	-16384

7.5.4 Objects 0x2308 / 0x2309 / 0x2312 / 0x2314: Differentiator

<DifferentiatorGain>
The pressure controller's differentiator gain that effects the positive input.
Usual values: 0.01…0.08

ValvePressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2308	1…16	DifferentiatorGain	FLOAT32	rw	Y	0…+inf	DSV

<DifferentiatorGainDecompress>
The pressure controller's differentiator gain decompress that effects the negative input.
Usual values: 0.01…0.08

PumpPressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2314	1…16	DifferentiatorGainDe-compress	FLOAT32	rw	Y	0…+inf	DSV

<DifferentiatorT1>
The time constant of the pressure controller's differentiator in seconds.

ValvePressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2309	1…16	DifferentiatorT1	FLOAT32	rw	Y	0…+inf	DSV

<DifferentialPart>
The differentiator portion of the pressure controller's differentiator. This parameter can be used to observe the behavior of the differentiator portion.

ValvePressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2312	0	DifferentialPart	FLOAT32	ro	N	FLOAT32	None

7.5.5 Stroke ring position depend pressure gain

A change in the stroke ring position will cause a change in the pressure. The feedback of the stroke ring position to the pressure controller will allow a higher p gain for the pressure control.

7.5.5.1 Object 0x2324[N]: Spool position feedback gain

This parameter allows the setting of the strength of the feedback (0 = no feedback).

SpoolPositionFeedBackGain							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2324	N=1…16	SpoolPositionFeedBackGain	FLOAT32	rw	Y	Float	0.0

7.5.5.2 Object 0x2325; Spool position feedback out

This parameter is used to monitor the scaled stroke position.

SpoolPositionFeedBackOut							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2325	0	SpoolPositionFeedBackOut	FLOAT32	ro	-	Float	None

7.5.6 Motor Revolutions Per Minute (RPM) dependent pressure controller

One major aspect in the development of new industrial machines is a low energy consumption and noise level. A way to tackle these objectives, in machines containing radial piston pumps, is to reduce the rotational speed of the motor and thus reduce the rotational speed of the radial piston pumps. This reduction of the revolutions per minute can be applied in phases where the maximum output of the pump is not necessary, e.g. the casting mould is closed and the injected plastics is about to cool down.

To maintain the control dynamic despite the lower rotational speed, the pressure controller output must be amplified the more the lower the rotational speed is. Therefore the regular pressure controller output is divided by a factor which is proportional to the *RPM_normed* (normed revolutions per minute).

The *RPM_normed* is the delayed output of the transducer interface <ActualValue1…4> (0x6110…0x6113) with the interface number <MotorRevolutionsPerMinuteInterfaceNumber> (0x2161). If the input for the rotational speed is the demand value for the electric motor, the effect of the rotational speed dependency must be delayed due to the fact, that the motor will need some time to achieve the demanded rotational speed. This is achieved by an added PT1.

A higher hydraulic capacity has the same effect to the pressure control then a lower RPM, so we will multiply the inverse hydraulic capacity to the RPM dependent gain.

With the abbreviations:

RPM_depend_gain

HYDR_Reference = <HydraulicCapacityReference> (0x2326)

HYDR_Value = <HydraulicCapacityValue> (0x2327)

We calculate:

$$\text{RPM\_depend\_gain} = \frac{1}{\text{normed\_RPM}} \cdot \frac{\text{HYDR\_Value}}{\text{HYDR\_Reference}}$$

The inverse RPM_depend_gain can be monitored with <MotorRevolutionsPerMinute_last_scaled> (0x2313).

7.5.6.1 Object 0x2160: MotorRevolutionsPerMinute_ActualValue

This parameter may be used to receive the RPM via CAN PDO. Before the machine controller is sending the CAN PDO, the <MotorRevolutionsPerMinute_Default> (0x2166) is valid.

MotorRevolutionsPerMinute_ActualValue							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2160	1	MotorRevolutionsPerMinute_ActualValue	INT16	rw	N	10…INT16	<RPM…Default> (0x2166)
0x2160	2	RPM_Unit	INT8	rw	-	INT8	0
0x2160	3	RPM_Prefix	INT8	rw	-	INT8	0

7.5.6.2 Object 0x2166: MotorRevolutionsPerMinute_Default

This parameter is used as default value for <MotorRevolutionsPerMinute_ActualValue> (0x2160). If the input of the transducer interface is the PDO writable parameter <MotorRevolutionsPerMinute_ActualValue> (0x2160) then the default must be 1,500 = nominal rotational speed = 100 % to get gain = 1.0 after reset before the PDO is running.

MotorRevolutionsPerMinute_Default							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2166	1	MotorRevolutionsPerMinute_Default	INT16	rw	Y	1…INT16	16384
0x2166	2	RPM_Unit	INT8	rw	-	INT8	0
0x2166	3	RPM_Prefix	INT8	rw	-	INT8	0

7.5.6.3 Object 0x2164: MotorRevolutionsPerMinute_TimeConstant

PT delay time of the e function in ms. After approx. five times *TimeConstant* the internal rpm equals the input rpm.

MotorRevolutionsPerMinute_TimeConstant							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2164	0	MotorRevolutionsPerMinute_TimeConstant	INT16	rw	Y	INT16	0

7.5.6.4 Object 0x2161: MotorRevolutionsPerMinute_InterfaceNumber

Number to select the interface which is reading the <MotorRevolutionsPerMinute_ActualValue> (0x2160).

InterfaceNumber = 0 will switch off the rpm depending gain function (gain = 1).

MotorRevolutionsPerMinute_InterfaceNumber							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2161	0	MotorRevolutionsPerMinute_InterfaceNumber	INT8	rw	Y	0…4	0

7.5.6.5 Object 0x2313: MotorRevolutionsPerMinute_last_scaled

This parameter is the last scaled and delayed RPM multiplied with the hydraulic capacity. It is the inverse RPM depend gain.

MotorRevolutionsPerMinute_last_scaled							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2313	0	MotorRevolutionsPerMinute_last_scaled	FLOAT32	ro	-	FLOAT32	0

7.5.6.6 Objects 0x2326, 0x2327: HydraulicCapacityReference and HydraulicCapacityValue

A higher hydraulic capacity has the same effect to the pressure control then a lower rpm, so we will multiply the inverse hydraulic capacity to the rpm dependent gain.

$$\text{Hydraulic capacity} = \frac{\text{<HydraulicCapacityReference> (0x2326)}}{\text{<HydraulicCapacityValue> (0x2327)}}$$

HydraulicCapacityReference							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2326	1	HydraulicCapacityReference	INT16	rw	Y	INT16	16384
0x2326	2	HydraulicCapacityUnit	UINT8	ro	-	UINT8	0
0x2326	3	HydraulicCapacityPrefix	INT8	ro	-	INT8	0

HydraulicCapacityValue							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2327	1	HydraulicCapacityValue	INT16	rw	Y	INT16	<HydraulicCapacityReference> (0x2326)
0x2327	2	HydraulicCapacityUnit	UINT8	ro	-	UINT8	0
0x2327	3	HydraulicCapacityPrefix	INT8	ro	-	INT8	0

7.5.7 Object 0x2418: Pressure controller output

This parameter indicates the actual pressure controller output value which is forwarded to the position controller.

ValvePressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2418	0	ControllerOutput	INT16	ro	N	INT16	None

7.5.8 Objects 0x2315…0x2316 / 0x2318…0x2319: Position feedback

Description on how the following parameters act:
⇨ Figure 52, page 110

<SpoolPositionFeedbackPart>

PumpPressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2315	1…2	SpoolPositionFeed-backPart	FLOAT32	ro	Y	FLOAT32	None

<SpoolPositionFeedbackGain>

PumpPressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2316	1…16	SpoolPositionFeed-backGain	FLOAT32	rw	Y	0…+inf	DSV

<SpoolPositionFeedbackGainHighPassFiltered>

PumpPressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2318	1…16	SpoolPositionFeed-backGainHighPassFil-tered	FLOAT32	rw	Y	0…+inf	DSV

<SpoolPositionFeedbackHighPassTimeConstant>

PumpPressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2319	1…16	SpoolPositionFeed-backHighPassTimeCon-stant	FLOAT32	rw	Y	0…+inf	DSV

7.5.9 Object 0x2317: Suck limitation

<SuckLimitationForPD>
This parameter limits the pressure controller output and thus ensures a minimum flow of the pump.

PumpPressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2317	1…16	SuckLimitationForPD	INT16	rw	Y	-32768…0	-16384

7.5.10 Object 0x2350: Active parameter set number

The pressure controller contains parameters to influence the control behavior. A parameter setup of the pressure controller is called pressure controller parameter set. For the pressure controller 16 parameter sets can be saved. The switching is done by the parameter <ActiveParameterSetNumber> (0x2350).

ValvePressureControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2350	0	ActiveParameterSetNumber	UINT8	rw	Y	1…16	1

The following table shows all pressure controller parameters. The <ActiveParameterSetNumber> (0x2350) sets the sub-indexes of the controller parameters.

Pressure controller parameter set

Index	**Parameter name**
0x2304	<ProportionalGain> ⇨ Chapter "7.5.1 Objects 0x2304 / 0x2311: Proportional gain", page 111
0x2305	<IntegratorGain> ⇨ Chapter "7.5.2 Objects 0x2305 / 0x2307 / 0x2310: Integrator", page 111
0x2307	<IntegratorControlRange> ⇨ Chapter "7.5.2 Objects 0x2305 / 0x2307 / 0x2310: Integrator", page 111
0x2308	<DifferentiatorGain> ⇨ Chapter "7.5.4 Objects 0x2308 / 0x2309 / 0x2312 / 0x2314: Differentiator", page 112
0x2309	<DifferentiatorT1> ⇨ Chapter "7.5.4 Objects 0x2308 / 0x2309 / 0x2312 / 0x2314: Differentiator", page 112
0x230A	<UpperOutputLimit> ⇨ Chapter "7.5.3 Objects 0x230A / 0x230B: Upper/lower output limit", page 112
0x230B	<LowerOutputLimit> ⇨ Chapter "7.5.3 Objects 0x230A / 0x230B: Upper/lower output limit", page 112
0x2314	<DifferentiatorGainDecompress> ⇨ Chapter "7.5.4 Objects 0x2308 / 0x2309 / 0x2312 / 0x2314: Differentiator", page 112
0x2316	<SpoolPositionFeedbackGain> ⇨ Chapter "7.5.8 Objects 0x2315…0x2316 / 0x2318…0x2319: Position feedback", page 116
0x2317	<SuckLimitationForPD> ⇨ Chapter "7.5.9 Object 0x2317: Suck limitation", page 116
0x2318	<SpoolPositionFeedbackGainHighPassFiltered> ⇨ Chapter "7.5.8 Objects 0x2315…0x2316 / 0x2318…0x2319: Position feedback", page 116
0x2319	<SpoolPositionFeedbackHighPassTimeConstant> ⇨ Chapter "7.5.8 Objects 0x2315…0x2316 / 0x2318…0x2319: Position feedback", page 116

Table 35: Pressure controller parameter set

7.6 p/Q closed loop

The p/Q controller (control mode = 5) is a combination of the pressure and volume-flow functions, i.e., it enables regulation of the volume flow resp. pressure.

In the p/Q closed loop, a pressure and a position are provided as a setpoint and the actual pressure (0x6381) and the position (0x6301) are measured.
⇨ Figure 53, page 119

The pressure controller output value (0x2418) is compared with the pump position controller output (0x2158). The lower of these demand values is forwarded to the position controller.

The p/Q controller makes the transition between position controller and pressure controller automatically.

Status word bits
There are two status word bits which provide information whether the p/Q control type is active and if a control fault is detected:

- Bit 8 - Pressure controller effective
  This status word bit is set, if, and only if, the p/Q control type is active.
- Bit 11 - Control error
  The control error bit in the status word is formed from the linking of the two controller deviations for pressure and position. If this bit is set, a controller error is detected.

⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46
⇨ Chapter "7.12 Monitoring", page 133

Description of the position and pressure controller parameters as well as the control monitoring functions:
⇨ Chapter "7.3 Stroke ring (spool) position controller", page 96
⇨ Chapter "7.5 Pressure controller", page 110

7.6.1 Controller structure

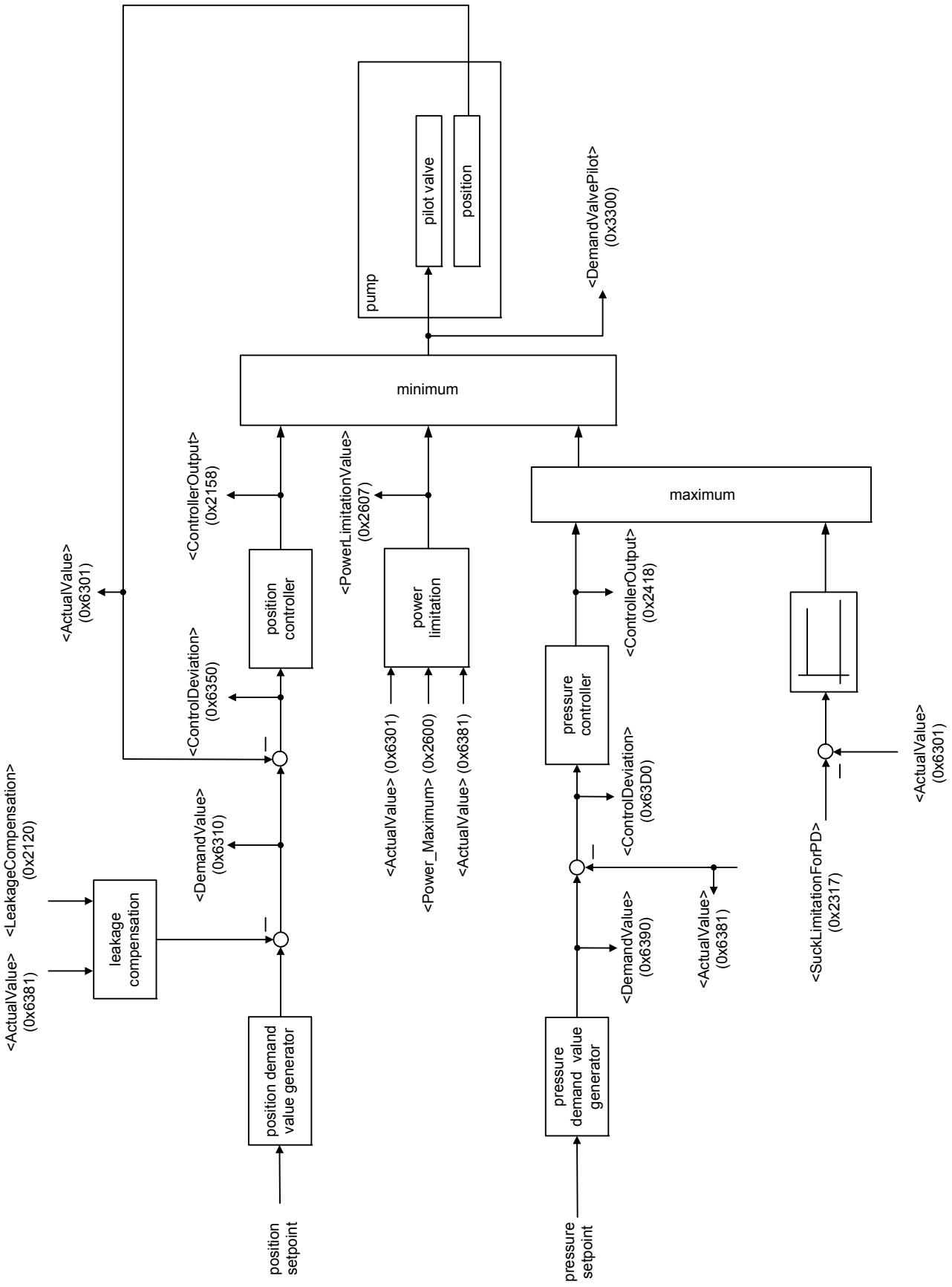


Figure 53: p/Q controller

Leakage compensation
Detailed information on the leakage compensation:
⇨ Chapter "7.3.7 Leakage compensation", page 100

Suck limitation
Detailed information on the suck limitation:
⇨ Chapter "7.5.9 Object 0x2317: Suck limitation", page 116

7.6.1.1 Object 0x2158: Controller output

ValveMainStageControl							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2158	0	ControllerOutput	INT16	ro	N	INT16	None

7.7 Power limitation

The maximum flow of the pump can be limited by means of the power limitation function. The limit value, i.e., the maximum power is written to the parameter <Power_Maximum> (0x2600). If this power limit is reached, the position demand is reduced.
⇨ Figure 43, page 96

The following figure shows the structure of the power limitation function and the parameters influencing this behavior.

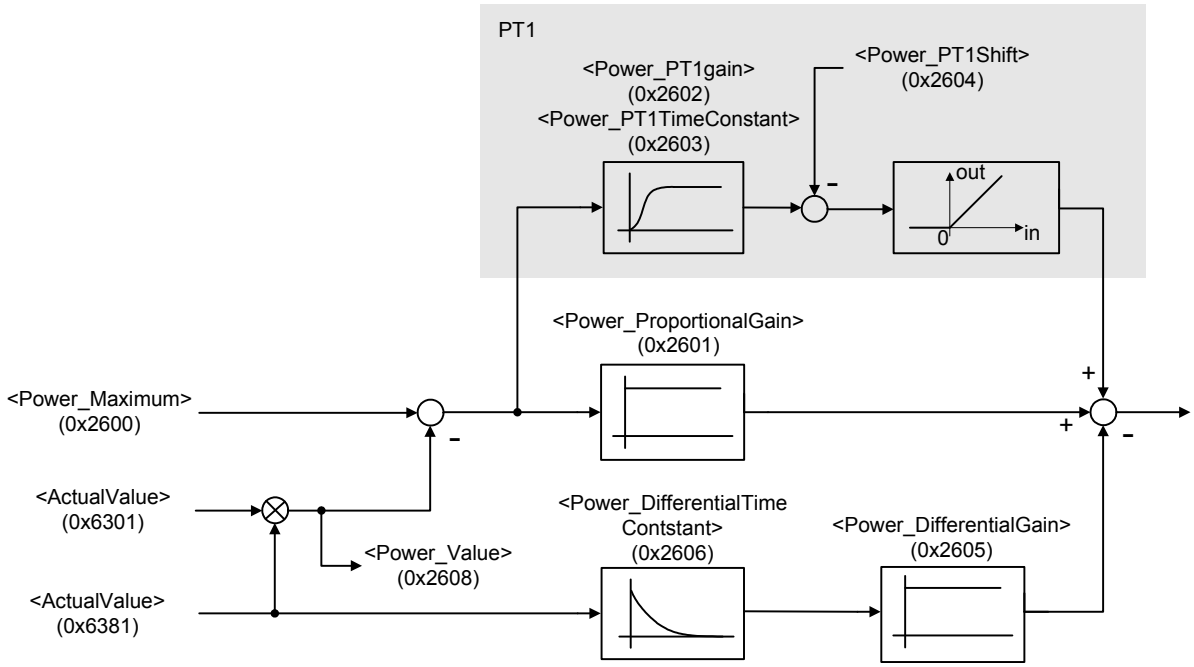


Figure 54: Power limitation structure

Enabling of the power limitation function
The power limitation function is switched on and off with the bit 11 in the control word. If bit 11 is set to true, power limitation is enabled.
⇨ Chapter "5.1.2 Object 0x6040: Control word", page 38

PT1
The additional low pass filter of the power difference allows a delayed controller takeover from position to power control. Hence, the limiting of the position is delayed by the low pass. This allows the pump to exceed the power limit for a short time period depending on the time constant of the low pass element.

7.7.1 Objects 0x2600…0x2606 / 0x2608: Power limitation

<Power_Maximum>
If this maximum power value is reached, the flow demand is reduced.

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2600	0	Power_Maximum	INT16	rw	Y	0…16384	DSV

<Power_ProportionalGain>
The proportional gain of the power limitation controller.

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2601	0	Power_ProportionalGain	FLOAT32	rw	Y	0…+inf	DSV

<Power_PT1Gain>
The gain of the PT1 element.

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2602	0	Power_PT1Gain	FLOAT32	rw	Y	0…+inf	DSV

<Power_PT1TimeConstant>
The time constant of the PT1 element in seconds.

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2603	0	Power_PT1TimeConstant	FLOAT32	rw	Y	0…+inf	DSV

<Power_PT1Shift>
The <Power_PT1Shift> defines the threshold above which the power deviation is passed through.

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2604	0	Power_PT1Shift	INT16	rw	Y	0…32767	DSV

<Power_DifferentialGain>
The differentiator gain of the power limitation controller.

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2605	0	Power_DifferentialGain	FLOAT32	rw	Y	0…+inf	DSV

<Power_DifferentialTimeConstant>
The time constant of the power limitation controller's differentiator in seconds.

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2606	0	Power_DifferentialTime Constant	FLOAT32	rw	Y	0…+inf	DSV

<Power_Value>
Actual power value.

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x2608	0	PowerValue	INT16	ro	N	INT16	None

7.8 Master/Slave operation

Scenario
Two or more pumps are feeding one hydraulic system to increase the max. flow. This could cause some undefined states in the system (e.g., pump1 = 100 % flow; pump2 = -100 % flow).

The RKP-D has the capability to manage such situations. One pump is operating as a master pump. All other pumps are working as a slave pump. The master processes all set commands: pressure, flow and power limit. The slaves only follow the position of the master. The actual value of the stroke ring and controller output is broadcasted to the slaves via local CAN.

For a description of the hydraulic conditions refer to the Operation Manual RKP-D.

The local CAN Network is reserved for the master/slave operation.

In master/slave operation, there must be only one active pressure controller (master) where the remaining pumps (slaves) are purely flow controlled by the master pump.

Enabling master/slave communication
The enabling of the communication between master and slave pump and the selection whether a pump is operating as master or slave is done with the parameter <MasterSlaveSelector> (0x21A5).

Enabling the operation of a pump as a slave pump is set by the control word bit 9 (either local control word or control word via bus). If this bit is set for a pump, it operates as a slave pump, i.e., the slave pump gets the flow command from the master pump.

Description of control words:
⇨ Chapter "5.1.2 Object 0x6040: Control word", page 38
⇨ Chapter "5.1.3 Object 0x4040: Local control word", page 39

Values on master/slave side
On the master side the transmitted position is held by the parameter <ActualValue> (0x6301). On the slave side the received position command is held by the parameter <SpoolSetPointFromMaster> (0x21A4). The parameters 0x6301 and 0x21A4 both hold the master position.
⇨ Chapter "7.3.2.1 Object 0x6301: Actual value", page 97

The pump controller output from the master is transmitted as well. This is done through the parameter <DemandValvePilot> (0x3300). The receive parameter on the slave side is <SpoolFeedForwardFromMaster> (0x21A7).

