USER MANUAL

FIRMWARE

DIGITAL INTERFACE VALVE WITH ETHERCAT INTERFACE

(B99226-DV010-B-400; Version 0.3, 11/07)

WHAT MOVES YOUR WORLD

MOOG

Copyright
© 2007 Moog GmbH
Hanns-Klemm-Straße 28
71034 Böblingen
Deutschland

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

Subject to change without notice.
We reserve the right to make changes to this document at any time and without specified reasons.

Completeness
This document is complete only when used in conjunction with the product-related hardware and software documentation required for the relevant application, as for example operating instructions of the valve and other manuals.

Selection and qualification of personnel
As specified in the product-related hardware and software documentation required for the relevant application, only users properly qualified and authorized for these tasks may work with and on our products.

Note
This document has been prepared with great care in compliance with the relevant regulations, state-of-the-art technology and our many years of knowledge and experience, and the contents have been generated to the best of the authors' knowledge. However, the possibility of error remains and improvements are possible.
Please feel free to submit any comments about possible errors and incomplete information to us.

Table of Contents

1 General Description 5
1.1 About this manual 5
1.2 Scope of delivery 5
1.3 EtherCAT Introduction 5
1.4 References / Normen 6
1.5 Abbreviations 8

2 EtherCAT Fieldbus Interface 9
2.1 Technical Data 9
2.2 Bus topology 10
2.3 EtherCAT Protocol 10
2.3.1 Process Data Objects (PDO) / Process Data Channel 11
2.3.2 Service Data Objects (SDO) / Parameter Channel 12
2.3.3 Emergency Message (EMCY) 13
2.3.4 Network State Machine 14
2.3.5 Distributed Clock 16
2.4 Additional configurations 16
2.4.1 PDO Watchdog 16
2.4.2 Device Addressing 17
2.5 Store/Restore 18
2.5.1 Store 18
2.5.2 Restore 18
2.6 Status LEDs 19
2.6.1 Module Status LED (MS) 19
2.6.2 Network Status LED (NS IN / NS OUT) 19
2.7 CANopen Object Dictionary 20
2.8 XML - Slave Configuration File and Slave Information Interface (SII) 20

3 Appendix 21
3.1 EtherCAT XML Slave Device Description File 21
3.1.1 PdoType 21
3.1.2 EntryType 22

1 General Description

1.1 About this manual

This manual describes the software interface of the EtherCAT real time Ethernet field bus extension for the DIV valve. With the factory configuration the valve can be operated without any further parameter configuration in the valve. Application specific configuration can be done using Moog's Valve Configuration Service Tool MoVaCo.

The electrical installation is described in the manual "CA63420-002_BI_Elektrische_Anschlüsse.pdf" How to configure the network master to establish a field bus communication with the valve is described in the manual "Getting Started - Fieldbus Interface for Moog Servo Valves".

This Reference Manual only describes the EtherCAT specific configuration. All general parameter are described in the manual "USER MANUAL FIRMWARE, SERVO VALVES pQ WITH CAN BUS INTERFACE" B99224-DV010-BE400.

1.2 Scope of delivery

- Ventil mit EtherCAT Feldbusschnittstelle
- XML configuration file for the configuration of the network master e.g. z.B. D672-5704-0001.xml (Die EtherCAT- .xml ist nicht zu verwechseln mit der MoVaCo-xml-Konfigurationsdatei.)
- EDS configuration file for the SDO configuration EDS SDO Konfigurationsdatei z.B. D672-5704-0001.eds

1.3 EtherCAT Introduction

The EtherCAT fieldbus interface provides a 100Base-TX full duplex Real Time Ethernet connection to the digital valves using standard EtherCAT frames according to IEEE 802.3. The valve is always connected as a slave node within an EtherCAT fieldbus network.

VDMA-PROFIL

In einer Arbeitsgruppe innerhalb des Verbandes Deutscher Maschinen- und Anlagenbau e.V. (VDMA) wurde in enger Zusammenarbeit aller namhaften Hydraulikhersteller ein Geräteprofil „Profile Fluid Power technology" erarbeitet, das die Kommunikation von Hydraulikkomponenten über Feldbus beschreibt und einheitliche Funktionen und Parameter definiert, um für die Kommunikation – Hersteller übergreifend – ein standardisiertes Austauschformat zu schaffen. Dieses Profil ist in allen Moog Ventilen mit Feldbusschnittstelle implementiert.

Umsetzung des VDMA Geräteprofils auf den EtherCAT Feldbus

In der CiA-Organisation wurde das busunabhängige VDMA-Geräteprofil auf den CANopen standard abgebildet woraus das CAN-Busspezifische Geräteprofil DS408 „Device Profile Fluid Power Technology, Proportional Valves and Hydrostatic Transmissions" entstanden ist.