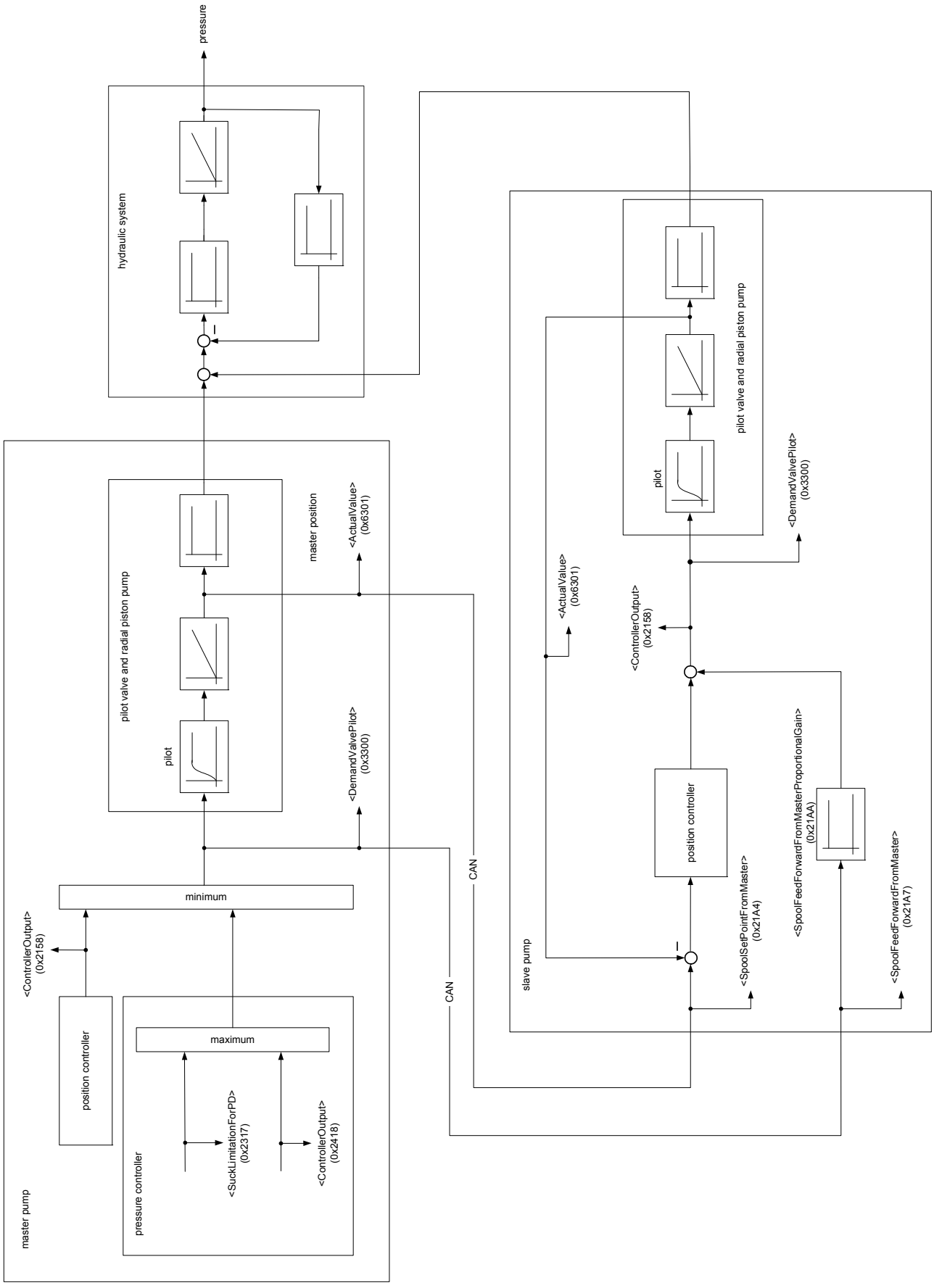


Figure 55: RKP-D control structure with slave pump

7.8.1 Objects 0x21A4 / 0x21A5 / 0x21A7 / 0x21AA: Master slave operation

<SpoolSetPointFromMaster>
By setting bit 9 to true in the control word, the pump is working in the slave modus. This command can be monitored by the parameter <SpoolSetPointFromMaster> in the slave pump.
On the master side the transmitted position is held by the parameter <ActualValue> (0x6301). On the slave side the received position command is held by the parameter <SpoolSetPointFromMaster> (0x21A4).

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x21A4	0	SpoolSetPointFromMaster	INT16	rw	N	INT16	None

Description of control words:
⇨ Chapter "5.1.2 Object 0x6040: Control word", page 38
⇨ Chapter "5.1.3 Object 0x4040: Local control word", page 39

<MasterSlaveSelector>
This parameter is used to enable the local network, i.e., the communication between the master and slave pumps and to define a pump as the master or slave pump.

- In master/slave operation, there must be only one active pressure controller (master) where the remaining pumps (slaves) are purely flow controlled by the master pump.
- A slave pump must not be configured to be slave 2 or 3, where no other pump is already configured to slave 1. The slave pumps must have different slave numbers.

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x21A5	0	MasterSlaveSelector	INT8	rw	Y	-1…3	DSV

Values description

Value	**Function**	**Termination resistor**
-2	Master	No
-1	Master	Yes
0	Solo	No
1	Slave 1	Yes
2	Slave 2	No
3	Slave 3	No

Table 36: Master slave selection

<SpoolFeedForwardFromMaster>
This parameter holds on the slave side the pump controller output transmitted by the master (written to the parameter <DemandValvePilot> (0x3300) on master side).

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x21A7	0	SpoolFeedForwardFromMaster	INT16	rw	N	INT16	None

<SpoolFeedForwardFromMasterProportionalGain>
Proportional gain for the pump controller output transmitted by the master (on slave side held by the parameter <SpoolFeedForwardFromMaster>, 0x21A7).

PumpController							
Index	Sub-index	Name	Data type	Access	Persis-tence	Value range	Default
0x21AA	0	SpoolFeedForwardFromMaster-ProportionalGain	FLOAT32	rw	Y	FLOAT32	DSV

7.8.2 Object 0x5B00: Module identifier

This parameter represents the local CAN identifier of the pump.

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

7.9 Local holding pressure switchover

When running a RKP-D in the device mode "setpoint via bus", setpoint data is typically transferred from the machine's controller to the pump with a fixed cycle time. The ideal time for the switchover from position control to hold pressure control lays in between the cycle period and may vary from shot to shot. Caused by the tiny, but unavoidable jitter, the repeatability of the injection process decreases, which may impact the product quality.

To counteract this phenomenon, the "local holding pressure switchover" function has been incorporated into the RKP-D. This function manages the transition from the velocity control phase to holding pressure phase in almost real time, without getting the machine controller involved. This way the transmission delay between the device and the machine controller has no influence.

This function is only needed for injection machines and fieldbus operated RKP-D pumps.

Operation

Local holding pressure switchover can only be activated if the RKP-D is in device state 'ACTIVE' and in control mode p/Q only.

Once this function has been activated (by setting bit 13 of the control word), the RKP-D starts monitoring the pressure actual value until the hold pressure trigger level (parameter <HoldPressureTrigger>, 0x2107) is reached (see the following figure).

At this moment the RKP-D activates the internal pressure setpoints (<PressureSetpoint>, 0x2108 and <SpoolPositionSetpoint>, 0x2109) and sends an acknowledgement of the performed holding pressure switchover by setting immediately bit 14 of the status word.

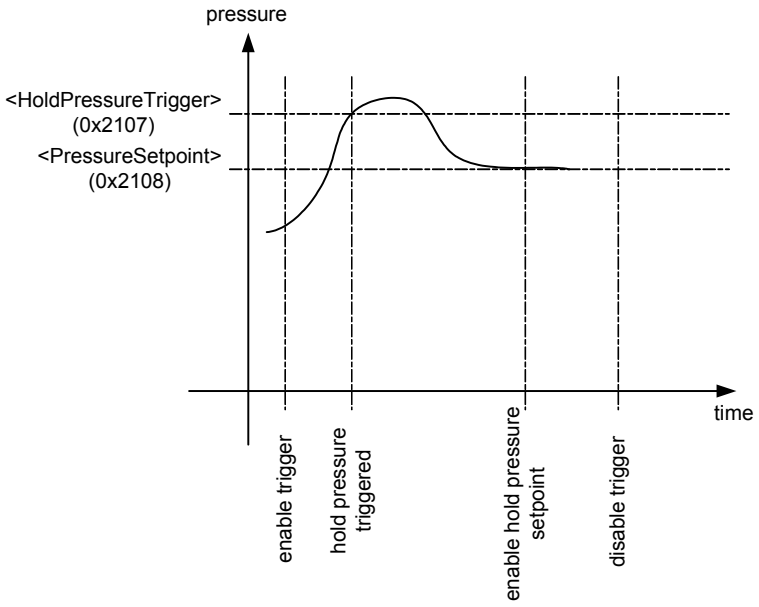


Figure 56: Hold pressure setpoint behavior

<ControlWord> bit 13	Meaning
0	Disable trigger
1	Enable trigger

Table 37: Control word bit 13 values

In case the holding pressure trigger level is not reached, the superior controller sends a trigger signal after a certain time.

Control word and status word bits

Local holding pressure switchover can be deactivated by resetting bit 13 of the control word, which results in a reset of bit 14 of the status word on the pump side.

Furthermore the holding pressure switchover can be forced externally by setting the bit 14 in the control word at anytime. The completion of this action will also be acknowledged by setting of bit 14 in the status word. Bit 14 of the status word is set, whenever it is switched over to local holding pressure setpoints internally.

The following table shows the relations between the control word bits 13 and 14 and the status word bit 14.

<ControlWord>		<StatusWord>	
Bit 14	**Bit 13**	**Bit 14**	**Remark**
0	0	0	Holding pressure setpoints remain inactive.
0	1	0/1	Pump continues monitoring trigger level. When reaching the trigger level, local holding pressure set-points will be activated.
1	X	1	Local holding pressure setpoints activated, regardless of trigger level.

Table 38: Control word and status word bits for local holding pressure switchover

Control word bits

Bit	<ControlWord>
0…12	⇨ Chapter "5.1.2 Object 0x6040: Control word", page 38
13	Activate holding pressure switchover
14	Force holding pressure switchover

Table 39: Control word bits

Status word bits

Bit	<StatusWord>
0…12	⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46
14	Holding pressure setpoints active

Table 40: Status word bits

Status machine of the hold pressure switchover

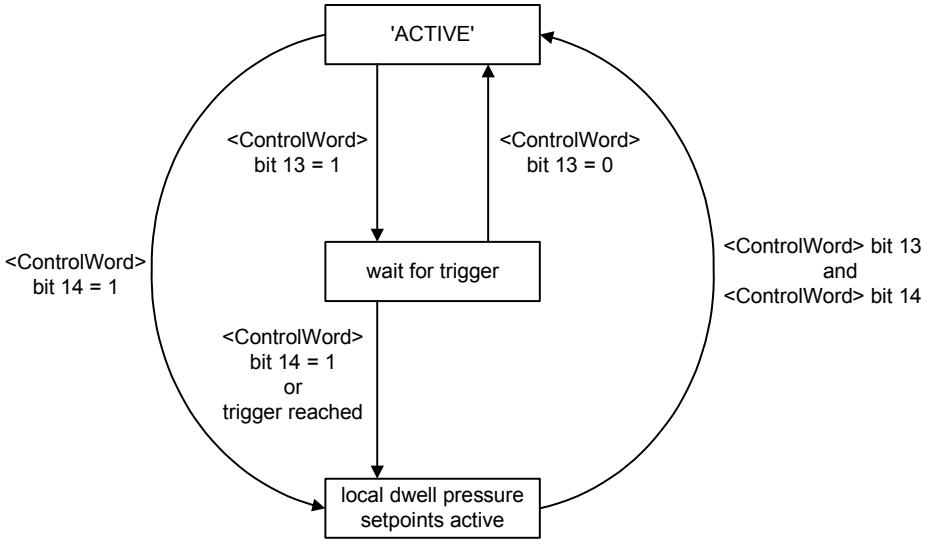


Figure 57: Status machine of the hold pressure enable

7.9.1 Objects 0x2107…0x2109: Local holding pressure switchover

<HoldPressureTrigger>
This parameter sets the holding pressure trigger level at which the internal holding pressure setpoints (parameters <PressureSetpoint> and <SpoolPositionSepoint> are activated by the RKP-D when the local holding pressure switchover is activated (see Figure 56).

HoldPressureControl							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2107	1	HoldPressureTrigger	INT16	rw	Y	INT16	16384

<PressureSetpoint>
The pressure setpoint is activated by the RKP-D when the holding pressure trigger level (<HoldPressureTrigger>, 0x2107) is reached (see Figure 56). This pressure is held until a new pressure setpoint is transmitted or the trigger is disabled.

HoldPressureControl							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2108	1	PressureSetpoint	INT16	rw	Y	INT16	DSV

<SpoolPositionSetpoint>
The position setpoint is activated by the RKP-D when the holding pressure trigger level (<HoldPressureTrigger>, 0x2107) is reached (see Figure 56). This position is held until a new pressure setpoint is transmitted or the trigger is disabled.

HoldPressureControl							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2109	1	SpoolPositionSetpoint	INT16	rw	Y	INT16	16384

7.10 Flushing mode

This function is included in digital pumps with internal pressure supply. The flushing mode prevents overheating of the pump. If all relevant demand values (p, Q, or both) are below 1 %, the pump might overheat because there is not enough flow to cool the pump. Therefore the pump goes into flushing mode after the time stored in the parameter 0x21A1 (<FlushingTime>) has elapsed.

The flushing time represented by the parameter 0x21A1 is stored in seconds. Typically the value is 180. Hence, if the relevant demand values are below 1 % for the duration of 3 minutes, the pump will go into flushing mode.

PumpController							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x21A1	0	FlushingTime	UINT16	ro	Y	0…300	180

7.11 Analog parameter set switching

This feature allows to choose pressure controller parameter settings through an analog input. The influenced parameters build a parameter set. This chapter describes how the parameter sets can be chosen via an analog input.

A parameter set consists of the following parameters:

- The saved control word (0x2141)
- The parameter selecting the hybrid mode (0x2148)
- The parameter addressing a pressure controller parameter set (0x2142)

The analog parameter set switching is done over an interface. The interface is assigned to an analog input. Depending on the analog input signal level a parameter set is activated. The selection of the interface is done by the parameter <SourceTransducer> (0x2143). The value of the parameter <ActiveParameterSet> (0x2146) corresponds to a certain input range of the interface (see Table 41).

Description of the interface assignment:
⇨ Chapter "6.3 Actual value transducer interface", page 57

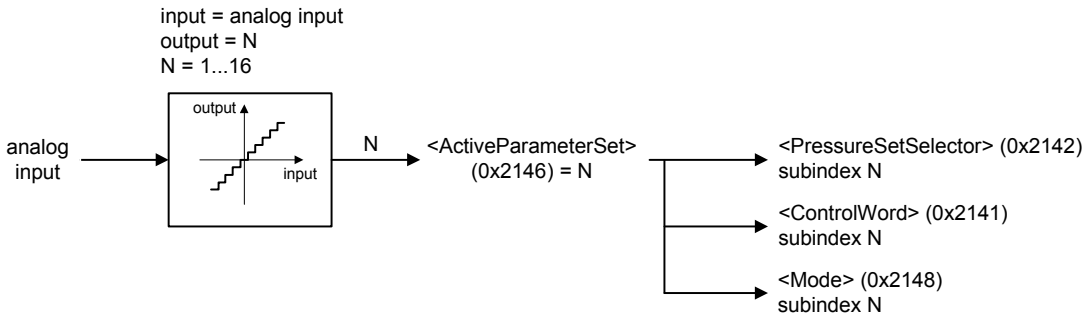


Figure 58: Assignment of analog input value to active parameter set

The analog input is quantized according Table 41. The resulting value from 1…16 is assigned to <ActiveParameterSet> (0x2146). The <ActiveParameterSet> sets the sub-index of three parameters. The <PressureSetSelector> (0x2142) defines which controller parameters will be active.

Description of <PressureSetSelector> (0x2142):
⇨ Chapter "7.5.10 Object 0x2350: Active parameter set number", page 117

The following table shows the relation between the currently active control parameter set (indicated by parameter 0x2146) and the interface input range.
Description of analog input types:
⇨ Chapter "6.4 Analog inputs", page 68

Value of <ActiveParameterSet> (0x2146)	Analog input type 1, 3 (±100 % corresponds to ±10 V or ±10 mA)		Analog input type 2, 4, 7, 10 (0…100 % corresponds to 0…10 V or 0…10 mA)		Analog input type 5, 8 (0…100 % corresponds to 4…20 mA)	
	Low limit	High limit	Low limit	High limit	Low limit	High limit
1	-10	-6,8	0	1,6	4	6,56
2	-6,2	-5,8	1,9	2,1	7,04	7,36
3	-5,2	-4,8	2,4	2,6	7,84	8,16
4	-4,2	-3,8	2,9	3,1	8,64	8,96
5	-3,2	-2,8	3,4	3,6	9,44	9,76
6	-2,2	-1,8	3,9	4,1	10,24	10,56
7	-1,2	-0,8	4,4	4,6	11,04	11,36
8	-0,2	0,2	4,9	5,4	11,84	12,16

Table 41: Relation between value of <ActiveParameterSet> and interface input range (part 1 of 2)

Value of <ActiveParameterSet> (0x2146)	Analog input type 1, 3 (±100 % corresponds to ±10 V or ±10 mA)		Analog input type 2, 4, 7, 10 (0…100 % corresponds to 0…10 V or 0…10 mA)		Analog input type 5, 8 (0…100 % corresponds to 4…20 mA)	
	Low limit	High limit	Low limit	High limit	Low limit	High limit
9	0,8	1,2	5,4	5,6	12,64	12,96
10	1,8	2,2	5,9	6,1	13,44	13,76
11	2,8	3,2	6,4	6,6	14,24	14,56
12	3,8	4,2	6,9	7,1	15,04	15,36
13	4,8	5,2	7,4	7,6	15,84	16,16
14	5,8	6,2	7,9	8,1	16,64	16,96
15	6,8	7,2	8,4	8,6	17,44	17,76
16	7,8	10	8,9	10	18,24	20

Table 41: Relation between value of <ActiveParameterSet> and interface input range (part 2 of 2)

The <PressureSetSelector> (0x2142) assigns an active parameter set to the desired pressure controller parameter set. This means, each sub-index of 0x2142 holds a value of 1…16 which is written into the parameter <ActiveParameterSetNumber> (0x2350).

List of possible controller parameters:
⇨ Table 35, page 117

Description of active parameter set:
⇨ Chapter "7.11 Analog parameter set switching", page 130

Description of <PressureSetSelector> (0x2142):
⇨ Chapter "7.11.1 Objects 0x2141…0x2146 / 0x2148: Active parameter set", page 132

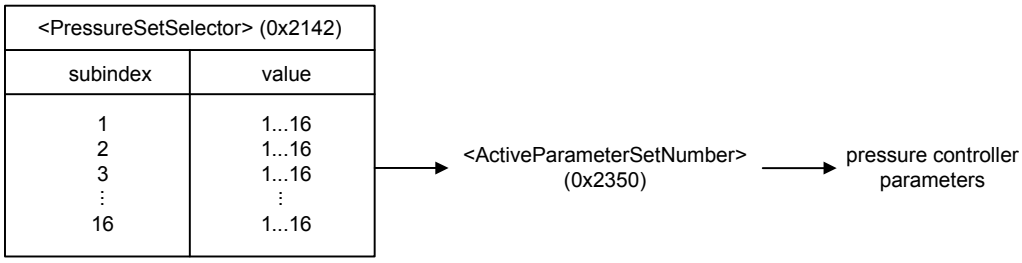


Figure 59: Assignment of <PressureSetSelector> sub-index to active parameter set number

7.11.1 Objects 0x2141…0x2146 / 0x2148: Active parameter set

<SourceTransducer>
Selection of the interface that is assigned to an analog input.
Interface definition:
⇨ Chapter "6.3 Actual value transducer interface", page 57

AnalogParameterSetSwitching							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2143	0	SourceTransducer	UINT8	rw	Y	1…4	DSV

<Enable>
Activates the analog parameter set switching function.

AnalogParameterSetSwitching							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2145	0	Enable	UINT8	rw	Y	0…1	DSV

<ActiveParameterSet>
Indicates the currently active control parameter set.

AnalogParameterSetSwitching							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2146	0	ActiveParameterSet	UINT8	ro	N	0…16	DSV

<ControlWord>

AnalogParameterSetSwitching							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2141	1…16	ControlWord	UINT16	rw	Y	1…65535	DSV

The parameter sets allow to save a control word. At activation of a set, the saved control word is applied to the local control word (<LocalControlWord>, 0x4040).

The sub-index of the control word points to the corresponding control parameter set.

Example:
Sub-index 2 of 0x2141 holds the value 7 (dec).

This means, the device is set to the device status 'ACTIVE' if the control parameter set 2 is activated by a corresponding signal applied at the selected analog input.

Description of the device states:
⇨ Chapter "5.2.1 DSM states", page 41

<PressureSetSelector>
This parameter holds a pressure controller parameter set number. At activation of a set, the saved number is applied to the active parameter set number (<ActiveParameterSetNumber>, 0x2350).

AnalogParameterSetSwitching							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2142	1	PressureSetSelector	UINT8	rw	Y	1…16	1
0x2142	2	PressureSetSelector	UINT8	rw	Y	1…16	2
…							
0x2142	16	PressureSetSelector	UINT8	rw	Y	1…16	16

<Mode>
This parameter is used to activate/deactivate the hybrid mode. Each pressure parameter set has a hybrid mode. Writing the value 1 to one of the parameter's sub-indexes means hybrid mode is active for the corresponding parameter set. 0 means hybrid mode is deactivated for the selected parameter set.

Description hybrid mode correction:
⇨ Chapter "7.2.7 Hybrid mode correction", page 95

AnalogParameterSetSwitching							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2148	1…16	Mode	UINT8	rw	Y	0…16	DSV

7.12 Monitoring

The control deviation monitoring is only active if the associated controller is active. In p/Q mode, the <ControlMode> (0x6043) is set to 5 (p/Q controller), the effective deviation monitoring depends on the active controller indicated by the bit 8 (pressure controller effective) of the <StatusWord> (0x6041).

⇨ Chapter "7.1.1 Object 0x6043: Control mode", page 82
⇨ Chapter "5.2.3 Object 0x6041: Status word", page 46

7.12.1 Spool position control deviation monitoring

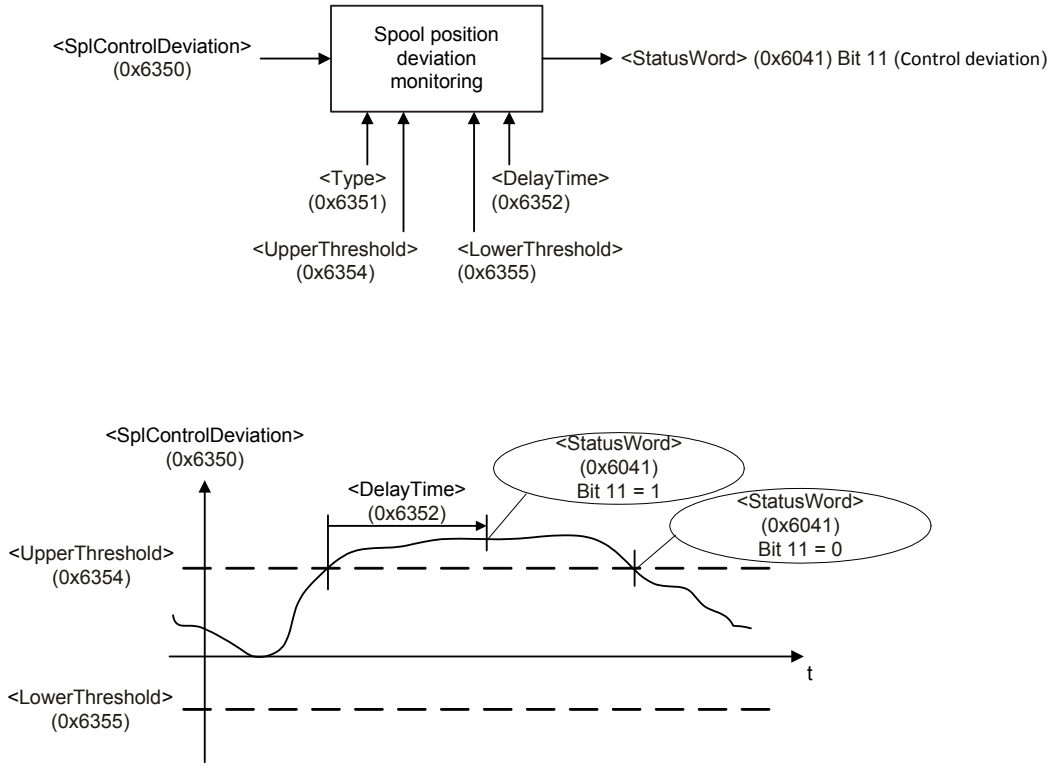


Figure 60: Spool position control deviation monitoring

⇨ Chapter "7.3.3.1 Object 0x6350: Control deviation", page 97

7.12.1.1 Object 0x6351: Type

The parameter <Type> (0x6351) is used to activate or deactivate the standard spool position control deviation monitoring function.

ValvePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6351	0	Type	INT8	rw	Y	0…1	0

Value description

<Type>	Description
0	Spool position control deviation monitoring off.
1	Spool position control deviation monitoring on.

Table 42: Possible values of parameter <Type> (0x6351)

7.12.1.2 Object 0x6352: Delay time

The delay time defines the minimal duration of a control deviation before a fault is active. The Time is set to zero if the position is inside the window, if the fault is acknowledged or if one of the following parameters is changed: <Type> (0x6351), <UpperThreshold> (0x6354), <LowerThreshold> (0x6355) or <DelayTime>(0x6352).

ValvePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6352	1	DelayTime	UINT16	rw	Y	UINT16	30
0x6352	2	Unit	UINT8	ro	-	UINT8	3
0x6352	3	Prefix	INT8	ro	-	INT8	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.12.1.3 Object 0x6354: Upper threshold

This parameter defines the upper threshold of the control deviation's tolerance band.

ValvePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6354	1	UpperThreshold	INT16	rw	Y	INT16	512
0x6354	2	Unit	UINT8	ro	-	UINT8	0
0x6354	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.12.1.4 Object 0x6355: Lower threshold

This parameter defines the lower threshold of the control deviation's tolerance band.

ValvePositionControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x6355	1	LowerThreshold	INT16	rw	Y	INT16	–512
0x6355	2	Unit	UINT8	ro	-	UINT8	0
0x6355	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.12.2 Pressure control deviation monitoring

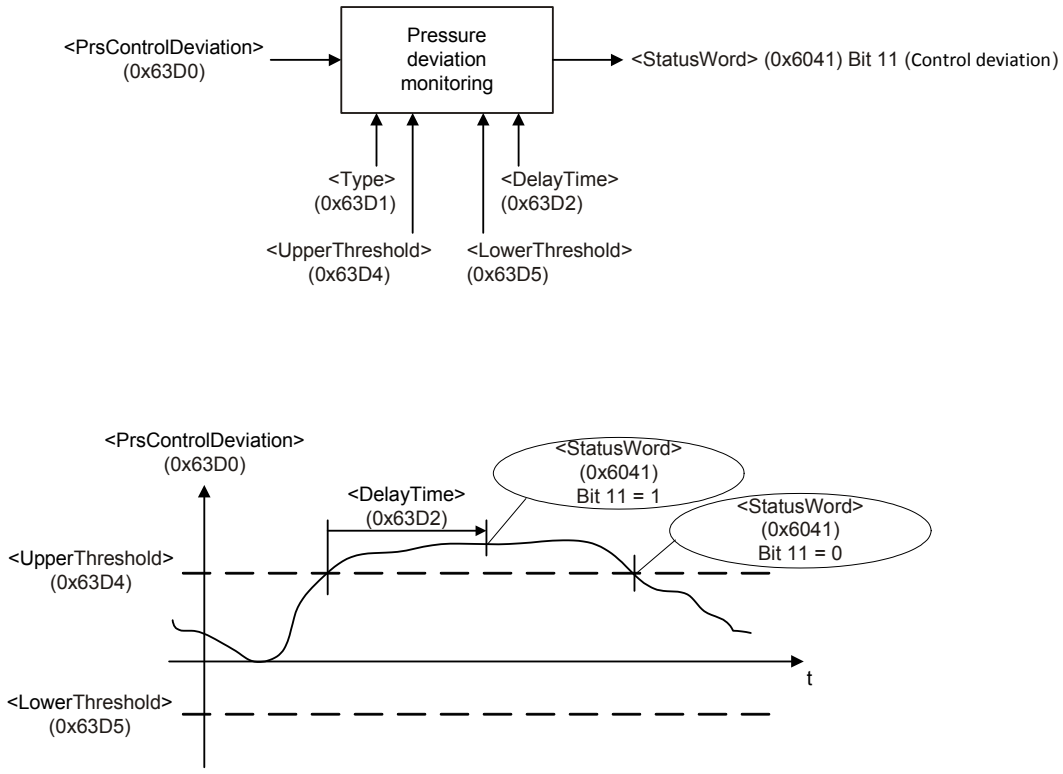


Figure 61: Pressure control deviation monitoring

7.12.2.1 Object 0x63D1: Type

The parameter <Type> (0x63D1) is used to activate or deactivate the standard pressure control deviation monitoring function.

ValvePressureControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x63D1	0	Type	INT8	rw	Y	0…1	0

Value description

<Type>	**Description**
0	Pressure control deviation monitoring off.
1	Pressure control deviation monitoring on.

Table 43: Possible values of parameter <Type> (0x63D1)

7.12.2.2 Object 0x63D2: Delay time

The delay time defines the minimal duration of a control deviation before a fault is active. The Time is set to zero if the position is inside the window, if the fault is acknowledged or if one of the following parameters is changed: <Type> (0x63D1), <UpperThreshold> (0x63D4), <LowerThreshold> (0x63D5) or <DelayTime>(0x63D2).

ValvePressureControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x63D2	1	DelayTime	UINT16	rw	Y	UINT16	30
0x63D2	2	Unit	UINT8	ro	-	UINT8	3
0x63D2	3	Prefix	INT8	ro	-	INT8	–3

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.12.2.3 Object 0x63D4: Upper threshold

This parameter defines the upper threshold of the control deviation's tolerance band.

ValvePressureControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x63D4	1	UpperThreshold	INT16	rw	Y	INT16	512
0x63D4	2	Unit	UINT8	ro	-	UINT8	0
0x63D4	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.12.2.4 Object 0x63D5: Lower threshold

This parameter defines the lower threshold of the control deviation's tolerance band.

ValvePressureControl_ControlMonitoring							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x63D5	1	LowerThreshold	INT16	rw	Y	INT16	–512
0x63D5	2	Unit	UINT8	ro	-	UINT8	0
0x63D5	3	Prefix	INT8	ro	-	INT8	0

⇨ Chapter "2.8.3 Units and prefix parameter", page 17

7.12.3 Failsafe monitoring

This monitoring is used to monitor the so called "failsafe position" that means the spring centered spool position. The control window is defined by an <UpperLimit> (0x2421) and a <LowerLimit> (0x2422), which are set by the factory. If the failsafe spool position monitoring function is enabled by setting the parameter <DigitalOutputType1> (0x2420) to 1 (failsafe spool position monitoring on), the servo valve monitors the failsafe position of the spool position <ActualValue> (0x6301) and sets the digital output 1.

⇨ Chapter "6.7.2 Object 0x5E41: Digital output type", page 77

WARNING

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.

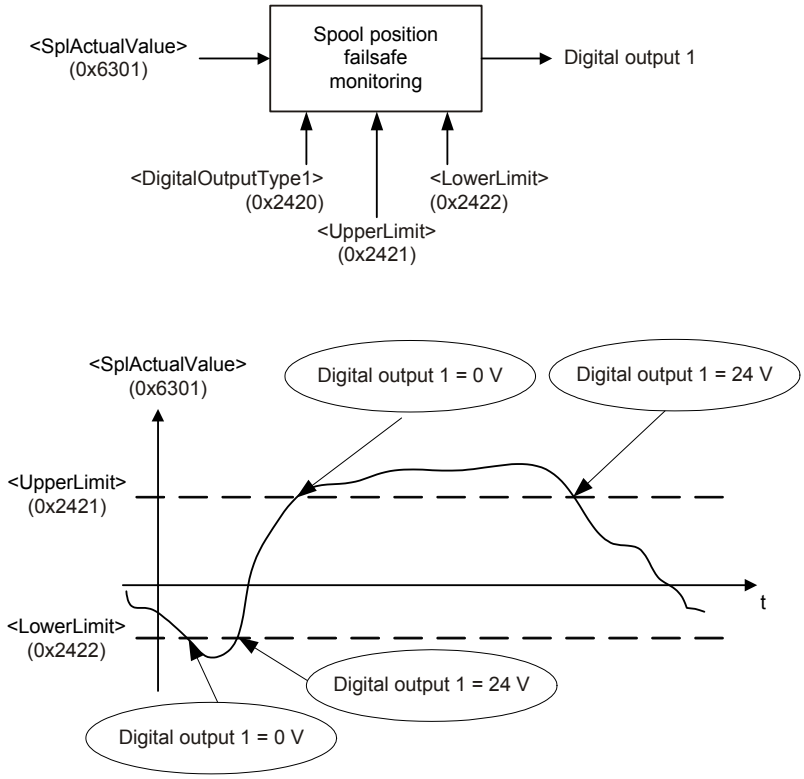


Figure 62: Failsafe monitoring

Electrical signal

Spool position <SplActualValue> (0x6301)	Electrical signal (negative logic)
Actual spool position value is within the failsafe monitoring window	Digital output 1 = 24 V
Actual spool position value is outside the failsafe monitoring window	Digital output 1 = 0 V

Failsafe spool position monitoring behavior depending on the DSM state <StatusWord> (0x6041)

<StatusWord> (0x6041) (DSM state)	Failsafe spool position monitoring behavior
'INIT'	No failsafe spool position monitoring active.
'DISABLED', 'HOLD', 'ACTIVE', 'FAULT DISABLED', 'FAULT HOLD'	Failsafe spool position monitoring active if one <DigitalOutputType> (0x2420) is set to 1 (failsafe spool position monitoring on).

7.12.3.1 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)…32768	32768

7.12.3.2 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)	–32768

7.12.4 Pilot/single stage actual spool position monitoring

In case of a cable break, the fault code 0x11 (Pilot/single stage LVDT cable break) is thrown and the device state machine state changes to 'NOT READY'.

WARNING

The servo valve must be serviced by Moog service technicians.
The servo valve may behave unpredictably.

7.12.5 Main/dual stage actual spool position monitoring

Monitoring is only active in case of device state greater 'INIT'. In case of a cable break, the fault code 0x14 (Main stage LVDT cable break) is generated and the device state machine state changes to 'NOT READY'.

WARNING

The servo valve must be serviced by Moog service technicians.
The servo valve may behave unpredictably.

7.12.6 Analog input cable break monitoring

Cable break monitoring is available for the analog input signals and for the sensor supply wires. The following table shows the possible cable break monitoring features. The cable break monitoring depends on the input type. This dependence is explained in this chapter.

	Power supply cable break monitoring	**Sensor cable break monitoring**		
		0…10 mA –10…10 mA	**4…20 mA**	**0…10 V –10…10 V**
Analog input 0	No	No	Yes (fault code 31)	No
Analog input 1	No	No	Yes (fault code 32)	No
Analog input 2	Yes (fault code 28)	No	Yes if <MonitoringCurrent> (0x3217) is set to 0 (fault code 33)	Yes if <MonitoringCurrent> (0x3217) is set to 1 (fault code 33)
Analog input 3	Yes (fault code 29)	No	Yes if <MonitoringCurrent> (0x3228) is set to 0 (fault code 34)	Yes if <MonitoringCurrent> (0x3228) is set to 1 (fault code 34)
Analog input 4	Yes (fault code 30)	No	Yes if <MonitoringCurrent> (0x3227) is set to 0 (fault code 35)	Yes if <MonitoringCurrent> (0x3227) is set to 1 (fault code 35)

Table 44: Cable break monitoring features

The following fault codes are generated in the case of a cable break. The fault reaction itself can be configured using the fault reaction logic.
⇨ Chapter "8.1.3 Fault reaction type", page 162

Input	**Fault code**		**Fault description**
	Dec.	**Hex.**	
Analog input 0	31	0x20	Analog input 0 current too low (4…20 mA)/ADC overflow (voltage)
Analog input 1	32	0x21	Analog input 1 current too low (4…20 mA)/ADC overflow (voltage)
Analog input 2	33	0x22	Analog input 2 current too low (4…20 mA)/ADC overflow (voltage)
	28	0x1D	Analog input 2 supply cable break/short circuit
Analog input 3	34	0x23	Analog input 3 current too low (4…20 mA)/ADC overflow (voltage)
	29	0x1E	Analog input 3 supply cable break/short circuit
Analog input 4	35	0x24	Analog input 4 current too low (4…20 mA)/ADC overflow (voltage)
	30	0x1F	Analog input 4 supply cable break/short circuit

Table 45: Possible fault codes

7.12.6.1 Object 0x3217: Monitoring current

If this parameter is set to 1 the cable break monitoring for the analog input 2 is enabled.

AnalogInput							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3217	0	MonitoringCurrent	UINT8	rw	Y	0…1	0

7.12.6.2 Object 0x3228: Monitoring current

If this parameter is set to 1 the cable break monitoring for the analog input 3 is enabled.

AnalogInput							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3228	0	MonitoringCurrent	UINT8	rw	Y	0…1	0

7.12.6.3 Object 0x3227: Monitoring current

If this parameter is set to 1 the cable break monitoring for the analog input 4 is enabled.

AnalogInput							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3227	0	MonitoringCurrent	UINT8	rw	Y	0…1	0

If the 0…10 mA or the ±10 mA analog input type is used, no sensor wire cable break monitoring is available. In this case only the sensor power supply monitoring for the analog inputs 2…4 is active.

If the 4…20 mA analog input type is used, cable break monitoring for all analog inputs is available.

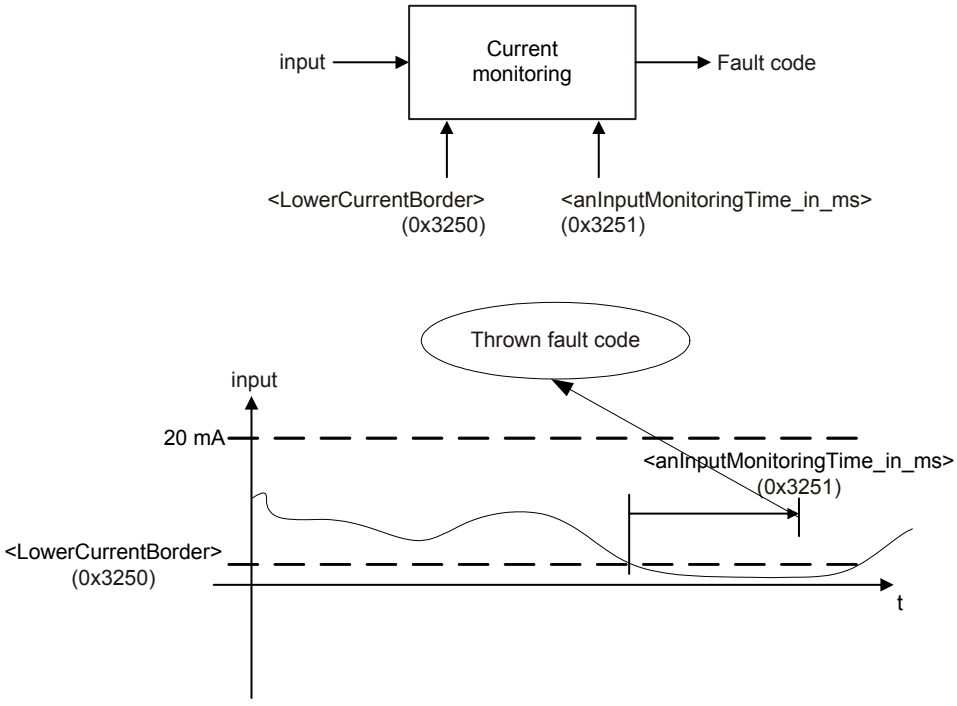


Figure 63: 4…20 mA analog input signal monitoring

If the current is below <LowerCurrentBorder> (0x3250) for <anInputMonitoringTime_in_ms> (0x3251) ms, cable break is detected.

If the 0…10 V or –10…10 V analog input type is used, cable break monitoring for all analog inputs is available. The sensor needs to be able to sink a current of at least 0.1 mA. A fault code is also generated by an input voltage above 11 V or an input voltage less than –11 V.

7.12.6.4 Object 0x3250: Lower current border

This parameter contains the lower current boarder.

AnalogInput							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3250	0	LowerCurrentBorder	FLOAT32	rw	Y	2.2…20.0	3.0

7.12.6.5 Object 0x3251: Analog input monitoring time

This parameter contains the delay time (in ms) before generating the fault code.

AnalogInput							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3251	0	anInputMonitoringTime_in_ms	UINT16	rw	Y	0…60000	10

7.12.7 Sensor power supply monitoring

For the analog inputs 2…4 a sensor power supply cable break monitoring exists. A sensor load current < 1 mA is interpreted as cable break. A short circuit of the sensor power supply is also recognized. Each sensor has its own fault detection. A short circuit on one of the sensors leads to a common fault response for all sensors.

7.12.8 Hardware monitoring

The hardware monitoring feature provides some hardware specific parameters such as power supply, board temperature and operating time.

7.12.8.1 Object 0x2803: CPU supply voltage

This parameter contains the value of the CPU supply voltage (in mV). A fault is generated if the parameter is outside its nominal range.

Nominal range: 3.1 V ≤ CpuSupplyVoltage ≤ 3.5 V

Hardware_DiagnosticData							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2803	0	CpuSupplyVoltage	UINT16	ro	-	UINT16	None

Value description

<CpuSupplyVoltage>	Thrown fault code	Fault description
CpuSupplyVoltage < 3.1 V	0x07	Internal supply voltage too low.
CpuSupplyVoltage > 3.5 V	0x08	Internal supply voltage too high.

Table 46: Fault codes

The power supply voltage should be in the range of 18…32 V to ensure proper operation.

7.12.8.2 Object 0x2804: Power supply voltage

This parameter holds the value of the power supply voltage (in mV). A fault is generated if the parameter is outside its nominal range.

Nominal range: 17 V ≤ PowerSupplyVoltage ≤ 32.5 V

Hardware_DiagnosticData							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2804	0	PowerSupplyVoltage	UINT16	ro	-	UINT16	None

Fault description

<PowerSupplyVoltage>	**Thrown fault code**	**Fault description**
PowerSupplyVoltage < 17 V	0x05	Power supply voltage too low.
PowerSupplyVoltage > 32.5 V	0x06	Power supply voltage too high.

Table 47: Fault codes

7.12.8.3 Object 0x2805: PCB temperature

This parameter contains the temperature (in °C) of the servo valve electronics. A fault is generated if the parameter value is below or exceeds the following temperature values:

PCB temperature < –20 °C
PCB temperature > 85 °C
PCB temperature > 105 °C

Hardware_DiagnosticData							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2805	0	PcbTemperature	INT16	ro	-	INT16	None

Fault description

<PowerSupplyVoltage>	**Thrown fault code**	**Fault description**
PCB temperature < –20 °C	0x0D	Electronics temperature too low.
PCB temperature > 85 °C	0x0E	Electronics temperature too high.
PCB temperature > 105 °C	0x0F	Electronics temperature exceeded.

Table 48: Fault codes

The PCB temperature should not exceed the range of –20…85 °C to ensure proper operation. The electronics temperature has a big impact on the electronics lifetime. The longest service life is achieved when the higher temperatures are avoided.

7.12.8.4 Object 0x280D: Operating time

The parameter <PowerOnTime> (0x280D) contains the power on time (in minutes) since production of the servo valve.

The parameter <OperatingTime> (0x280D) contains the time (in minutes) the servo valve is in the device state machine (DSM) states 'HOLD', 'FAULT HOLD' or 'ACTIVE'.

Hardware_DiagnosticData							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x280D	1	PowerOnTime	UINT32	ro	-	UINT32	0
0x280D	2	OperatingTime	UINT32	ro	-	UINT32	0

7.13 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.

7.13.1 Data logger state machine

The data logger is controlled by a state machine. The states and the transitions are explained in this chapter.