EtherCAT hat in seinen Protokolldefinitionen vieles von CANopen übernommen. So ist das gesamte Protokoll zur Konfiguration und Analyse des Ventils (SDO und emergency) identisch zu CiA DS-408. Damit benötigt EtherCAT kein eigenes Geräteprofil für Servo Ventile.

1.4 References / Normen

Moog

[1] „USER MANUAL FIRMWARE, SERVO VALVES pQ WITH CAN BUS INTERFACE”
B99224-DV010-BE400, Moog Industrial

[2] Getting Started - Fieldbus Interface for Moog Servo Valves, Moog Industrial

[3] “EtherCAT XML Slave Device Description file“, EtherCAT Organisation

EtherCAT

IEC61158-2_12	TYPE 12 EtherCAT, PHYSICAL LAYER SPECIFICATION AND SERVICE DEFINITION
IEC61158-3_12	TYPE 12, DATA-LINK SERVICE DEFINITION
IEC61158-4_12	TYPE 12, DATA-LINK PROTOCOL SPECIFICATION
IEC61158-5_12	TYPE 12, APPLICATION LAYER SERVICE DEFINITION
IEC61158-6_12	TYPE 12, APPLICATION LAYER PROTOCOL SPECIFICATION
EtherCATDevice Description_V0i3i0.pdf	Description of the XML-Schema of the “EtherCAT XML Slave Device Description File”
ISO/IEC 8802.3	EtherCAT Frame
ISO/IEC 8802-3	100Base-TX (IEEE 802.3 Clause 24)
IEC 61076-2-101, Amendment 1	M12 Connector
IEC 61158	Internationale Feldbusnormenreihe (Digital data communications for measurement and control - Fieldbus for use in industrial control systems)(Datalink layer service definition, Data Link Layer protocol specification, Application Layer service definition, Application Layer protocol definition)
IEC 61784	Digital data communication for measurement and control
IEC 61800-7	Generic interface and use of profiles for power drive systems – Interface definition

IEC Commission Electrotechnique Internazionale, International Electrontechnical Commission

EtherCAT Technology Group

ETG Headquarters
Ostendstraße 196
90482 Nürnberg
Germany

www.ethercat.org

Profile Fluid Power Technology

Proportional Valves and Hydrostatic Transmissions, Version 1.5

VDMA Zentrale
Lyoner Straße 18
60528 Frankfurt
Germany
www.vdma.org

CANopen

CiA DS-301	CANopen Specification of the Application Layer and Communication Profile
CiA DS-306	Electronic Data Sheet Specification
CiA DS-408	Device Profile Fluid Power Technology, Proportional Valves and Hydrostatic Transmissions
EN 50325-4:2002	This is the international norm of the CiA DS-301 specification based on the norm „ISO 11898 (CAN) - part 4: CANopen".
DIN ISO 11898-1:2003	specifies the data link layer (DLL) and physical signaling of the controller area network (CAN).

1.5 Abbreviations

µs	Micro seconds
AL	Application layer, see also Network Status Machine
CAN	Controller Area Network, specified by CAN in Automation (CiA)
CiA	CAN in Automation, international user’s and manufacturer’s organisation
CANopen	EN 50325-4, Osi Layer 7 protocol, specified by CAN in Automation (CiA)
CAT 5	category for twisted pair wires standardised in ANSI/TIA/EIA-568-B.1-2001
DC	Distributed Clock
DIV	Digital Interface Valve
Eds / EDS	Electronic Datasheets, containing a description of the CANopen object dictionary
ESC	EtherCAT Slave Controller
EtherCAT	Ethernet Control Automation Technologie
IEC	International Electrontechnical Commission
IEEE	Institute of Electrical and Electronics Engineers Inc.
IN	Input
IP	Internet Protocol
ISO	International Organisation for Standardization
m	Meter
MBit	Mega Bit
Mbit/s	Million bits per second
MoVaCo	Moog Valve Configuration Software
NMT	Network Management. It performs initialization, configuration and error handling in a CAN network.
OUT	Ouput
PDI	Physical Device Interface (in the slave device)
PDO	Process Data Object. Object for data exchange between several devices.
Phy	PHY Physical layer entity sub layer (→ISO/IEC 8802.3)
PLC	programmable logic controller
s	Second
SDO	Service Data Object. Peer to peer communication with access to the Object Dictionary of a device.
SII	Slave Information Interface
TCP	Transmission Control Protocol (one of the core protocols of internet)
UDP	User Datagram Protocol (one of the core protocols of internet)
UDP	User Datagram Protocol (→RFC 768)
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V.
XML	Extensible Markup Language, is a general-purpose language, combining text and extra information about the text.
100Base-TX	Twisted Pair Ethernet wiring with 100MBit/sec (Fast Ethernet)