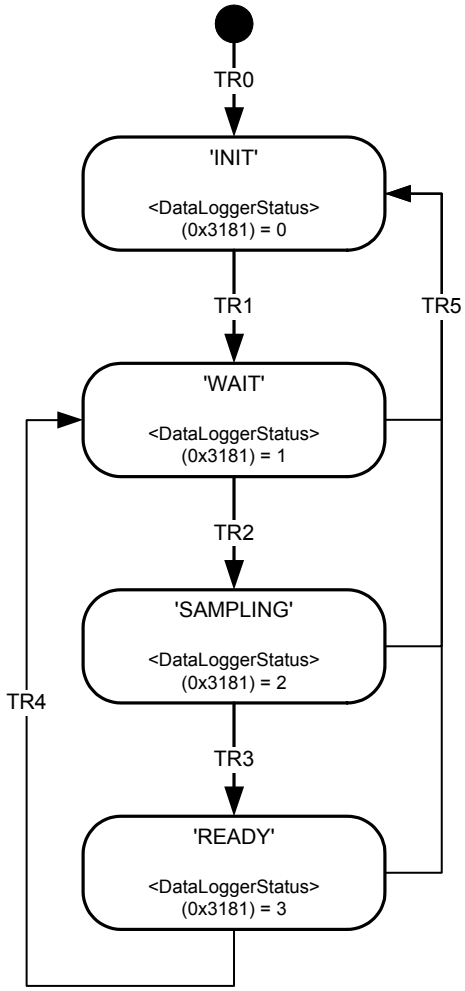


Figure 64: Data logger state machine

	<DataLoggerTriggerType> (0x3188)		
State	**0 (Free)**	**1 (Normal)**	**2 (Single)**
'INIT'	Initialize data logger.		
'WAIT'	The trigger condition is always true. If the <Control> (0x3180) is set to 1 (enable data logger), the state changes to 'SAMPLE'.	If the <Control> (0x3180) is set to 1 (enable data logger), the channels are continuously sampled to allow pre trigger. If the trigger condition is active, the start index <SampleStartOffset> (0x3187) in the ring buffer memory <Memory> (0x3186) is defined and the state is change to 'SAMPLE'.	
'SAMPLE'	The channels are continuously sampled until the ring buffer is full. If the ring buffer is full the state changes to 'READY'.		
'READY'	If the <Control> (0x3180) is set to 1 (enable data logger), the state changes to 'WAIT'.	State stays in 'READY' until <Control> (0x3180) is set to 1 (enable data logger).	

Table 49: States of the data logger state machine

Transition (TR)	Description
TR0	Start program.
TR1	Triggered by setting <Control> (0x3180) to 1 (enable data logger).
TR2	Trigger condition is active.
TR3	Data logger ring buffer is full.
TR4	Triggered by setting <Control> (0x3180) to 1 (enable data logger).
TR5	One of the following parameters has changed: <Divider> (0x3182) <EnableChannel1…4> (0x3184) <ChannelParameter1…4> (0x3185) <TriggerType> (0x3188) <TriggerParameter> (0x3189)

Table 50: Transitions of the data logger state machine

7.13.1.1 Object 0x3180: Control

This parameter enables the data logger.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3180	0	Control	UINT8	rw	N	0…1	None

Value description

<Control>	Description
0	Disable data logger.
1	Enable data logger.

Table 51: Possible values of parameter <Control> (0x3180)

7.13.1.2 Object 0x3181: Status

State of the data logger state machine.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3181	0	Status	UINT8	ro	-	0…3	None

Value description

<Status>	Description
0	'INIT'
1	'WAIT'
2	'SAMPLING'
3	'READY'

Table 52: Possible values of parameter <Status> (0x3181)

7.13.2 Channel settings

Four channels can be used within the data logger. All readable parameters are available as input for the channels.

7.13.2.1 Object 0x3185: Channel parameter

The parameters <ChannelParameter1…4> define the parameters which shall be sampled. The values are composed of index, sub-index and bit length of the chosen parameter.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3185	1	ChannelParameter1	UINT32	rw	Y	UINT32	0x63100110
0x3185	2	ChannelParameter2	UINT32	rw	Y	UINT32	0x63010110
0x3185	3	ChannelParameter3	UINT32	rw	Y	UINT32	0x63900110
0x3185	4	ChannelParameter4	UINT32	rw	Y	UINT32	0x63810110

Value description

<ChannelParameter1…4>				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Parameter bit length
Example	0x63	0x10	0x01	0x10

The example value is 0x63100110, which refers to the <SplDemandValue> (0x6310), sub-index 0x01 with a length of 16 bit (16=0x10).

7.13.2.2 Object 0x3184: Enable channel

Any channel can be switched on or off with this parameter.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3184	1	EnableParameter1	UINT8	rw	Y	0…1	0
0x3184	2	EnableParameter2	UINT8	rw	Y	0…1	0
0x3184	3	EnableParameter3	UINT8	rw	Y	0…1	0
0x3184	4	EnableParameter4	UINT8	rw	Y	0…1	0

Value description

<EnableParameter1…4>	**Description**
0	Channel disabled.
1	Channel enabled.

Table 53: Possible values of parameter <EnableParameter> (0x3184)

7.13.3 Sample frequency

The maximum sample frequency is limited to 10000 samples per seconds (10kHz). This sample frequency can be set to a smaller sample frequency with the parameter <Divider> (0x3182).

$$\text{New sample frequency} = \frac{\text{Maximum sample frequency}}{\text{<Divider> (0x3182)}}$$

The measuring time is increased by the factor <Divider> (0x3182).

7.13.3.1 Object 0x3182: Divider

This parameter contains an integer number to reduce the sampling frequency.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3182	0	Divider	UINT16	rw	Y	1…65535	1

Value description

<Divider>	Description
1	10000 samples per second (10 kHz).
2…65534	10000/2…10000/65534 samples per second.
65535	10000/65535 = 0.1526 samples per second.

Table 54: Possible values of parameter <Divider> (0x3182)

7.13.4 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).

7.13.4.1 Object 0x3189: Trigger parameter

The <TriggerParameter> (0x3189) defines the parameter which is used as trigger signal.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3189	0	TriggerParameter	UINT32	rw	Y	UINT32	0x63100110

Value description

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

The default value is 0x63100110, which refers to the <SplDemandValue> (0x6310), sub-index 0x01 with a length of 16 bit (16=0x10).

7.13.4.2 Object 0x3188: Trigger type

This parameter contains the trigger type of the data logger.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3188	0	TriggerType	UINT8	rw	Y	0…2	1

Value description

<TriggerType>	Description
0	FREE (the trigger condition is always true).
1	NORMAL (if the trigger condition matches, the data logger returns to wait state completed).
2	SINGLE (if the trigger condition matches, the data logger enters the ready state).

Table 55: Possible values of parameter <TriggerType> (0x3188)

7.13.4.3 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

7.13.4.4 Object 0x318A: Trigger coupling

This parameter contains the trigger coupling type of the data logger.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x318A	0	TriggerCoupling	UINT8	rw	Y	0…2	1

Value description

<TriggerCoupling>	Description
0	AC (AC part of the trigger signal is passed and is compared with the trigger level).
1	DC (the trigger signal is passed directly and is compared with the trigger level).
2	BITMASK (the trigger signal is passed directly and is compared bit-wise with the trigger bitmask).

Table 56: Possible values of parameter <TriggerCoupling> (0x318A)

7.13.4.5 Object 0x318B: Trigger slope

The <TriggerSlope> (0x318B) defines the edge of the signal which starts the sampling procedure.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x318B	0	TriggerSlope	UINT8	rw	Y	1…3	1

Value description

<TriggerSlope>	Description
1	Rising (trigger on a rising edge).
2	Falling (trigger on a falling edge).
3	Both (trigger on both, rising or falling edge).

Table 57: Possible values of parameter <TriggerSlope> (0x318B)

7.13.4.6 Object 0x318D: Trigger position

The <TriggerPosition> (0x318D) is provided as number of samples which shifts the starting point in the ring buffer.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x318D	0	TriggerPosition	INT32	rw	Y	INT32	0

Value description

<TriggerPosition>	Description
> 0	Post trigger.
= 0	No delay.
< 0	Pre trigger.

Table 58: Possible values of parameter <TriggerPosition> (0x318D)

7.13.5 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.

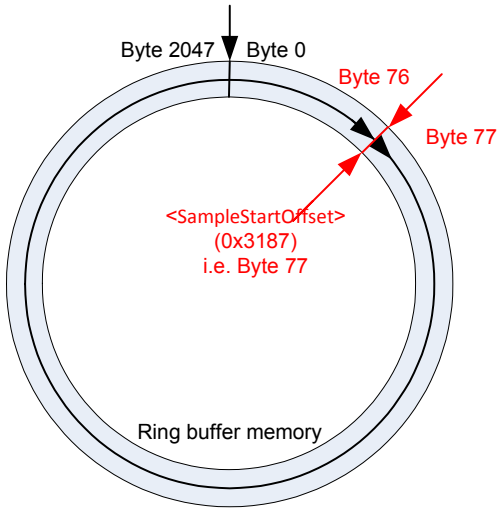


Figure 65: Data memory organization

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.

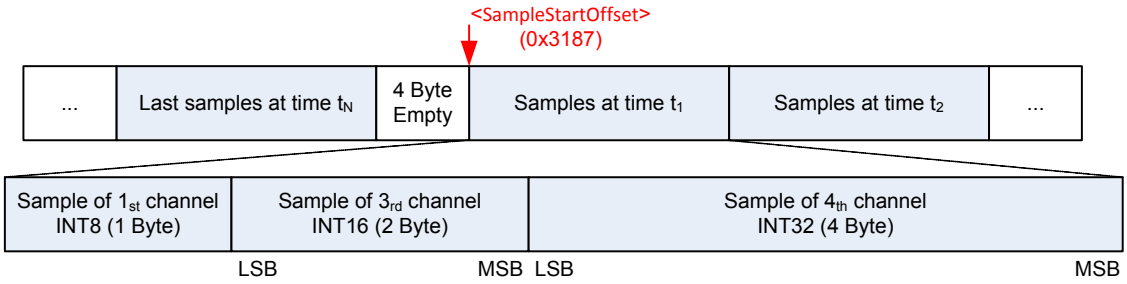


Figure 66: Data memory - mixed channel data

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.

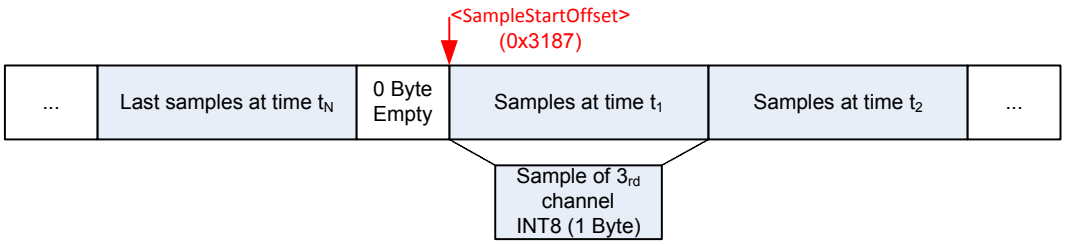


Figure 67: Data memory - one channel with INT8 parameter

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.

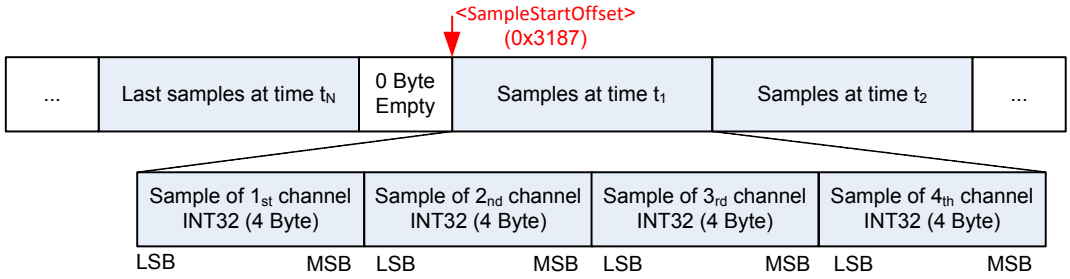


Figure 68: Data memory - four channels with INT32 parameters

7.13.5.1 Object 0x3186: Memory

The parameter <Memory> (0x3186) contains the sampled information of the four channels. The parameter is an array of UINT8 with 2048 entries/bytes.

DataLogger							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3186	0	Memory	DOMAIN	ro	-	None	None

7.13.5.2 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

7.13.5.3 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

7.14 Function generator

The servo valve internal function generator can be used to generate a periodic signal with specific shapes, thereby enabling an engineer or technician to test and examine a servo valve.
The function generator has two outputs:

- An output signal
  This signal can be used e.g. as setpoint value for the servo valve to optimize the servo valves behavior.
  Different shapes, amplitude and offset can be configured.
  ⇨ Chapter "7.14.3.1 Object 0x3101: Output signal", page 155
- 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 "7.14.3.2 Object 0x3102: Square output (Trigger signal)", page 156

7.14.1 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.

7.14.1.1 Rectangular output signal (type 1)

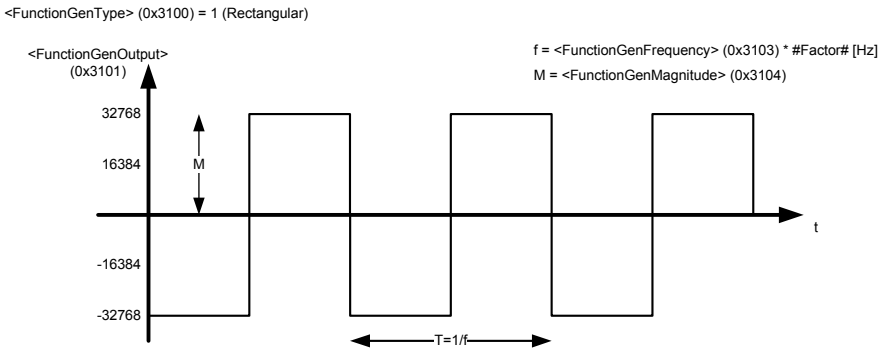


Figure 69: Rectangular output signal (type 1)

7.14.1.2 Triangle output signal (type 2)

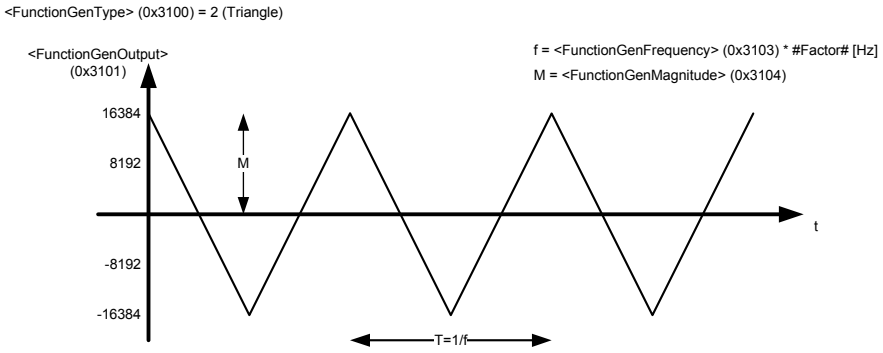


Figure 70: Triangle output signal (type 2)

7.14.1.3 Sawtooth signal (type 3)

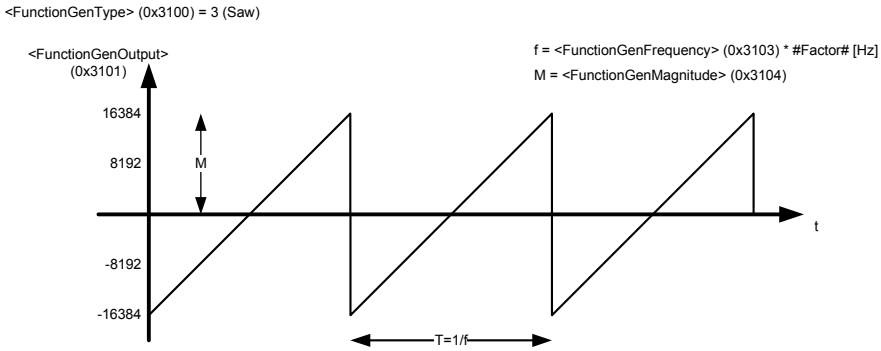


Figure 71: Saw signal (type 3)

The output signal can be inverted by setting the parameter <FunctionGenSign> (0x3105) to –1.
⇨ Chapter "7.14.1.9 Object 0x3107: Sign", page 154

7.14.1.4 Trapezoid signal (type 4)

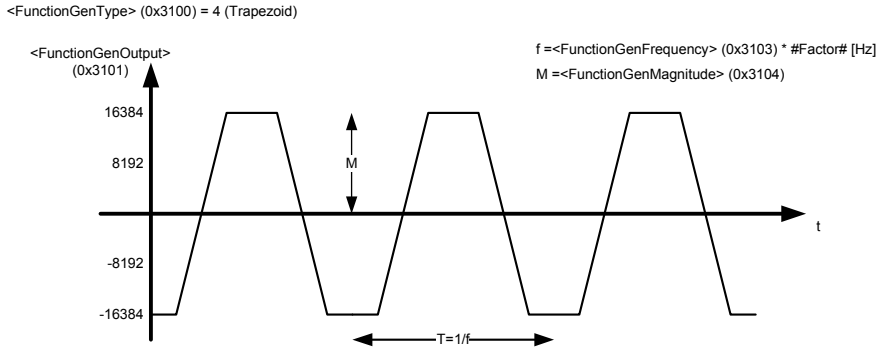


Figure 72: Trapezoid signal (type 4)

7.14.1.5 Sine signal (type 5)

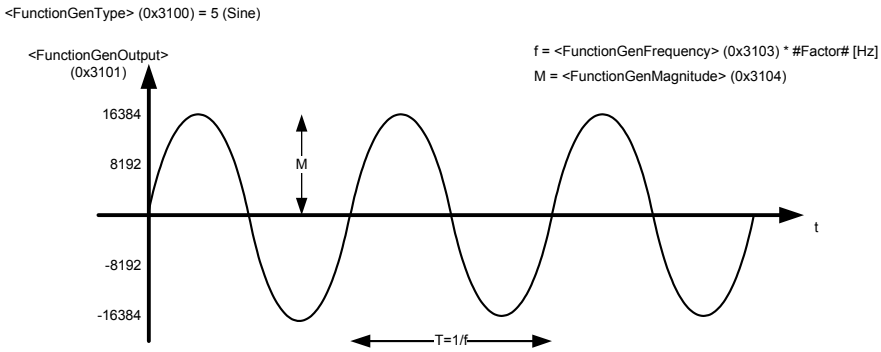


Figure 73: Sine signal (type 5)

7.14.1.6 Object 0x3100: Type

This parameter defines the function generator output signal shape.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3100	0	FunctionGenType	INT8	rw	Y	0…5	0

Value description

<FunctionGenType>	Description
0	Function generator switched off.
1	Rectangular signal output.
2	Triangle signal output.
3	Sawtooth signal output.
4	Trapezoid signal output.
5	Sine signal output.

Table 59: Possible values of parameter <Type> (0x3100)

7.14.1.7 Object 0x3104: Magnitude

This parameter is the magnitude of the function generator output signal in increments. To configure e.g. a set-point value of ±100 %, ±16384 has to be set as magnitude.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3104	0	FunctionGenMagnitude	INT16	rw	N	0…32767	0

7.14.1.8 Object 0x3105: Offset

This parameter is the offset of the function generator output signal in increments.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3105	0	FunctionGenOffset	INT16	rw	N	INT16	0

7.14.1.9 Object 0x3107: Sign

This parameter is the sign of the function generator output signal.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3107	0	FunctionGenSign	INT8	rw	Y	–1…1	1

7.14.2 Function generator output signal frequency

The function generator output frequency is defined as:

f = <FunctionGenFrequency> (0x3103) · #Factor#

The #Factor# is dependent on the frequency prefix parameter <FunctionGenFrequencyPrefix> (0x3108). The Frequency unit is Hertz [Hz] or [1/s].

7.14.2.1 Object 0x3103: Frequency

This parameter defines the function generator output signals frequency which is multiplied with the frequency prefix factor to get the function generator output signals frequency in Hertz [Hz] or [1/s].

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3103	0	FunctionGenFrequency	UINT16	rw	Y	1…10000	10

7.14.2.2 Object 0x3108: Frequency prefix

This parameter defines a factor which is multiplied with the parameter <FunctionGenFrequency> (0x3103) to get the function generator output signals frequency in Hertz [Hz] or [1/s].

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3108	0	FunctionGenFrequencyPrefix	INT8	rw	Y	–4…0	0

Value description

<FunctionGenFrequencyPrefix>	Resulting #Factor#
0	1.0
–1	0.1
–2	0.01
–3	0.001
–4	0.0001

Table 60: Possible values of parameter <FunctionGenFrequencyPrefix> (0x3108)

7.14.3 Function generator output signals

The function generator output signals can be used for several tasks. An important one is the use as the set-point for the servo valve. This can be done by mapping the output signal to one of the following servo valve setpoint inputs:

- **Spool position setpoint value:** <SplSetpointParameter> (0x3320)
  ⇨ Chapter "6.2.3.2 Object 0x3320: Setpoint parameter", page 54
- **Pressure setpoint value:** <PrsSetpointParameter> (0x3310)
  ⇨ Chapter "6.2.4.2 Object 0x3310: Setpoint parameter", page 56

7.14.3.1 Object 0x3101: Output signal

This is the function generator output signal.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3101	0	FunctionGenOutput	INT16	ro	-	INT16	None

7.14.3.2 Object 0x3102: Square output (Trigger signal)

This signal has the same frequency as the function generator output signal but is fixed to a rectangular shape and with a magnitude of 32767 increments with no offset.

FunctionGenerator							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x3102	0	FunctionGenSquareOutput	INT16	ro	-	INT16	None

Value description

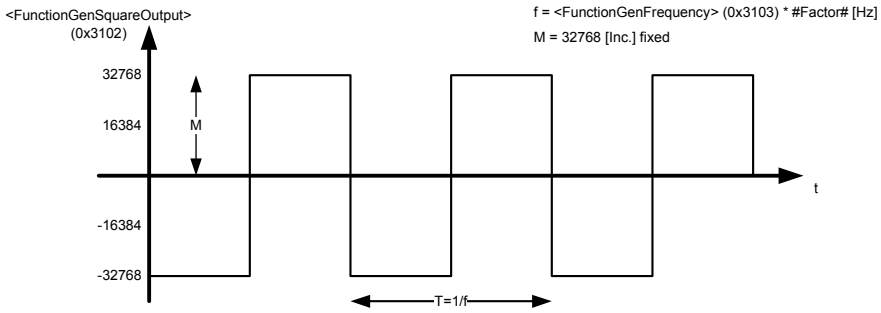


Figure 74: Trigger signal

8 Diagnostics

8.1 Fault reaction

If a malfunction occurs, the servo valve software throws a fault. The corresponding fault is set in the fault state <FaultStatus> (0x2831) and <FaultRetainStatus> (0x2834) parameters in bit coded form. If no fault reaction for this fault is defined within the parameter <FaultReactionType> (0x2830) no further fault reaction is done. If a fault reaction is configured for the thrown fault code at least an emergency message is sent and the <Error-Register> (0x1001) parameter will be set according to the error group of the fault code. Then the fault code and the error code are saved in an array <StandardErrorField> (0x1003) which holds the last eight thrown faults. Afterwards the error description string is saved in the array <FaultReactionDescription> (0x2832). The <FaultHistoryNumber> (0x2833) is set to the number of thrown faults.

If the configured fault reaction of the actual thrown fault requests a change of the device state, the corresponding transition of the device state machine will be forced.

8.1.1 Fault reaction flow chart

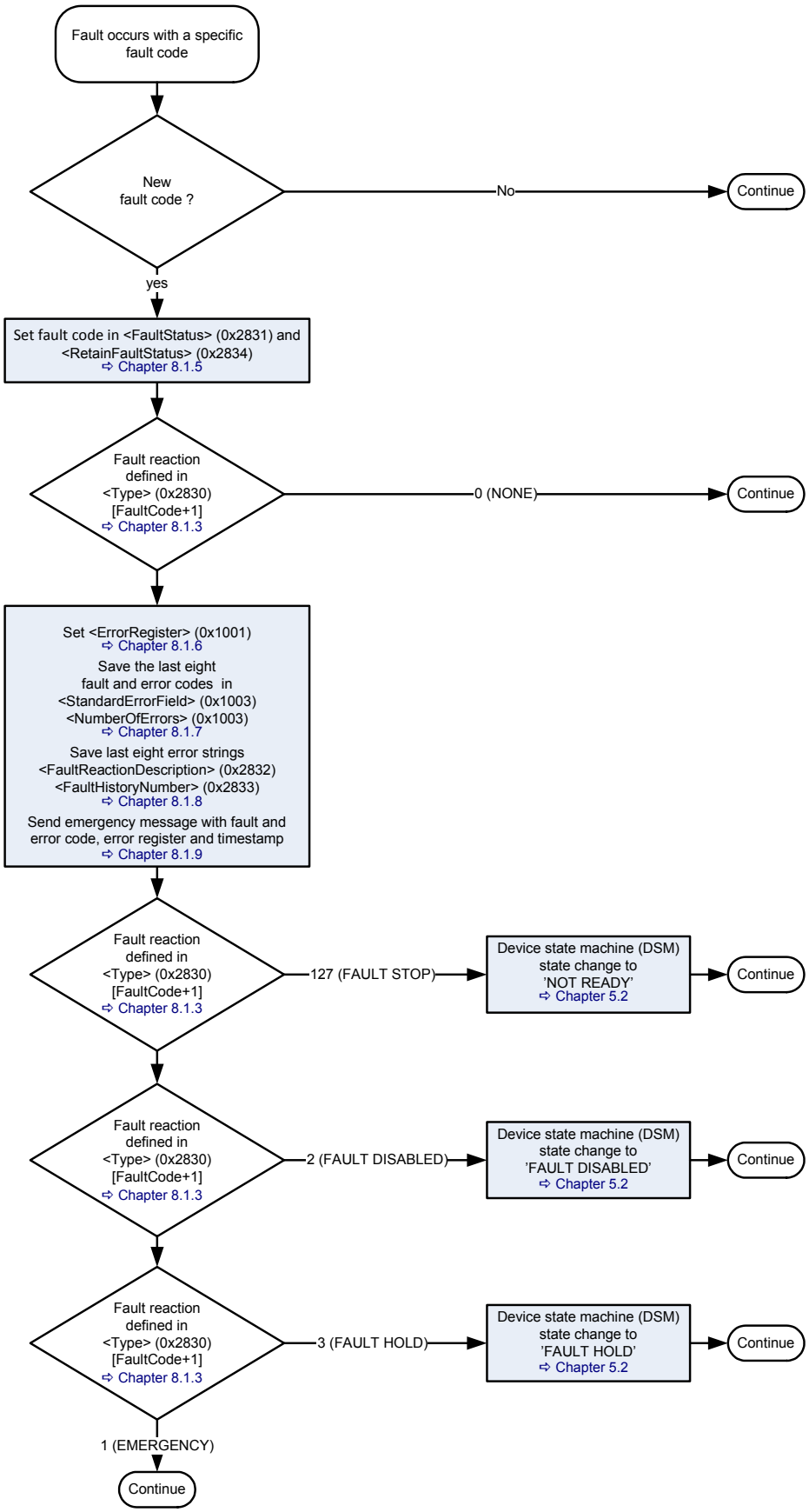


Figure 75: Fault reaction flow chart

8.1.2 Possible fault codes

The possible fault codes are shown in the following table. The used fault codes depend on the servo valve model and hardware. The fault code 0 means that no fault is pending.

To get the sub-index of the fault reaction type <FaultReactionType> (0x2830), the fault code must be incremented by one.

Fault code			Dword		<FaultReactionType> (0x2830)		
Dec.	Hex.	Fault description	Num	Bit	Sub-Index	Value range	De-fault
0	0x00	Error reset or no error	-	-	1	0	0
1	0x01	Error microprocessor core	1	1	2	127	127
2	0x02	Error digital signal processor	1	2	3	127	127
3	0x03	Error DSP program download	1	3	4	127	127
4	0x04	Error DSP realtime data transmission	1	4	5	0...3	2
5	0x05	Power supply voltage too low	1	5	6	0...3	2
6	0x06	Power supply voltage too high	1	6	7	0...3	2
7	0x07	Internal supply voltage too low	1	7	8	127	127
8	0x08	Internal supply voltage too high	1	8	9	127	127
9	0x09	Internal reference voltage too low	1	9	10	127	127
10	0x0A	Internal reference voltage too high	1	10	11	127	127
11	0x0B	Internal current too low	1	11	12	127	127
12	0x0C	Internal current too high	1	12	13	127	127
13	0x0D	Electronics temperature too low (< –20 °C)	1	13	14	0...3	2
14	0x0E	Electronics temperature too high (> 85 °C)	1	14	15	0...3	1
15	0x0F	Electronics temperature exceeded (> 105 °C)	1	15	16	0...3	2
16	0x10	Current sensor circuit failure	1	16	17	127	127
17	0x11	Pilot/single stage LVDT cable break	1	17	18	127	127
18	0x12	Pilot/single stage LVDT position out of range	1	18	19	127	127
19	0x13	Pilot/single stage LVDT circuit failure	1	19	20	127	127
20	0x14	Main stage LVDT cable break	1	20	21	0...3	0
21	0x15	Main stage LVDT position out of range	1	21	22	0...3	0
22	0x16	Main stage LVDT circuit failure	1	22	23	127	127
23	0x17	Internal pressure transducer cable break	1	23	24	127	127
24	0x18	Internal pressure transducer circuit failure	1	24	25	127	127
25	0x19	Internal pressure transducer pressure peak	1	25	26	0...3	0
26	0x1A	Analog input 0 supply cable break/short circuit	1	26	27	0...3	0
27	0x1B	Analog input 1 supply cable break/short circuit	1	27	28	0...3	0
28	0x1C	Analog input 2 supply cable break/short circuit	1	28	29	0...3	0
29	0x1D	Analog input 3 supply cable break/short circuit	1	29	30	0...3	0
30	0x1E	Analog input 4 supply cable break/short circuit	1	30	31	0...3	0
31	0x1F	Analog input 0 current too low (4...20 mA)/ADC overflow (voltage)	1	31	32	0...3	0
32	0x20	Analog input 1 current too low (4...20 mA)/ADC overflow (voltage)	1	32	33	0...3	0
33	0x21	Analog input 2 current too low (4...20 mA)/ADC overflow (voltage)	2	1	34	0...3	0
34	0x22	Analog input 3 current too low (4...20 mA)/ADC overflow (voltage)	2	2	35	0...3	0
35	0x23	Analog input 4 current too low (4...20 mA)/ADC overflow (voltage)	2	3	36	0...3	0
36	0x24	Analog input 0 circuit failure	2	4	37	0...3	0
37	0x25	Analog input 1 circuit failure	2	5	38	0...3	0

Table 61: Possible fault codes (part 1 of 4)

Fault code			Dword		<FaultReactionType> (0x2830)		
Dec.	Hex.	Fault description	Num	Bit	Sub-Index	Value range	De-fault
38	0x26	Analog input 2 circuit failure	2	6	39	0...3	0
39	0x27	Analog input 3 circuit failure	2	7	40	0...3	0
40	0x28	Analog input 4 circuit failure	2	8	41	0...3	0
41	0x29	Encoder channel a cable break	2	9	42	0...3	0
42	0x2A	Encoder channel b cable break	2	10	43	0...3	0
43	0x2B	Encoder channel z cable break	2	11	44	0...3	0
44	0x2C	SSI error	2	12	45	0...3	0
45	0x2D	Power driver	2	13	46	127	127
46	0x2E	Internal random access memory	2	14	47	127	127
47	0x2F	Internal program memory	2	15	48	127	127
48	0x30	Internal nonvolatile memory	2	16	49	127	127
49	0x31	Out of memory error	2	17	50	0...3	2
50	0x32	Software coding	2	18	51	0...3	2
51	0x33	Software reset (watchdog) occurred	2	19	52	0...3	2
52	0x34	Interrupt time exceeded	2	20	53	0...3	2
53	0x35	Task time exceeded	2	21	54	0...3	2
54	0x36	Parameter initialization error	2	22	55	0...3	2
55	0x37	Node identifier data memory corrupted	2	23	56	0...3	2
56	0x38	User data memory corrupted	2	24	57	0...3	2
57	0x39	Restore data memory corrupted	2	25	58	127	127
58	0x3A	Factory data memory corrupted	2	26	59	127	127
59	0x3B	Calibration data memory corrupted	2	27	60	127	127
60	0x3C	Diagnosis data memory corrupted	2	28	61	0...3	0
61	0x3D	Position control monitoring	2	29	62	0...3	0
62	0x3E	Velocity control monitoring	2	30	63	0...3	0
63	0x3F	Force control monitoring	2	31	64	0...3	0
64	0x40	Flow control monitoring	2	32	65	0...3	0
65	0x41	Pressure control monitoring	3	1	66	0...3	0
66	0x42	Current control monitoring	3	2	67	0...3	0
67	0x43	Spool position control monitoring	3	3	68	0...3	2
68	0x44	Trajectory generator processing error	3	4	69	0...3	0
69	0x45	Eventhandler exception	3	5	70	0...3	0
70	0x46	Local CAN general fault	3	6	71	0...3	0
71	0x47	Local CAN buffer overflow	3	7	72	0...3	0
72	0x48	Local CAN in error passive mode	3	8	73	0...3	0
73	0x49	Local CAN recovered from bus-off	3	9	74	0...3	0
74	0x4A	Local CAN RPD01 time out	3	10	75	0...3	0
75	0x4B	Local CAN RPD02 time out	3	11	76	0...3	0
76	0x4C	Local CAN RPD03 time out	3	12	77	0...3	0
77	0x4D	Local CAN RPD04time out	3	13	78	0...3	0
78	0x4E	Local CAN RPD01 data	3	14	79	0...3	0
79	0x4F	Local CAN RPD02 data	3	15	80	0...3	0
80	0x50	Local CAN RPD03 data	3	16	81	0...3	0
81	0x51	Local CAN RPD04 data	3	17	82	0...3	0
82	0x52	Local CAN TPD01 time out	3	18	83	0...3	0

Table 61: Possible fault codes (part 2 of 4)

Fault code			Dword		<FaultReactionType> (0x2830)		
Dec.	Hex.	Fault description	Num	Bit	Sub-Index	Value range	De-fault
83	0x53	Local CAN TPD02 time out	3	19	84	0...3	0
84	0x54	Local CAN TPD03 time out	3	20	85	0...3	0
85	0x55	Local CAN TPD04 time out	3	21	86	0...3	0
86	0x56	Local CAN TPD01 data	3	22	87	0...3	0
87	0x57	Local CAN TPD02 data	3	23	88	0...3	0
88	0x58	Local CAN TPD03 data	3	24	89	0...3	0
89	0x59	Local CAN TPD04 data	3	25	90	0...3	0
90	0x5A	CAN general fault	3	26	91	0...3	0
91	0x5B	CAN buffer overflow	3	27	92	0...3	0
92	0x5C	CAN in error passive mode	3	28	93	0...3	0
93	0x5D	CAN recovered from bus-off	3	29	94	0...3	0
94	0x5E	CAN RPD01 time out	3	30	95	0...3	0
95	0x5F	CAN RPD02 time out	3	31	96	0...3	0
96	0x60	CAN RPD03 time out	3	32	97	0...3	0
97	0x61	CAN RPD04 time out	4	1	98	0...3	0
98	0x62	CAN RPD01 data	4	2	99	0...3	0
99	0x63	CAN RPD02 data	4	3	100	0...3	0
100	0x64	CAN RPD03 data	4	4	101	0...3	0
101	0x65	CAN RPD04 data	4	5	102	0...3	0
102	0x66	CAN TPD01 time out	4	6	103	0...3	0
103	0x67	CAN TPD02 time out	4	7	104	0...3	0
104	0x68	CAN TPD03 time out	4	8	105	0...3	0
105	0x69	CAN TPD04 time out	4	9	106	0...3	0
106	0x6A	CAN TPD01 data	4	10	107	0...3	0
107	0x6B	CAN TPD02 data	4	11	108	0...3	0
108	0x6C	CAN TPD03 data	4	12	109	0...3	0
109	0x6D	CAN TPD04 data	4	13	110	0...3	0
110	0x6E	CAN life guard error or heartbeat error	4	14	111	0...3	0
111	0x6F	CAN SYNC producer time out	4	15	112	0...3	0
112	0x70	CAN SYNC consumer time out	4	16	113	0...3	0
113	0x71	EtherCAT communication fault	4	17	114	0...3	0
114	0x72	EtherCAT RPDO time out	4	18	115	0...3	1
115	0x73	EtherCAT RPDO data	4	19	116	0...3	0
116	0x74	EtherCAT TPDO time out	4	20	117	0...3	0
117	0x75	EtherCAT TPDO data	4	21	118	0...3	0
118	0x76	PROFIBUS general fault	4	22	119	0...3	0
119	0x77	Reserved	4	23	120	unused	unused
120	0x78	Reserved	4	24	121	unused	unused
121	0x79	Reserved	4	25	122	unused	unused
122	0x7A	Reserved	4	26	123	unused	unused
123	0x7B	Reserved	4	27	124	unused	unused

Table 61: Possible fault codes (part 3 of 4)

Fault code			Dword		<FaultReactionType> (0x2830)		
Dec.	Hex.	Fault description	Num	Bit	Sub-Index	Value range	De-fault
124	0x7C	Reserved	4	28	125	unused	unused
125	0x7D	Reserved	4	29	126	unused	unused
126	0x7E	Reserved	4	30	127	unused	unused
127	0x7F	Reserved	4	31	128	unused	unused
128	0x80	Reserved	4	32	129	unused	unused

Table 61: Possible fault codes (part 4 of 4)

8.1.3 Fault reaction type

The fault reaction parameter <FaultReactionType> (0x2830) can be used to configure the fault behavior for each fault code.

8.1.3.1 Object 0x2830: Fault reaction type

The fault reaction for each fault event can be configured by selecting different fault reaction types 0 to 3. The reaction 127 (non-removable error) is predefined. This fault reaction type cannot be changed.

To get the ndex of the fault reaction type <FaultReactionType> (0x2830), the fault code must be incremented by 101.

FaultReaction							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2830	1…128	FaultReactionType	INT8	rw	Y	⇨ Table 61, page 159	

Value description

<FaultReactionType>	Description
0 (NONE)	No fault reaction, error is ignored. Independent whether a malfunction for the monitored fault is detected, the device does not react on this event. The device continues to operate. Special care must be taken, as the malfunction may have an impact on the device.
1 (EMERGENCY)	Send an emergency message. If a malfunction for the monitored fault is detected, an emergency message will be sent onto the field bus. The device continues to operate. Special care must be taken, as the malfunction may have an impact on the device.
2 (FAULT DISABLED)	Switch to device state 'FAULT DISABLED'. If a malfunction for the monitored fault is detected, the device state machine enters the 'FAULT DISABLED' state and an emergency message will be sent onto the field bus. The power stage of the device is switched off, while all device functions are still alive. The device must be re-enabled with the control word or the enable signal in order to return into normal operation.
3 (FAULT_HOLD)	Switch to device state 'FAULT_HOLD'. If a malfunction for the monitored fault is detected, the device enters the 'FAULT_HOLD' state and an emergency message will be sent onto the field bus. The hold setpoint of the device is in effect and adjusted. The device must be re-enabled with the control word or the enable signal in order to return into normal operation.
127 (FAULT STOP)	Switch to device state 'NOT_READY'. If a malfunction for the monitored fault is detected, the device enters the 'NOT_READY' state and an emergency message will be sent onto the field bus. The power stage of the device is switched off, while almost all device functions are stopped. The device must be serviced.
Otherwise	Reserved

Table 62: Fault reaction settings

8.1.4 Error codes depending on fault codes

The following table combines the Moog specific fault codes with the error codes, send with an emergency message.

Details about the error codes used for monitoring: ⇨ Chapter "7.12 Monitoring", page 133

Fault code	Error code defined in Device Profile Fluid Power	Error description
11, 12	2210	Internal current too high or low
16	2211	Over current in external sensor supply
8, 10	3210	Internal voltage too high
7, 9	3220	Internal voltage too low
6	3411	Power supply voltage too high
5	3412	Power supply voltage too low
14, 15	4211	Temperature of electronic components too high
13	4212	Temperature of electronic components too low
24	5211	Internal pressure transducer circuit failure
17, 18, 19	5212	Pilot/single stage actual spool position
36	5213	Analog input 0 circuit failure
37	5214	Analog input 1 circuit failure
38	5215	Analog input 2 circuit failure
39	5216	Analog input 3 circuit failure
40	5217	Analog input 4 circuit failure
22	5218	Main stage actual spool position sensor
1, 2, 3, 4	5220	Microprocessor core
23, 25	5231	Internal pressure transducer cable break / pressure peak
41, 42, 43, 44	5232	Encoder cable break / SSI error
26, 31	5233	Analog input 0: supply cable error / signal out of range
27, 32	5234	Analog input 1: supply cable error / signal out of range
28, 33	5235	Analog input 2: supply cable error / signal out of range
29, 34	5236	Analog input 3: supply cable error / signal out of range
30, 35	5237	Analog input 4: supply cable error / signal out of range
20, 21	5238	Main stage actual spool position out of range / cable break
45	5410	Power driver
46	5510	RAM
47	5520	EPROM
48	5530	EEPROM
51	6010	Software reset (watchdog)
50	6101	Software coding
52	6102	Interrupt time exceeded
53	6103	Task time exceeded
49	6104	Out of memory
69	6201	Event handler
55	6311	Node identifier data
56	6312	User data
57	6313	Restore data
58	6314	Factory data
59	6315	Calibration data
60	6316	Diagnosis data

Table 63: Possible error codes depending on fault codes (part 1 of 3)

Fault code	Error code defined in Device Profile Fluid Power	Error description
54	6320	Parameter error
90, 113, 118	8100	Fieldbus communication
70	8101	Local CAN communication
91	8110	CAN overrun
71	8111	Local CAN overrun (objects lost)
92	8120	CAN in error passive mode
72	8121	Local CAN in error passive mode
110	8130	Life guard error or heartbeat error
93	8140	CAN recovered from bus off
73	8141	Local CAN recovered from bus off
94, 114	8231	RPD01 time out
95	8232	RPD02 time out
96	8233	RPD03 time out
97	8234	RPD04 time out
102, 116	8235	TPD01 time out
103	8236	TPD02 time out
104	8237	TPD03 time out
105	8238	TPD04 time out
111	8239	SYNC producer time out
112	823A	SYNC consumer time out
98, 115	8241	RPD01 data
99	8242	RPD02 data
100	8243	RPD03 data
101	8244	RPD04 data
106, 117	8245	TPD01 data
107	8246	TPD02 data
108	8247	TPD03 data
109	8248	TPD04 data
74	8251	Local RPD01 time out
75	8252	Local RPD02 time out
76	8253	Local RPD03 time out
77	8254	Local RPD04 time out
82	8255	Local TPD01 time out
83	8256	Local TPD02 time out
84	8257	Local TPD03 time out
85	8258	Local TPD04 time out
78	8261	Local RPD01 data
79	8262	Local RPD02 data
80	8263	Local RPD03 data
81	8264	Local RPD04data
86	8265	Local TPD01 data
87	8266	Local TPD02 data
88	8267	Local TPD03 data
89	8268	Local TPD04 data
67	8301	Position control monitoring
65	8302	Pressure control monitoring
61	8303	Position control monitoring

Table 63: Possible error codes depending on fault codes (part 2 of 3)

Fault code	Error code defined in Device Profile Fluid Power	Error description
62	8304	Velocity control monitoring
63	8305	Force control monitoring
64	8306	Flow control monitoring
66	8307	Current control monitoring
68	8308	Trajectory generation

Table 63: Possible error codes depending on fault codes (part 3 of 3)

8.1.5 Fault status

The bit coded fault status indicates which faults are currently reported for the device. Each bit of the fault status array (4 words with 32 bit, built with the sub-indexes 1…4 of the fault status) stands for a fault code. The corresponding fault code is equal to the bit position in the 128 bit field (4x32 bit).

8.1.5.1 Object 0x2831: Fault status

Actual reported faults in bit coded form.

FaultReaction							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2831	1	FaultStatus (fault code 1…31)	UINT32	ro	-	UINT32	None
0x2831	2	FaultStatus (fault code 32…63)	UINT32	ro	-	UINT32	None
0x2831	3	FaultStatus (fault code 64…95)	UINT32	ro	-	UINT32	None
0x2831	4	FaultStatus (fault code 96…128)	UINT32	ro	-	UINT32	None

8.1.5.2 Object 0x2834: Fault retain status

All reported faults in bit coded form since powering on the servo valve.

FaultReaction							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2834	1	FaultRetainStatus (fault code 1…31)	UINT32	rw	N	UINT32	None
0x2834	2	FaultRetainStatus (fault code 32…63)	UINT32	rw	N	UINT32	None
0x2834	3	FaultRetainStatus (fault code 64…95)	UINT32	rw	N	UINT32	None
0x2834	4	FaultRetainStatus (fault code 96…128)	UINT32	rw	N	UINT32	None

8.1.6 Error register

The <ErrorRegister> (0x1001) displays the error information about the last reported fault in bit-coded form. Bit 0 of the <ErrorRegister> (0x1001) is set as soon as an error occurs on the servo valve.

The error codes of older faults are stored in the <PredefinedErrorField> (0x1003).
⇨ Chapter "8.1.7 Last eight fault codes and error codes", page 166

8.1.6.1 Object 0x1001: Error register

ErrorRegister							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1001	0	ErrorRegister	UINT8	ro	-	UINT8	None

This object shall provide error information. The CANopen device maps internal errors into this object. This information is also sent with the emergency object.

Value description

<ErrorRegister>	Description
Bit	
0	Generic error (any error)
1	Current error
2	Voltage error
3	Temperature error
4	Communication error
5	Reserved
6	Reserved
7	Reserved

Table 64: Possible values of parameter <ErrorRegister> (0x1001)

If a specific error occurs, the corresponding bit shall be set to 1_b. The bits are cleared automatically when the error has gone.

8.1.7 Last eight fault codes and error codes

Every time, a fault is thrown which triggered a fault reaction, information about the fault is stored to the <StandardErrorField> (0x1003) parameter array. The <StandardErrorField> (0x1003) parameter array contains a list of up to 8 entries. This error code provides information about the reason of the error. The parameter <NumberOfErrors> (0x1003) holds information about the number of errors currently recorded. Every new error is stored in the first element of the parameter array <StandardErrorField> (0x1003), the older ones move down in the list. If the maximum number of entries is reached and a new fault occurred the oldest fault information will be deleted.

Writing the value 0 to the object <NumberOfErrors> (0x1003), sub-index 0, deletes the entire error code entries.

Only the first eight elements of the parameter array <StandardErrorField> (0x1003), sub-index 1…8 are used.

8.1.7.1 Object 0x1003: Predefined error field

This object contains the last eight error codes, fault codes and the number of recorded errors.

PreDefinedErrorField							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1003	0	NumberOfErrors	UINT32	rw	N	UINT32	None
0x1003	1…16	StandardErrorField	UINT32	ro	-	UINT32	None

Value description

Parameter	Description
<NumberOfErrors>	Number of actual recorded errors.
<StandardErrorField>	Array of recorded errors.

Table 65: Possible values of parameter <PreDefinedErrorField> (0x1003)

<StandardErrorField>				
Byte	3	2	1	0
Description	Additional information		Error code ⇨ Chapter "8.1.4 Error codes depending on fault codes", page 163	
	Reserved	Fault code ⇨ Chapter "8.1.2 Possible fault codes", page 159		

Example

The parameter <StandardErrorField> (0x1003), sub-index 1 holds the decimal value 3167536 (corresponds to 0x305530 hex).
The coding of the value is shown in the following table:

<StandardErrorField>				
Byte	3	2	1	0
Description	Additional information		Error code	
	Reserved	Fault code		
Content	0x00	0x30	0x5530	

Result:

Error code 0x5530: EEPROM error
Fault code 0x30: Internal non-volatile memory

8.1.8 Last eight error message descriptions

The last eight error description strings can be read by the parameter <FaultReactionDescription> (0x2832). The parameter <FaultHistoryNumber> (0x2833) selects one of the last eight error description strings. The newest error description string is shown if the <FaultHistoryNumber> (0x2833) is set to zero and the oldest saved error description string is shown if the <FaultHistoryNumber> (0x2833) is set equal to the parameter <NumberOfErrors> (0x1003).

8.1.8.1 Object 0x2832: Fault reaction description

The parameter <FaultReactionDescription> (0x2832) contains the fault reaction description string depending on the <FaultHistoryNumber> (0x2833) including the fault time in minutes since production of the servo valve. The format is "A/B @ M min STRING". A = displayed fault number, B = count of faults, M = fault time, STRING = description.

FaultReaction							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2832	0	FaultReactionDescription	STRING	ro	-	None	None

8.1.8.2 Object 0x2833: Fault history number

The parameter <FaultHistoryNumber> (0x2833) selects the fault description shown in the parameter <FaultReactionDescription> (0x2832).

FaultReaction							
Index	Sub-index	Name	Data type	Access	Persistence	Value range	Default
0x2833	0	FaultHistoryNumber	UINT8	rw	N	0…7	None

8.1.9 Emergency message

Every time a configured error occurs on the servo valve, it sends an emergency message with error register, error code and timestamp to the master. The emergency message will also be sent if all errors has disappeared. In this case the fault code 0x00 (Error reset or no error) will be sent.

⇨ Chapter "2.7.1 Emergency (EMCY) protocol (0x01)", page 15
⇨ Chapter "8.1.4 Error codes depending on fault codes", page 163

The coding of the emergency message is as follows:

Byte	7	6	5	4	3	2	1	0
Description	Moog-specific error code					Error register	Emergency error code ⇨ Chapter "8.1.4 Error codes depending on fault codes", page 163	
	Power on time in minutes				Fault code ⇨ Chapter "8.1.2 Possible fault codes", page 159			

The displayed byte order of the transmitted emergency message is depending on the field bus master.

Example

In the example below a simple reproducible fault is described. The fault code 0x05 (power supply voltage too low) will occur if the power supply voltage is less than 17 V. The corresponding error code is 0x3412 and the <ErrorRegister> (0x1001) is set to 0x4. The time since power on of the servo valve until the fault occurred is 1000 or in hex 0x03E8 minutes.

Byte	7	6	5	4	3	2	1	0
Description	Moog-specific error code					Error register	Emergency error code	
	Power on time in minutes				Fault code			
Content	0x00	0x00	0x03	0xE8	0x05	0x04	0x34	0x12
Result	0x000003E805043412							

8.1.10 Fault disappears

If all faults has disappeared the <ErrorRegister> (0x1001) and the <FaultStatus> (0x2831) are set to zero. To confirm that no faults are present, the error code 0x00 (Error reset or no error) will be sent via an emergency message to the field bus master. If the device state machine (DSM) is in the state 'FAULT HOLD', 'FAULT DISABLED' or 'NOT READY', the DSM must be set to 'ACTIVE' again. This can be done by the #ControlWord# or the enable signal (digital input 0).

⇨ Chapter "8.1.11 Fault acknowledgement", page 169

8.1.11 Fault acknowledgement

Depending on the configured fault reaction, the servo valve sends out an emergency message and changes the device state machine to the corresponding fault state.

⇨ Chapter "8.1.3 Fault reaction type", page 162

In order to get out of the fault state the fault must be acknowledged. This can be achieved by

- Sending the #ControlWord# to the servo valve with the bit 3 (fault reset) is set.
- Toggling the enable signal (digital input 0).

⇨ Chapter "5.1 Local mode", page 37
⇨ Chapter "5.2 Device state machine (DSM)", page 40
⇨ Chapter "6.7.2 Object 0x5E41: Digital output type", page 77

If the fault is not fixed or other faults are still present, the servo valve will fall back into the DSM state defined by the fault reaction type <FaultReactionType> (0x2830).

8.2 Internal errors

The following parameters store information which could assist to debug software malfunctions. The user may be asked to pass these values to our service personal in order to identify software malfunctions.

8.2.1 Object 0x2822: Internal error code

This object contains the first five occurred internal error codes since the firmware reset.

ErrorHandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2822	1…5	InternalErrorCode	UINT32	ro	-	UINT32	0

8.2.2 Object 0x2823: Internal error time

The time stamp of the occurrence of the last five error codes were saved in this array. The time stamps were stored in minutes since power on.

ErrorHandler							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x2823	1…5	InternalErrorTime	UINT32	ro	-	UINT32	0

For your notes.

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 172
- 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 factory parameters", page 173

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.

		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.

Table 66: Behavior of saveable and volatile parameters

The write access to any parameter will not affect its default value.

9.1 Storing parameters

The current values of all parameters declared as non-volatile (persistence = "Y") can be stored in a non-volatile memory on the servo valve.

9.1.1 Object 0x1010: Store parameters

Storing is proceeded by writing the signature 0x65766173 ("save") to one of the following parameters.

StoreParameters							
Index	Sub-index	Parameter name	Data type	Access	Persistence	Value range	Default
0x1010	1	SaveAllParameters	UINT32	rw	N	UINT32	None
0x1010	2	SaveCommunicationParameters	UINT32	rw	N	UINT32	None
0x1010	3	SaveApplicationParameters	UINT32	rw	N	UINT32	None
0x1010	4	SaveManufacturerDefinedParameters	UINT32	rw	N	UINT32	None

Value description

Parameter	**Description**
<SaveAllParameters>	Saves all non-volatile parameters in the servo valve's non-volatile memory.
<SaveCommunicationParameters>	Saves all non-volatile communication parameters (index range 0x1000…0x1FFF) in the servo valve's non-volatile memory.
<SaveApplicationParameters>	Saves all non-volatile application parameters (index range 0x6000…0x9FFF) in the servo valve's non-volatile memory.
<SaveManufacturerDefinedParameters>	Saves all non-volatile manufacturer-defined parameters (index range 0x2000…0x5FFF) in the servo valve's non-volatile memory.

Table 67: Possible values of parameter 0x1010

9.2 Restoring factory parameters

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	None
0x1011	2	RestoreCommunicationDefaultParameters	UINT32	rw	N	UINT32	None
0x1011	3	RestoreApplicationDefaultParameters	UINT32	rw	N	UINT32	None
0x1011	4	RestoreManufacturerDefinedDefaultParameters	UINT32	rw	N	UINT32	None

Value description

Parameter	**Description**
<RestoreAllDefaultParameters>	Restores the factory settings for all parameters in the servo valve.
<RestoreCommunicationDefaultParameters>	Restores all communication parameters (index range 0x1000…0x1FFF) in the servo valve.
<RestoreApplicationDefaultParameters>	Restores all application parameters (index range 0x6000…0x9FFF) in the servo valve.
<RestoreManufacturerDefinedDefaultParameters>	Restores all manufacturer-defined parameters (index range 0x2000…0x5FFF) in the servo valve.

Table 68: Possible values of parameter 0x1011

After restoring the factory parameters the valve will generate a new start to get the factory setting values effective.

⇨ Chapter "5.2 Device state machine (DSM)", page 40
⇨ Chapter "2.14 EtherCAT® network state machine (ESM)", page 26

The following table shows the necessary state changes to activate the restored values.

Parameter	**Set the device state machine to 'INIT' #ControlWord# MHD = 000**	**Set the field bus network state machine to 'INIT'**
<RestoreAllDefaultParameters>	X	X
<RestoreCommunicationDefaultParameters>		X
<RestoreApplicationDefaultParameters>	X	
<RestoreManufacturerDefinedDefaultParameters>	X	

Table 69: State changes needed to activate the restored values

For your notes.

10 Object dictionary

WARNING

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

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x0002	0	Y	dums08	CiA 301	DataType, Signed one byte integer INTEGER8	INT8	rw	N	INT8	None
0x0003	0	Y	dums16	CiA 301	DataType, Signed two byte integer INTEGER16	INT16	rw	N	INT16	None
0x0004	0	Y	dums32	CiA 301	DataType, Signed four byte integer INTEGER32	INT32	rw	N	INT32	None
0x0005	0	Y	dumu08	CiA 301	DataType, Unsigned one byte integer UNSIGNED8	UINT8	rw	N	UINT8	None
0x0006	0	Y	dumu16	CiA 301	DataType, Unsigned two byte integer UNSIGNED16	UINT16	rw	N	UINT16	None
0x0007	0	Y	dumu32	CiA 301	DataType, Unsigned four byte integer UNSIGNED32	UINT32	rw	N	UINT32	None
0x0008	0	Y	dumf32	CiA 301	DataType, Float32 Float32	FLOAT32	rw	N	FLOAT32	None
0x0009	0	N	dumchr	CiA 301	DataType, Visible_String Visible_String	STRING	rw	N	None	None
0x1000	0	N	devtyp	CiA 301	Device, Type DeviceType	UINT32	ro	-	UINT32	408
0x1001	0	Y	errreg	CiA 301	Device, Error register ErrorRegister	UINT8	ro	-	UINT8	None
0x1002	0	Y	manstsreg	CiA 301	Device, ManufacturerStatusRegister ManufacturerStatusRegister	UINT32	ro	-	UINT32	None
0x1003	0	N	preerrnum	CiA 301	Device, Predefined error field NumberOfErrors	UINT32	rw	N	UINT32	None
0x1003	1…16	N	preerrfld[0…15]	CiA 301	Device, Predefined error field StandardErrorField	UINT32	ro	-	UINT32	None

Table 70: Object dictionary (part 1 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x1008	0	N	mandevnam	CiA 301	Device, Manufacturer device name ManufacturerDeviceName	STRING(64)	ro	-	None	None
0x1009	0	N	manhdwver	CiA 301	Device, Manufacturer hardware version ManufacturerHardwareVersion	STRING(64)	ro	-	None	None
0x100A	0	N	mansfwver	CiA 301	Device, Manufacturer software version ManufacturerSoftwareVersion	STRING(64)	ro	-	None	None
0x1010	1	N	stopar[0]	CiA 301	Device, Store parameters SaveAllParameters	UINT32	rw	N	UINT32	1
0x1010	2	N	stopar[1]	CiA 301	Device, Store parameters SaveCommunicationParameters	UINT32	rw	N	UINT32	1
0x1010	3	N	stopar[2]	CiA 301	Device, Store parameters SaveApplicationParameters	UINT32	rw	N	UINT32	1
0x1010	4	N	stopar[3]	CiA 301	Device, Store parameters SaveManufacturerParameters	UINT32	rw	N	UINT32	1
0x1011	1	N	rstpar[0]	CiA 301	Device, Restore factory settings RestoreAllParasToFactorySettings	UINT32	rw	N	UINT32	1
0x1011	2	N	rstpar[1]	CiA 301	Device, Restore factory settings RestoreCommunParasToFactorySettings	UINT32	rw	N	UINT32	1
0x1011	3	N	rstpar[2]	CiA 301	Device, Restore factory settings RestoreApplicationParasToFactorySettings	UINT32	rw	N	UINT32	1
0x1011	4	N	rstpar[3]	CiA 301	Device, Restore factory settings RestoreManufacturerParasToFactorySettings	UINT32	rw	N	UINT32	1
0x1018	1	N	ideobj[0]	CiA 301	Device, Identity object VendorId	UINT32	ro	-	UINT32	40
0x1018	2	N	ideobj[1]	CiA 301	Device, Identity object ProductCode	UINT32	ro	-	UINT32	0
0x1018	3	N	ideobj[2]	CiA 301	Device, Identity object RevisionNumber	UINT32	ro	-	UINT32	0
0x1018	4	N	ideobj[3]	CiA 301	Device, Identity object SerialNumber	UINT32	ro	-	UINT32	0
0x1600	0	N	pdrmapnum	CiA 301	EtherCAT, RxPDO mapping RPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	3
0x1600	1…8	N	pdrmap[0…7]	CiA 301	EtherCAT, RxPDO mapping RPdo1_ApplicPara1…8	UINT32	rw	Y	UINT32	…
0x1A00	0	N	pdtmapnum	CiA 301	EtherCAT, TxPDO mapping TPdo1_NumberOfMappedApplicParaInPdo	UINT8	rw	Y	0…8	3
0x1A00	1…8	N	pdtmap[0…7]	CiA 301	EtherCAT, TxPDO mapping TPdo1_ApplicPara1…8	UINT32	rw	Y	UINT32	…

Table 70: Object dictionary (part 2 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x1C00	0	N	smctnchn	IEC 61158-x-12	EtherCAT, SM communication type SM_CommunicationType_NoOfChannels	UINT8	ro	-	UINT8	4
0x1C00	1	N	smct[0]	IEC 61158-x-12	EtherCAT, SM communication type SM0_Communication_Type	UINT8	ro	-	UINT8	1
0x1C00	2	N	smct[1]	IEC 61158-x-12	EtherCAT, SM communication type SM1_Communication_Type	UINT8	ro	-	UINT8	2
0x1C00	3	N	smct[2]	IEC 61158-x-12	EtherCAT, SM communication type SM2_Communication_Type	UINT8	ro	-	UINT8	3
0x1C00	4	N	smct[3]	IEC 61158-x-12	EtherCAT, SM communication type SM3_Communication_Type	UINT8	ro	-	UINT8	4
0x1C10	0	N	smc0pdrnum	IEC 61158-x-12	EtherCAT, SM 0 receive PDO assignment SM0_MBX_Receive_NoOfAssignedPDOs	UINT8	ro	-	0…4	0
0x1C11	0	N	smc1pdtnum	IEC 61158-x-12	EtherCAT, SM 1 transmit PDO assignment SM1_MBX_Send_NoOfAssignedPDOs	UINT8	ro	-	0…4	0
0x1C12	0	N	smc2pdrnum	IEC 61158-x-12	EtherCAT, SM 2 receive PDO assignment SM2_NoOfAssignedReceivePDOs	UINT8	ro	-	0…4	1
0x1C12	1	N	smc2pdr[0]	IEC 61158-x-12	EtherCAT, SM 2 receive PDO assignment SM2_receive_Pdo1_Assignment	UINT16	ro	-	UINT16	0x1600
0x1C12	2	N	smc2pdr[1]	IEC 61158-x-12	EtherCAT, SM 2 receive PDO assignment SM2_receive_Pdo2_Assignment	UINT16	ro	-	UINT16	0
0x1C12	3	N	smc2pdr[2]	IEC 61158-x-12	EtherCAT, SM 2 receive PDO assignment SM2_receive_Pdo3_Assignment	UINT16	ro	-	UINT16	0
0x1C12	4	N	smc2pdr[3]	IEC 61158-x-12	EtherCAT, SM 2 receive PDO assignment SM2_receive_Pdo4_Assignment	UINT16	ro	-	UINT16	0
0x1C13	0	N	smc3pdtnum	IEC 61158-x-12	EtherCAT, SM 3 transmit PDO assignment SM3_NoOfAssignedTransmitPDOs	UINT8	ro	-	0…4	1
0x1C13	1	N	smc3pdt[0]	IEC 61158-x-12	EtherCAT, SM 3 transmit PDO assignment SM3_transmit_Pdo1_Assignment	UINT16	ro	-	UINT16	0x1A00
0x1C13	2	N	smc3pdt[1]	IEC 61158-x-12	EtherCAT, SM 3 transmit PDO assignment SM3_transmit_Pdo2_Assignment	UINT16	ro	-	UINT16	0
0x1C13	3	N	smc3pdt[2]	IEC 61158-x-12	EtherCAT, SM 3 transmit PDO assignment SM3_transmit_Pdo3_Assignment	UINT16	ro	-	UINT16	0
0x1C13	4	N	smc3pdt[3]	IEC 61158-x-12	EtherCAT, SM 3 transmit PDO assignment SM3_transmit_Pdo4_Assignment	UINT16	ro	-	UINT16	0
0x200A	0	N	iopbkpldver	Moog DCV	Software_PiggyBack, ManufactutrerIOPiggybackVersion ManufactutrerIOPiggybackVersion	UINT16	ro	-	UINT16	0
0x200F	0	N	pwrdly	Moog DCV	Device,PowerOnDelay PowerOnDelay	UINT8	rw	Y	0…10	0

Table 70: Object dictionary (part 3 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x2107	1	Y	hldtrg	Moog DCV	HoldPressureControl, HoldPressureTrigger HoldPressureTrigger	INT16	rw	Y	INT16	16384
0x2107	2	N	hldprsuni	Moog DCV	HoldPressureControl, HoldPressureTrigger Unit	UINT8	ro	-	UINT8	0
0x2107	3	N	hldprsprf	Moog DCV	HoldPressureControl, HoldPressureTrigger Prefix	INT8	ro	-	INT8	0
0x2108	1	Y	hldprsset	Moog DCV	HoldPressureControl, PressureSetpoint PressureSetpoint	INT16	rw	Y	INT16	0
0x2108	2	N	hldprsuni	Moog DCV	HoldPressureControl, PressureSetpoint Unit	UINT8	ro	-	UINT8	0
0x2108	3	N	hldprsprf	Moog DCV	HoldPressureControl, PressureSetpoint Prefix	INT8	ro	-	INT8	0
0x2109	1	Y	hldsplset	Moog DCV	HoldPressureControl, SpoolPositionSetpoint SpoolPositionSetpoint	INT16	rw	Y	INT16	16384
0x2109	2	N	hldspluni	Moog DCV	HoldPressureControl, SpoolPositionSetpoint Unit	UINT8	ro	-	UINT8	0
0x2109	3	N	hldsplprf	Moog DCV	HoldPressureControl, SpoolPositionSetpoint Prefix	INT8	ro	-	INT8	0
0x2120	0	N	prslekfct	Moog DCV	PumpController, LeakageCompensation LeakageCompensation	FLOAT32	rw	Y	FLOAT32	0.0
0x2141	1…16	N	parctlwrd[0…15]	Moog DCV	AnalogParameterSetSwitching, ControlWord ControlWord	UINT16	rw	Y	1...65535	1
0x2142	1…16	N	parprsset-num[0…15]	Moog DCV	AnalogParameterSetSwitching, PressureSetSelector PressureSetSelector	UINT8	rw	Y	1...16	1
0x2143	0	N	parsetitf	Moog DCV	AnalogParameterSetSwitching, SourceTransducer SourceTransducer	UINT8	rw	Y	1...4	4
0x2145	0	N	parsetena	Moog DCV	AnalogParameterSetSwitching, Enable Enable	UINT8	rw	Y	0...1	0
0x2146	0	N	parsetnum	Moog DCV	AnalogParameterSetSwitching, ActiveParameterSet ActiveParameterSet	UINT8	ro	-	0...16	0
0x2147	0	N	parhybflw	Moog DCV	AnalogParameterSetSwitching, HybridFlow HybridFlow	INT16	rw	Y	INT16	0
0x2148	1…16	N	parsetmod[0…15]	Moog DCV	AnalogParameterSetSwitching, Mode Mode	UINT8	rw	Y	0...1	0
0x2149	0	N	stgitfnum	Moog DCV	MainStageControl, MainStageInterface MainStageInterface	UINT8	rw	Y	0...4	0
0x2158	0	Y	stgposout	Moog DCV	ValveMainStageControl, ControllerOutput	INT16	ro	-	INT16	None

Table 70: Object dictionary (part 4 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x215B	1	Y	stgsplval	Moog DCV	MainStageControl, MainStagePositionActualValue MainStagePositionActualValue	INT16	ro	-	INT16	None
0x215C	0	N	stgposcusogn	Moog DCV	MainStageControl, MainStageCustomerOverallGain MainStageCustomerOverallGain	FLOAT32	rw	Y	0.0...2.0	1.0
0x2160	1	Y	rpmval	Moog DCV	ValvePressureControl, MotorRevolutionsPerMinute_ActualValue MotorRevolutionsPerMinute_ActualValue	INT16	rw	N	10...32767	<MotorRevo-lutionsPerMinute_Default> (0x2166)
0x2160	2	N	rpmuni	Moog DCV	ValvePressureControl, MotorRevolutionsPerMinute_ActualValue MotorRevolutionsPerMinute_Unit	UINT8	ro	-	UINT8	0
0x2160	3	N	rpmprf	Moog DCV	ValvePressureControl, MotorRevolutionsPerMinute_ActualValue MotorRevolutionsPerMinute_Prefix	INT8	ro	-	INT8	0
0x2161	0	N	rpmitfnum	Moog DCV	ValvePressureControl, MotorRevolutionsPerMinute_InterfaceNumber MotorRevolutionsPerMinute_InterfaceNumber	INT8	rw	Y	-128...4	0
0x2164	0	N	rpmptm	Moog DCV	ValvePressureControl, MotorRevolutionsPerMinute_TimeConstant MotorRevolutionsPerMinute_TimeConstant	FLOAT32	rw	Y	FLOAT32	0.0
0x2166	1	N	rpmdef	Moog DCV	ValvePressureControl, MotorRevolutionsPerMinute_Default MotorRevolutionsPerMinute_Default	INT16	rw	Y	1...32767	16384
0x2166	2	N	rpmuni	Moog DCV	ValvePressureControl, MotorRevolutionsPerMinute_Default MotorRevolutionsPerMinute_Unit	UINT8	ro	-	UINT8	0
0x2166	3	N	rpmprf	Moog DCV	ValvePressureControl, MotorRevolutionsPerMinute_Default MotorRevolutionsPerMinute_Prefix	INT8	ro	-	INT8	0
0x21A1	0	N	pmpflstim	Moog DCV	PumpController, FlushingTime FlushingTime	UINT16	rw	Y	0...180	180
0x21A4	0	Y	splsetmst	Moog DCV	PumpController, SpoolSetPointFromMaster SpoolSetPointFromMaster	INT16	rw	N	INT16	0
0x21A5	0	N	pmpmstslv	Moog DCV	PumpController, MasterSlaveSelector MasterSlaveSelector	INT8	rw	Y	-2...3	0
0x21A7	0	Y	splfwdmst	Moog DCV	PumpController, SpoolFeedForwardFromMaster SpoolFeedForwardFromMaster	INT16	rw	N	INT16	0
0x21AA	0	Y	splfwdmstpgn	Moog DCV	PumpController, SpoolFeedForwardFromMasterProportionalGain SpoolFeedForwardFromMasterProportionalGain	FLOAT32	rw	Y	FLOAT32	0.0
0x2304	1…16	Y	cmpprspgn[0…15]	Moog DCV	ValvePressureControl, Proportional gain ProportionalGain	FLOAT32	rw	Y	0.0…+inf	0.0
0x2305	1…16	Y	cmpprsign[0…15]	Moog DCV	ValvePressureControl, Integrator gain INTegratorGain	FLOAT32	rw	Y	0.0…+inf	0.0
0x2307	1…16	Y	cmpprsicr[0…15]	Moog DCV	ValvePressureControl, IntegratorControlRange IntegratorControlRange	INT16	rw	Y	0...32767	163

Table 70: Object dictionary (part 5 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x2308	1…16	Y	cmpprsdgn[0…15]	Moog DCV	ValvePressureControl, Differentiator gain DifferentiatorGain	FLOAT32	rw	Y	0.0…+inf	0.0
0x2309	1…16	Y	cmpprsdtm[0…15]	Moog DCV	ValvePressureControl, Differentiator T1 DifferentiatorT1	FLOAT32	rw	Y	0.0…+inf	0.0
0x230A	1…16	Y	cmpprsupp[0…15]	Moog DCV	ValvePressureControl, Upper output limit UpperOutputLimit	INT16	rw	Y	<LowerOutput-Limit>(0x230B[1])…32767	32767
0x230B	1…16	Y	cmpprslow[0…15]	Moog DCV	ValvePressureControl, Lower output limit LowerOutputLimit	INT16	rw	Y	-32768…<UpperOutput-Limit>(0x230A[1])	-32768
0x230D	1…16	N	cmpprsitf[0…15]	Moog DCV	ValvePressureControl, Active transducer interface area A ActiveTransducerInterfaceAreaA	INT8	rw	Y	1…4	1
0x230F	1…16	N	cmpprsitfb[0…15]	Moog DCV	ValvePressureControl, Active transducer interface area B ActiveTransducerInterfaceAreaB	INT8	rw	Y	0…4	0
0x2310	0	Y	cmpprsint	Moog DCV	ValvePressureControl, Integrator part IntegratorPart	FLOAT32	ro	-	FLOAT32	0.0
0x2311	0	Y	cmpprspro	Moog DCV	ValvePressureControl, Proportional part ProportionalPart	FLOAT32	ro	-	FLOAT32	0.0
0x2312	0	Y	cmpprsdt1	Moog DCV	ValvePressureControl, Differantial part DifferantialPart	FLOAT32	ro	-	FLOAT32	0.0
0x2313	0	N	cmpprsrpm	Moog DCV	PumpPressureControl, MotorRevolutionsPerMinute_last_scaled MotorRevolutionsPerMinute_last_scaled	FLOAT32	ro	-	FLOAT32	1.0
0x2314	1…16	Y	cmpprs-dgndwn[0…15]	Moog DCV	PumpPressureControl, DifferentiatorGainDecompress DifferentiatorGainDecompress	FLOAT32	rw	Y	0.0...+inf	0.0
0x2315	1…2	Y	cmpprsspl[0…1]	Moog DCV	PumpPressureControl, SpoolPositionFeedbackPart SpoolPositionFeedbackPart	FLOAT32	ro	-	FLOAT32	0.0
0x2316	1…16	Y	cmpprssgn[0…15]	Moog DCV	PumpPressureControl, SpoolPositionFeedbackGain SpoolPositionFeedbackGain	FLOAT32	rw	Y	0.0...+inf	0.0
0x2317	1…16	Y	cmpprslim[0…15]	Moog DCV	PumpPressureControl, SuckLimitationForPD SuckLimitationForPD	INT16	rw	Y	-32768...0	-16384
0x2318	1…16	Y	cmp-prshpssgn[0…15]	Moog DCV	PumpPressureControl, SpoolPositionFeedbackGainHighPassFiltered SpoolPositionFeedbackGainHighPassFiltered	FLOAT32	rw	Y	0.0...+inf	0.0
0x2319	1…16	Y	cmpprshps-dtm[0…15]	Moog DCV	PumpPressureControl, SpoolPositionFeedbackHighPassTimeConstant SpoolPositionFeedbackHighPassTimeConstant	FLOAT32	rw	Y	0.0..+inf	0.0
0x2324	1…16	Y	cmpprsfb-spgn[0…15]	Moog DCV	PumpPressureControl, SpoolPositionFeedForwardGain SpoolPositionFeedForwardGain	FLOAT32	rw	Y	0...+inf	0.0
0x2325	0	Y	cmpprsfbsout	Moog DCV	PumpPressureControl, SpoolPositionFeedForwardOut SpoolPositionFeedForwardOut	FLOAT32	ro	-	FLOAT32	0.0
0x2326	1	N	hydcapref	Moog DCV	PumpPressureControl, HydraulicCapacityReference HydraulicCapacityReference	INT16	rw	Y	1...32767	16384

Table 70: Object dictionary (part 6 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x2326	2	N	hydcapuni	Moog DCV	PumpPressureControl, HydraulicCapacityReference HydraulicCapacityUnit	UINT8	ro	-	UINT8	0
0x2326	3	N	hydcapprf	Moog DCV	PumpPressureControl, HydraulicCapacityReference HydraulicCapacityPrefix	INT8	ro	-	INT8	0
0x2327	1	Y	hydcapval	Moog DCV	PumpPressureControl, HydraulicCapacityValue HydraulicCapacityValue	INT16	rw	Y	1...32767	<HydraulicCapacityReference> (0x2326)
0x2327	2	N	hydcapuni	Moog DCV	PumpPressureControl, HydraulicCapacityValue HydraulicCapacityUnit	UINT8	ro	-	UINT8	0
0x2327	3	N	hydcapprf	Moog DCV	PumpPressureControl, HydraulicCapacityValue HydraulicCapacityPrefix	INT8	ro	-	INT8	0
0x2329	0	N	cmpprsfbsoer	Moog DCV	PumpPressureControl, SpoolPositionFeedForwardObserverError SpoolPositionFeedForwardObserverError	FLOAT32	ro	-	FLOAT32	None
0x232A	0	N	cmpprsfbsdt1	Moog DCV	PumpPressureControl, SpoolPositionFeedForwardDerivative SpoolPositionFeedForwardDerivative	FLOAT32	ro	-	FLOAT32	None
0x232D	1…16	Y	cmpprsarerat[0…15]	Moog DCV	PumpPressureControl, Pressure_Area_Ratio Pressure_Area_Ratio	FLOAT32	rw	Y	FLOAT32	0.0
0x2350	0	Y	prssetnum	Moog DCV	ValvePressureControl, Active parameter set number ActiveParameterSetNumber	UINT8	rw	Y	1…16	1
0x23F0	1…4	N	prsfltbi[0…3]	Moog DCV	ValvePressureControl, PressureValueFilterBCoeff PressureValueFilterBCoeff	FLOAT32	rw	Y	FLOAT32	0.0
0x23F1	1…4	N	prsfltai[0…3]	Moog DCV	ValvePressureControl, PressureValueFilterACoeff PressureValueFilterACoeff	FLOAT32	rw	Y	FLOAT32	0.0
0x23F2	0	N	cmpprsflt	Moog DCV	ValvePressureControl, Actual pressure filter cutoff frequency ActualPressureFilterCutoffFrequency	FLOAT32	rw	Y	0.0…3333.3	0.0
0x23F3	0	N	prsfltord	Moog DCV	ValvePressureControl, Actual pressure filter order ActualPressureFilterOrder	UINT8	rw	Y	1…3	1
0x2418	0	Y	cmpprsout	Moog DCV	ValvePressureControl, Controller output ControllerOutput	INT16	ro	-	INT16	None
0x241F	0	N	cmpsplcusogn	Moog DCV	ValvePositionControl, CustomerOverallGain CustomerOverallGain	FLOAT32	rw	Y	0.0...2.0	1.0
0x2420	0	N	faisaftyp	Moog DCV	ValveDigitalOutputType, Digital output type DigitalOutput1Type	INT8	ro	-	0…4	2
0x2421	0	N	faisafupp	Moog DCV	ValveFailSafeWindowMonitoring, Upper limit UpperLimit	INT16	ro	-	<LowerLimit> (0x2422)…32767	16384
0x2422	0	N	faisaflow	Moog DCV	ValveFailSafeWindowMonitoring, Lower limit LowerLimit	INT16	ro	-	-32768…<UpperLimit> (0x2421)	-16384

Table 70: Object dictionary (part 7 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x2600	0	Y	pmppwrmax	Moog DCV	PumpController, Power_Maximum Power_Maximum	INT16	rw	Y	0.0...(1<<14)	
0x2601	0	N	pmppwrpgn	Moog DCV	PumpController, Power_ProportionalGain Power_ProportionalGain	FLOAT32	rw	Y	0.0...+inf	
0x2602	0	N	pmppwrpt1	Moog DCV	PumpController, Power_PT1Gain Power_PT1Gain	FLOAT32	rw	Y	0.0...+inf	
0x2603	0	N	pmppwrptm	Moog DCV	PumpController, Power_PT1TimeConstant Power_PT1TimeConstant	FLOAT32	rw	Y	0.0...+inf	
0x2604	0	N	pmppwrpts	Moog DCV	PumpController, Power_PT1Shift Power_PT1Shift	INT16	rw	Y	0...32767	
0x2605	0	N	pmppwrdgn	Moog DCV	PumpController, Power_DifferentialGain Power_DifferentialGain	FLOAT32	rw	Y	0.0...+inf	
0x2606	0	N	pmppwrdtm	Moog DCV	PumpController, Power_DifferentialTimeConstant Power_DifferentialTimeConstant	FLOAT32	rw	Y	0.0...+inf	
0x2608	0	Y	pwrval	Moog DCV	PumpController, PowerValue PowerValue	INT16	ro	-	INT16	
0x2803	0	Y	cpusup	Moog DCV	Hardware_DiagnosticData, Cpu supply voltage CpuSupplyVoltage	UINT16	ro	-	UINT16	None
0x2804	0	Y	pwrsup	Moog DCV	Hardware_DiagnosticData, Power supply voltage PowerSupplyVoltage	UINT16	ro	-	UINT16	None
0x2805	0	Y	pcbtmp	Moog DCV	Hardware_DiagnosticData, PCB temperature PcbTemperature	INT16	ro	-	INT16	None
0x280D	1	N	oprtim[0]	Moog DCV	Hardware_DiagnosticData, Operating time PowerOnTime	UINT32	ro	-	UINT32	0
0x280D	2	N	oprtim[1]	Moog DCV	Hardware_DiagnosticData, Operating time OperatingTime	UINT32	ro	-	UINT32	0
0x2822	1…5	N	errval[0…4]	Moog DCV	ErrorHandler, Internal error code InternalErrorCode	UINT32	ro	-	UINT32	…
0x2823	1…5	N	errtim[0…4]	Moog DCV	ErrorHandler, Internal error time InternalErrorTime	UINT32	ro	-	UINT32	…
0x2830	1…119	N	faurea[0…118]	Moog DCV	FaultReaction, Fault reaction type FaultReactionType	INT8	rw	Y	INT8	…
0x2831	1…4	Y	fausts[0…3]	Moog DCV	FaultReaction, Fault reaction status FaultStatus	UINT32	ro	-	UINT32	…
0x2832	0	N	faudsc	Moog DCV	FaultReaction, Fault reaction description FaultReactionDescription	STRING(64)	ro	-	None	None
0x2833	0	N	fauhis	Moog DCV	FaultReaction, Fault history number FaultHistoryNumber	UINT8	rw	N	0…7	None

Table 70: Object dictionary (part 8 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x2834	1…4	Y	faustsret[0…3]	Moog DCV	FaultReaction, Fault reaction retain status FaultRetainStatus	UINT32	rw	N	UINT32	…
0x3000	0	N	pwd	Moog DCV	Can, Password Password	UINT32	rw	N	UINT32	
0x3002	0	N	ethmodide	Moog DCV	EtherCAT, ModuleIdentifier/StationAlias ModuleIdentifier/StationAlias	UINT16	rw	Y	0...65535	
0x3100	0	Y	fcntyp	Moog DCV	FunctionGenerator, Type FunctionGenType	INT8	rw	N	0…5	0
0x3101	0	Y	fcndem	Moog DCV	FunctionGenerator, Output signal FunctionGenOutput	INT16	ro	-	INT16	0
0x3102	0	Y	fcnsqr	Moog DCV	FunctionGenerator, Trigger signal FunctionGenSquareOutput	INT16	ro	-	INT16	0
0x3103	0	Y	fcntim	Moog DCV	FunctionGenerator, Frequency FunctionGenFrequency	UINT16	rw	N	1…10000	10
0x3104	0	Y	fcnmag	Moog DCV	FunctionGenerator, Magnitude FunctionGenMagnitude	INT16	rw	N	0…32767	0
0x3105	0	Y	fcnofs	Moog DCV	FunctionGenerator, Offset FunctionGenOffset	INT16	rw	N	INT16	0
0x3107	0	Y	fcnsgn	Moog DCV	FunctionGenerator, Sign FunctionGenSign	INT8	rw	N	-1…1	1
0x3108	0	Y	fcnprf	Moog DCV	FunctionGenerator, Frequency prefix FunctionGenFrequencyPrefix	INT8	rw	N	-4…0	0
0x3180	0	N	dlgctl	Moog DCV	DataLogger, Control Control	UINT8	rw	N	0…1	0
0x3181	0	Y	dlgsts	Moog DCV	DataLogger, Status Status	UINT8	ro	-	0…3	None
0x3182	0	N	dlgdiv	Moog DCV	DataLogger, Divider Divider	UINT16	rw	Y	1…65535	1
0x3183	0	N	dlgsmp	Moog DCV	DataLogger, Number of samples NumberOfSamples	INT32	ro	-	0…2048	None
0x3184	1…4	N	dlgena[0…3]	Moog DCV	DataLogger, Enable channel EnableChannel1…4	UINT8	rw	Y	UINT8	…
0x3185	1…4	N	dlgpar[0…3]	Moog DCV	DataLogger, Channel parameter ChannelParameter1…4	UINT32	rw	Y	UINT32	…
0x3186	0	N	dlgmem	Moog DCV	DataLogger, Memory Memory	DOMAIN	ro	-	None	None
0x3187	0	N	dlgofs	Moog DCV	DataLogger, Sample start offset SampleStartOffset	UINT32	ro	-	UINT32	None

Table 70: Object dictionary (part 9 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x3188	0	N	trgtyp	Moog DCV	DataLogger, Trigger type TriggerType	UINT8	rw	Y	0…2	1
0x3189	0	N	trgpar	Moog DCV	DataLogger, Trigger parameter TriggerParameter	UINT32	rw	Y	UINT32	0x63100110
0x318A	0	N	trgcpl	Moog DCV	DataLogger, Trigger coupling TriggerCoupling	UINT8	rw	Y	0…2	1
0x318B	0	N	trgslp	Moog DCV	DataLogger, Trigger slope TriggerSlope	UINT8	rw	Y	1…3	1
0x318C	0	N	trglvl	Moog DCV	DataLogger, Trigger level or bitmask TriggerLevelOrBitmask	INT32	rw	Y	INT32	0
0x318D	0	N	trgpos	Moog DCV	DataLogger, Trigger position TriggerPosition	INT32	rw	Y	INT32	0
0x318E	0	N	trgtim	Moog DCV	DataLogger, TriggerTimeStamp TriggerTimeStamp	UINT32	ro	-	UINT32	
0x3200	0	N	an0typ	Moog DCV	AnalogueInput0, Input type InputType	INT8	rw	Y	INT8	1
0x3204	0	Y	an0val	Moog DCV	AnalogueInput0, Actual value ActualValue0	INT16	ro	-	INT16	None
0x3208	0	N	an1typ	Moog DCV	AnalogueInput1, Input type InputType	INT8	rw	Y	INT8	2
0x320C	0	Y	an1val	Moog DCV	AnalogueInput1, Actual value ActualValue1	INT16	ro	-	INT16	None
0x3210	0	N	an2typ	Moog DCV	AnalogueInput2, Input type InputType	INT8	rw	Y	INT8	2
0x3214	0	Y	an2val	Moog DCV	AnalogueInput2, Actual value ActualValue2	INT16	ro	-	INT16	None
0x3217	0	N	an2mon	Moog DCV	AnalogueInput2, Monitoring current MonitoringCurrent	UINT8	rw	Y	0…1	0
0x3218	0	N	an3typ	Moog DCV	AnalogueInput3, Input type InputType	INT8	rw	Y	INT8	2
0x321C	0	Y	an3val	Moog DCV	AnalogueInput3, Actual value ActualValue3	INT16	ro	-	INT16	None
0x3220	0	N	an4typ	Moog DCV	AnalogueInput4, Input type InputType	INT8	rw	Y	INT8	2
0x3224	0	Y	an4val	Moog DCV	AnalogueInput4, Actual value ActualValue4	INT16	ro	-	INT16	None
0x3227	0	N	an4mon	Moog DCV	AnalogueInput4, Monitoring current MonitoringCurrent	UINT8	rw	Y	0…1	0

Table 70: Object dictionary (part 10 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x3228	0	N	an3mon	Moog DCV	AnalogueInput3, Monitoring current MonitoringCurrent	UINT8	rw	Y	0…1	0
0x3235	0	Y	extlvdval	Moog DCV	ExternalLVDT, Actual value ActualMainStageSplPositionValue	INT16	ro	-	INT16	None
0x3237	1…3	N	extlvdref[0…2]	Moog DCV	External LVDT, Customer scaling CustomerScalingFactorNumerator, CustomerScalingFactorDenominator, CustomerScaliffset	INT16	rw	Y	0…1	16384, 16384, 0
0x3240	0	N	da0par	Moog DCV	AnalogueOutput0, Parameter Parameter	UINT32	rw	Y	UINT32	0x63010110
0x3243	0	N	da0typ	Moog DCV	AnalogueOutput0, Type Type	UINT8	rw	N	UINT8	0
0x3244	1	N	da0ref[0]	Moog DCV	AnalogueOutput0, Scaling Numerator	INT16	rw	Y	INT16	16384
0x3244	2	N	da0ref[1]	Moog DCV	AnalogueOutput0, Scaling Denominator	INT16	rw	Y	INT16	16384
0x3244	3	N	da0ref[2]	Moog DCV	AnalogueOutput0, Scaling Offset	INT16	rw	Y	INT16	0
0x3245	0	Y	da0val	Moog DCV	AnalogueOutput0, ActualOutputValue0	INT16	ro	-	INT16	None
0x3250	0	N	anamonlow	Moog DCV	AnalogueInput, Lower current border LowerCurrentBorder	FLOAT32	rw	Y	2.2…20.0	3.0
0x3251	0	N	anamontim	Moog DCV	AnalogueInput, Analog input monitoring time anInputMonitoringTime_in_ms	UINT16	rw	Y	0…60000	10
0x3260	0	N	da1par	Moog DCV	AnalogueOutput1, Parameter	UINT32	rw	Y	UINT32	0x63810110
0x3263	0	N	da1typ	Moog DCV	AnalogueOutput1, Type Type	UINT8	rw	N	UINT8	0
0x3264	0	N	vlvtrdpar	Moog DCV	Valve_ActualValueConditioning, Transducer port TransducerPort	UINT32	rw	N	UINT32	None
0x3265	1	N	da1ref[0]	Moog DCV	AnalogueOutput1, Scaling Numerator	INT16	rw	Y	INT16	16384
0x3265	2	N	da1ref[1]	Moog DCV	AnalogueOutput1, Scaling Denominator	INT16	rw	Y	INT16	16384
0x3265	3	N	da1ref[2]	Moog DCV	AnalogueOutput1, Scaling Offset	INT16	rw	Y	INT16	0
0x3266	0	Y	da1val	Moog DCV	AnalogueOutput1, ActualOutputValue1	INT16	ro	-	INT16	None

Table 70: Object dictionary (part 11 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x3270	0	N	vlvtrdstc	Moog DCV	Valve_ActualValueConditioning, Valve transducer structure ValveTransducerStructure	DOMAIN	rw	Y	None	
0x3300	0	Y	spldemplt	Moog DCV	ValvePositionControl, Demand valve pilot DemandValvePilot	INT16	ro	-	INT16	None
0x3301	0	Y	splvalplt	Moog DCV	ValvePositionControl, Actual valve pilot ActualValvePilot	INT16	ro	-	INT16	None
0x3310	0	N	prspar	Moog DCV	ValvePressureControl, Setpoint parameter PrsSetpointParameter	UINT32	rw	Y	UINT32	0x63800110
0x3320	0	N	splpar	Moog DCV	ValvePositionControl, Setpoint parameter SplSetpointParameter	UINT32	rw	Y	UINT32	0x63000110
0x3404	0	Y	prstrd	Moog DCV	PressureTransducer, Actual value ActualValue	INT16	ro	-	INT16	None
0x3506	0	N	lvdcusofs	Moog DCV	Lvdt, CustomerScalingOffset CustomerScalingOffset	INT16	rw	Y	-819..819	0
0x4000	0	N	xmldat	Moog DCV	Device, XmlDescription XmlDescription	DOMAIN	ro	-	None	None
0x403F	0	N	ctllocdef	Moog DCV	Device, Local control word default LocalControlWordDefault	UINT16	rw	Y	UINT16	263
0x4040	0	Y	ctlloc	Moog DCV	Device, Local control word LocalControlWord	UINT16	rw	N	UINT16	<LocalCon-trolWord-De-fault> (0x403F)
0x4042	0	N	devmoddef	Moog DCV	Device, Device mode default DeviceModeDefault	INT8	rw	Y	1…2	1
0x4043	0	N	ctlmoddef	Moog DCV	Device, ControlModeDefault ControlModeDefault	INT8	rw	Y	-1…14	2
0x5B00	0	N	locmodide	Moog DCV	LocalCAN, Module identifier LocalCANModuleIdentifier	UINT8	rw	Y	1…127	127
0x5E41	1…2	N	digouttyp[0…1]	Moog DCV	ValveDigitalOutputType, Digital output type DigitalOutputType0...1	INT8	rw	Y	0…4	3, 2
0x5E42	1…2	Y	digout[0…1]	Moog DCV	ValveDigitalOutputValue, Digital output value DigitalOutputValue0...1	INT8	rw	N	0…1	None
0x6040	0	Y	ctlwrd	CiA 408	Device, Control word ControlWord	UINT16	rw	N	UINT16	None
0x6041	0	Y	stswrd	CiA 408	Device, Status word StatusWord	UINT16	ro	-	UINT16	None
0x6042	0	Y	devmod	CiA 408	Device, Device mode DeviceMode	INT8	rw	N	1…4	<DeviceMod-eDefault> (0x4042)

Table 70: Object dictionary (part 12 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6043	0	Y	ctlmod	CiA 408	Device, Control mode ControlMode	INT8	rw	N	-1…14	<ControlMod-eDefault> (0x4043)
0x604F	0	Y	locmod	CiA 408	Device, Local Local	INT8	rw	Y	-128…1	0
0x6050	0	N	devver	CiA 408	Device, Version DeviceVersion	STRING(64)	ro	-	None	
0x6051	0	Y	devcodnum	CiA 408	Device, Code number CodeNumber	UINT16	rw	Y	UINT16	0
0x6052	0	N	sernum	CiA 408	Device, Serial number SerialNumber	STRING(64)	ro	-	None	
0x6053	0	N	devdsc	CiA 408	Device, Description Description	STRING(64)	rw	Y	None	
0x6054	0	N	devmdldsc	CiA 408	Device, Model description ModelDescription	STRING(64)	ro	-	None	
0x6055	0	N	devmdlurl	CiA 408	Device, Model URL ModelURL	STRING(64)	ro	-	None	www.moog.com
0x6056	0	Y	devprmcod	CiA 408	Device, Parameter set code ParameterSetCode	UINT8	rw	Y	0…254	0
0x6057	0	N	devvennam	CiA 408	Device, Vendor name VendorName	STRING(64)	ro	-	None	MOOG
0x605F	0	Y	devcap	CiA 408	Device, Capability Capability	UINT32	ro	-	0x01000000 …0x3F009000	0x3F000000
0x6100	0	N	vlvtrdmax	CiA 408	Valve_ActualValueConditioning, Max interface number MaxInterfaceNumber	UINT8	ro	-	UINT8	4
0x6101	0	N	vlvtrditf	CiA 408	Valve_ActualValueConditioning, Interface number InterfaceNumber	UINT8	rw	N	1…4	1
0x6102	0	N	vlvtrdtyp	CiA 408	Valve_ActualValueConditioning, Type Type	INT8	rw	N	INT8	None
0x6103	0	N	vlvtrdsgn	CiA 408	Valve_ActualValueConditioning, Sign Sign	INT8	rw	N	-1…1	1
0x6104	1	Y	vlvtrdval	CiA 408	Valve_ActualValueConditioning, Actual value ActualValue	INT16	ro	-	INT16	None
0x6110	1	Y	trditfval[0]	CiA 408	Valve_ActualValueConditioning, Actual value 1 ActualValue1	INT16	ro	-	INT16	None
0x6111	1	Y	trditfval[1]	CiA 408	Valve_ActualValueConditioning, Actual value 2 ActualValue2	INT16	ro	-	INT16	None

Table 70: Object dictionary (part 13 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6112	1	Y	trditfval[2]	CiA 408	Valve_ActualValueConditioning, Actual value 3 ActualValue3	INT16	ro	-	INT16	None
0x6113	1	Y	trditfval[3]	CiA 408	Valve_ActualValueConditioning, Actual value 4 ActualValue4	INT16	ro	-	INT16	None
0x6120	1	N	trdprsmin	CiA 408	Valve_ActualValueConditioning, Minimum pressure MinimumPressure	INT16	rw	N	INT16	None
0x6121	1	N	trdprsmax	CiA 408	Valve_ActualValueConditioning, Maximum pressure MaximumPressure	INT16	rw	N	INT16	16384
0x6122	1	N	trdprsare	CiA 408	Valve_ActualValueConditioning, Area Area	INT16	rw	N	INT16	0
0x6123	1	N	trdprsofs	CiA 408	Valve_ActualValueConditioning, Pressure offset PressureOffset	INT16	rw	N	INT16	0
0x6124	1	N	trdprssigmin	CiA 408	Valve_ActualValueConditioning, Minimum transducer signal MinimumTransducerSignal	INT16	rw	N	INT16	0
0x6125	1	N	trdprssigmax	CiA 408	Valve_ActualValueConditioning, Maximum transducer signal MaximumTransducerSignal	INT16	rw	N	INT16	16384
0x6128	1	N	trdgenmin	CiA 408	Valve_ActualValueConditioning, GeneralInputMinimum GeneralInputMinimum	INT16	rw	N	INT16	0
0x6129	1	N	trdgenmax	CiA 408	Valve_ActualValueConditioning, GeneralInputMaximum GeneralInputMaximum	INT16	rw	N	INT16	16384
0x612B	1	N	trdgenofs	CiA 408	Valve_ActualValueConditioning, GeneralInputOffset GeneralInputOffset	INT16	rw	N	INT16	0
0x612C	1	N	trdgensigmin	CiA 408	Valve_ActualValueConditioning, GeneralInputSignalMinimum GeneralInputSignalMinimum	INT32	rw	N	INT32	0
0x612D	1	N	trdgensigmax	CiA 408	Valve_ActualValueConditioning, GeneralInputSignalMaximum GeneralInputSignalMaximum	INT32	rw	N	INT32	16384
0x6300	1	Y	splset	CiA 408	ValvePositionControl, Setpoint SplSetpoint	INT16	rw	N	INT16	0
0x6300	2	N	spluni	CiA 408	ValvePositionControl, Setpoint Unit	UINT8	ro	-	UINT8	0
0x6300	3	N	splprf	CiA 408	ValvePositionControl, Setpoint Prefix	INT8	ro	-	INT8	0
0x6301	1	Y	splval	CiA 408	ValvePositionControl, Actual value SplActualValue	INT16	ro	-	INT16	None
0x6301	2	N	spluni	CiA 408	ValvePositionControl, Actual value Unit	UINT8	ro	-	UINT8	0
0x6301	3	N	splprf	CiA 408	ValvePositionControl, Actual value Prefix	INT8	ro	-	INT8	0

Table 70: Object dictionary (part 14 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6310	1	Y	spldem	CiA 408	ValvePositionControl_DemandValueGenerator, Demand value SplDemandValue	INT16	ro	-	INT16	None
0x6310	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator, Demand value Unit	UINT8	ro	-	UINT8	0
0x6310	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator, Demand value Prefix	INT8	ro	-	INT8	0
0x6311	1	Y	spldemref	CiA 408	ValvePositionControl_DemandValueGenerator, Reference value SplReferenceValue	INT16	ro	-	INT16	16384
0x6311	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator, Reference value Unit	UINT8	ro	-	UINT8	0
0x6311	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator, Reference value Prefix	INT8	ro	-	INT8	0
0x6314	1	Y	splsethld	CiA 408	ValvePositionControl_DemandValueGenerator, Hold setpoint SplHoldSetPoint	INT16	rw	Y	INT16	0
0x6314	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator, Hold setpoint Unit	UINT8	ro	-	UINT8	0
0x6314	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator, Hold setpoint Prefix	INT8	ro	-	INT8	0
0x6320	1	Y	spllimupp	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, Upper limit UpperLimit	INT16	rw	Y	<Lower-Limit>(0x6321)…32767	16384
0x6320	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, Upper limit Unit	UINT8	ro	-	UINT8	0
0x6320	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, Upper limit Prefix	INT8	ro	-	INT8	0
0x6321	1	Y	spllimlow	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, Lower limit LowerLimit	INT16	rw	Y	-32768…<Upper-Limit>(0x6320)	-16384
0x6321	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, Lower limit Unit	UINT8	ro	-	UINT8	0
0x6321	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_Limit, Lower limit Prefix	INT8	ro	-	INT8	0
0x6322	0	Y	spldemfct	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Factor Factor	UINT32	rw	Y	UINT32	0x00010001
0x6323	1	Y	spldemofs	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Offset Offset	INT16	rw	Y	INT16	0
0x6323	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Offset Unit	UINT8	ro	-	UINT8	0
0x6323	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_Scaling, Offset Prefix	INT8	ro	-	INT8	0

Table 70: Object dictionary (part 15 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6324	1	Y	splzrocor	CiA 408	ValvePositionControl_DemandValueGenerator_ZeroCorrection, Offset Offset	INT16	rw	Y	INT16	0
0x6324	2	N	spluni	CiA 408	ValvePositionControl_DemandValueGenerator_ZeroCorrection, Offset Unit	UINT8	ro	-	UINT8	0
0x6324	3	N	splprf	CiA 408	ValvePositionControl_DemandValueGenerator_ZeroCorrection, Offset Prefix	INT8	ro	-	INT8	0
0x6330	0	Y	splrmptyp	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Type Type	INT8	rw	Y	3	0
0x6331	1	Y	splrmpacl	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration time AccelerationTime	UINT16	rw	Y	UINT16	0
0x6331	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration time Unit	UINT8	ro	-	UINT8	3
0x6331	3	Y	splrmpaclprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration time AccelerationTime_Prefix	INT8	rw	Y	-4…0	-3
0x6332	1	Y	splrmpaclpos	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration time positive AccelerationTimePositive	UINT16	rw	Y	UINT16	0
0x6332	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration time positive Unit	UINT8	ro	-	UINT8	3
0x6332	3	Y	splrmpaclposprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration time positive AccelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3
0x6333	1	Y	splrmpaclneg	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration time negative AccelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x6333	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration time negative Unit	UINT8	ro	-	UINT8	3
0x6333	3	Y	splrmpaclnegprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Acceleration time negative AccelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3
0x6334	1	Y	splrmpdcl	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration time DecelerationTime	UINT16	rw	Y	UINT16	0
0x6334	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration time Unit	UINT8	ro	-	UINT8	3
0x6334	3	Y	splrmpdclprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration time DecelerationTime_Prefix	INT8	rw	Y	-4…0	-3
0x6335	1	Y	splrmpdclpos	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration time positive DecelerationTimePositive	UINT16	rw	Y	UINT16	0
0x6335	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration time positive Unit	UINT8	ro	-	UINT8	3
0x6335	3	Y	splrmpdclposprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration time positive DecelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3

Table 70: Object dictionary (part 16 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6336	1	Y	splrmpdclneg	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration time negative DecelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x6336	2	N	timuni	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration time negative Unit	UINT8	ro	-	UINT8	3
0x6336	3	Y	splrmpdclnegprf	CiA 408	ValvePositionControl_DemandValueGenerator_Ramp, Deceleration time negative DecelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3
0x6350	1	Y	splctldvn	CiA 408	ValvePositionControl, Control deviation SplControlDeviation	INT16	ro	-	INT16	None
0x6350	2	N	spluni	CiA 408	ValvePositionControl, Control deviation Unit	UINT8	ro	-	UINT8	0
0x6350	3	N	splprf	CiA 408	ValvePositionControl, Control deviation Prefix	INT8	ro	-	INT8	0
0x6351	0	Y	splmontyp	CiA 408	ValvePositionControl_ControlMonitoring, Type Type	INT8	rw	Y	0…1	0
0x6352	1	Y	splmontim	CiA 408	ValvePositionControl_ControlMonitoring, Delay time DelayTime	UINT16	rw	Y	UINT16	30
0x6352	2	N	timuni	CiA 408	ValvePositionControl_ControlMonitoring, Delay time Unit	UINT8	ro	-	UINT8	3
0x6352	3	N	timprf	CiA 408	ValvePositionControl_ControlMonitoring, Delay time Prefix	INT8	ro	-	INT8	-3
0x6354	1	Y	splmonupp	CiA 408	ValvePositionControl_ControlMonitoring, Upper threshold UpperThreshold	INT16	rw	Y	INT16	512
0x6354	2	N	spluni	CiA 408	ValvePositionControl, Upper threshold Unit	UINT8	ro	-	UINT8	0
0x6354	3	N	splprf	CiA 408	ValvePositionControl, Upper threshold Prefix	INT8	ro	-	INT8	0
0x6355	1	Y	splmonlow	CiA 408	ValvePositionControl_ControlMonitoring, Lower threshold LowerThreshold	INT16	rw	Y	INT16	-512
0x6355	2	N	spluni	CiA 408	ValvePositionControl, Lower threshold Unit	UINT8	ro	-	UINT8	0
0x6355	3	N	splprf	CiA 408	ValvePositionControl, Lower threshold Prefix	INT8	ro	-	INT8	0
0x6380	1	Y	prsset	CiA 408	ValvePressureControl, Setpoint PrsSetpoint	INT16	rw	N	INT16	None
0x6380	2	N	prsuni	CiA 408	ValvePressureControl, Setpoint Unit	UINT8	ro	-	UINT8	0
0x6380	3	N	prsprf	CiA 408	ValvePressureControl, Setpoint Prefix	INT8	ro	-	INT8	0

Table 70: Object dictionary (part 17 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x6381	1	Y	prsval	CiA 408	ValvePressureControl, Actual Value PrsActualValue	INT16	ro	-	INT16	None
0x6381	2	N	prsuni	CiA 408	ValvePressureControl, Actual Value Unit	UINT8	ro	-	UINT8	0
0x6381	3	N	prsprf	CiA 408	ValvePressureControl, Actual Value Prefix	INT8	ro	-	INT8	0
0x6390	1	Y	prsdem	CiA 408	ValvePressureControl_DemandValueGenerator, Demand value PrsDemandValue	INT16	ro	-	INT16	None
0x6390	2	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator, Demand value Unit	UINT8	ro	-	UINT8	0
0x6390	3	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator, Demand value Prefix	INT8	ro	-	INT8	0
0x6391	1	Y	prsref	Moog DCV	ValvePressureControl_DemandValueGenerator, PrsReferenceValue PrsReferenceValue	INT16	rw	N	0...32767	400
0x6391	2	N	prsrefuni	Moog DCV	ValvePressureControl_DemandValueGenerator, PrsReferenceValue Unit	UINT8	ro	-	UINT8	0x4E
0x6391	3	N	prsrefprf	Moog DCV	ValvePressureControl_DemandValueGenerator, PrsReferenceValue Prefix	INT8	ro	-	INT8	0
0x6394	1	Y	prssethld	CiA 408	ValvePressureControl_DemandValueGenerator, Hold setpoint PrsHoldSetPoint	INT16	rw	Y	INT16	0
0x6394	2	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator, Hold setpoint Unit	UINT8	ro	-	UINT8	0
0x6394	3	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator, Hold setpoint Prefix	INT8	ro	-	INT8	0
0x63A0	1	Y	prslimupp	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, Upper limit UpperLimit	INT16	rw	Y	<LowerLimit> (0x63A1)…32767	16384
0x63A0	2	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, Upper limit Unit	UINT8	ro	-	UINT8	0
0x63A0	3	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, Upper limit Prefix	INT8	ro	-	INT8	0
0x63A1	1	Y	prslimlow	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, Lower limit LowerLimit	INT16	rw	Y	-32768…<Upper- Limit>(0x63A0)	-16384
0x63A1	2	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, Lower limit Unit	UINT8	ro	-	UINT8	0
0x63A1	3	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator_Limit, Lower limit Prefix	INT8	ro	-	INT8	0
0x63A2	0	Y	prsdemfct	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Factor Factor	UINT32	rw	Y	UINT32	0x00010001

Table 70: Object dictionary (part 18 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x63A3	1	Y	prsdemofs	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Offset Offset	INT16	rw	Y	INT16	0
0x63A3	2	N	prsuni	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Offset Unit	UINT8	ro	-	UINT8	0
0x63A3	3	N	prsprf	CiA 408	ValvePressureControl_DemandValueGenerator_Scaling, Offset Prefix	INT8	ro	-	INT8	0
0x63B0	0	Y	prsrmptyp	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Type Type	INT8	rw	Y	0…3	0
0x63B1	1	Y	prsrmpacl	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Acceleration time AccelerationTime	UINT16	rw	Y	UINT16	0
0x63B1	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Acceleration time Unit	UINT8	ro	-	UINT8	3
0x63B1	3	Y	prsrmpaclprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Acceleration time AccelerationTime_Prefix	INT8	rw	Y	-4…0	-3
0x63B2	1	Y	prsrmpaclpos	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Acceleration time positive AccelerationTimePositive	UINT16	rw	Y	UINT16	0
0x63B2	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Acceleration time positive Unit	UINT8	ro	-	UINT8	3
0x63B2	3	Y	prsrmpaclposprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Acceleration time positive AccelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3
0x63B3	1	Y	prsrmpaclneg	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Acceleration time negative AccelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x63B3	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Acceleration time negative Unit	UINT8	ro	-	UINT8	3
0x63B3	3	Y	prsrmpaclnegprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Acceleration time negative AccelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3
0x63B4	1	Y	prsrmpdcl	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Deceleration time DecelerationTime	UINT16	rw	Y	UINT16	0
0x63B4	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Deceleration time Unit	UINT8	ro	-	UINT8	3
0x63B4	3	Y	prsrmpdclprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Deceleration time DecelerationTime_Prefix	INT8	rw	Y	-4…0	-3
0x63B5	1	Y	prsrmpdclpos	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Deceleration time positive DecelerationTimePositive	UINT16	rw	Y	UINT16	0
0x63B5	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Deceleration time positive Unit	UINT8	ro	-	UINT8	3
0x63B5	3	Y	prsrmpdclposprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Deceleration time positive DecelerationTimePositive_Prefix	INT8	rw	Y	-4…0	-3

Table 70: Object dictionary (part 19 of 20)

Index	Sub-index	PDO mapping	Short name	Specification	Block name, Object name Parameter name	Data type	Access	Persistence	Value range	Default
0x63B6	1	Y	prsrmpdclneg	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Deceleration time negative DecelerationTimeNegative	UINT16	rw	Y	UINT16	0
0x63B6	2	N	timuni	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Deceleration time negative Unit	UINT8	ro	-	UINT8	3
0x63B6	3	Y	prsrmpdclnegprf	CiA 408	ValvePressureControl_DemandValueGenerator_Ramp, Deceleration time negative DecelerationTimeNegative_Prefix	INT8	rw	Y	-4…0	-3
0x63D0	1	Y	prsctldvn	CiA 408	ValvePressureControl, Control deviation PrsControlDeviation	INT16	ro	-	INT16	None
0x63D0	2	N	prsuni	CiA 408	ValvePressureControl, Control deviation Unit	UINT8	ro	-	UINT8	0
0x63D0	3	N	prsprf	CiA 408	ValvePressureControl, Control deviation Prefix	INT8	ro	-	INT8	0
0x63D1	0	Y	prsmontyp	CiA 408	ValvePressureControl_ControlMonitoring, Type Type	INT8	rw	Y	0…1	0
0x63D2	1	Y	prsmontim	CiA 408	ValvePressureControl_ControlMonitoring, Delay time DelayTime	UINT16	rw	Y	UINT16	30
0x63D2	2	N	timuni	CiA 408	ValvePressureControl_ControlMonitoring, Delay time Unit	UINT8	ro	-	UINT8	3
0x63D2	3	N	timprf	CiA 408	ValvePressureControl_ControlMonitoring, Delay time Prefix	INT8	ro	-	INT8	-3
0x63D4	1	Y	prsmonupp	CiA 408	ValvePressureControl_ControlMonitoring, Upper threshold UpperThreshold	INT16	rw	Y	INT16	512
0x63D4	2	N	prsuni	CiA 408	ValvePressureControl_ControlMonitoring, Upper threshold Unit	UINT8	ro	-	UINT8	0
0x63D4	3	N	prsprf	CiA 408	ValvePressureControl_ControlMonitoring, Upper threshold Prefix	INT8	ro	-	INT8	0
0x63D5	1	Y	prsmonlow	CiA 408	ValvePressureControl_ControlMonitoring, Lower threshold LowerThreshold	INT16	rw	Y	INT16	-512
0x63D5	2	N	prsuni	CiA 408	ValvePressureControl_ControlMonitoring, Lower threshold Unit	UINT8	ro	-	UINT8	0
0x63D5	3	N	prsprf	CiA 408	ValvePressureControl_ControlMonitoring, LowerThreshold Prefix	INT8	ro	-	INT8	0

Table 70: Object dictionary (part 20 of 20)

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.

Argentina
+54 11 4326 5916
info.argentina@moog.com

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 6605
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
+44 168 429 6600
info.uk@moog.com

USA
+1 716 652 2000
info.usa@moog.com

www.moog.com/industrial

Moog Literature Name
- Firmware
Rev. -, October 2013, Id. CDS39670-en