2 EtherCAT Fieldbus Interface

2.1 Technical Data

100MBit/s transfer rate

100 Base Tx Physics (4 wires twisted pair) according CAT 5

Maximal cable length between two slave devices: 100m

Galvanically isolated

Termination: device internally

Network topology: tree and line

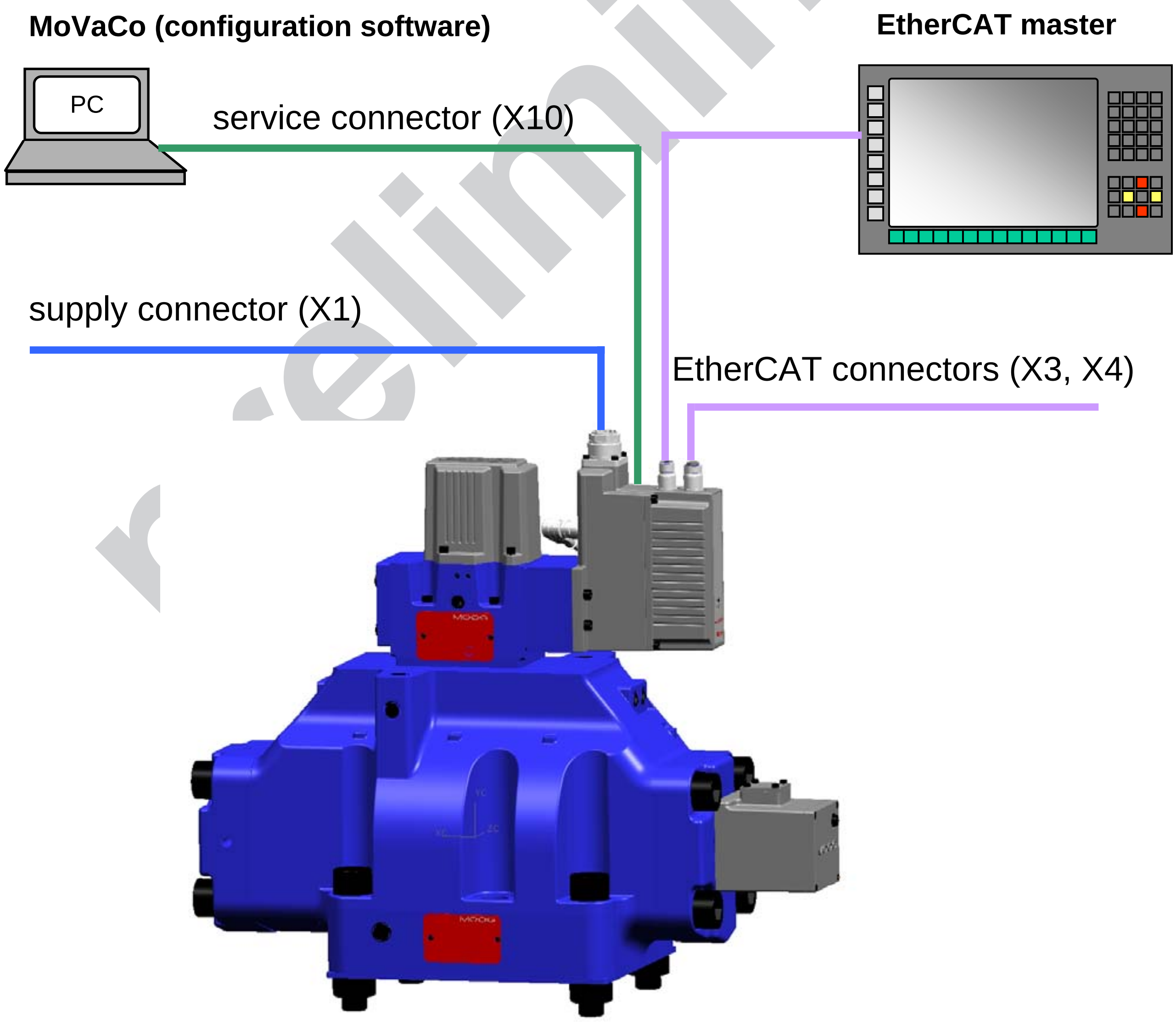
Update time: 100 distributed servo valves in 100 µs (set- and actual value 16bit)

Pass through delay: 4 µs per slave device

Maximal number of slave devices in a fieldbus line: 2^{16}, The real limit is the „Pass through delay" of the devices.

Ethernet standard ISO/IEC 8802.3 frame format, EtherType "0x88A4" according IEEE

Supported Protocols: PDO, CANopen SDO Mailbox-Interface "CoE"



2.2 Bus topology

The network physical topology is tree and line.

The connection between two devices is a point to point connection. Thus each slave device has an input and an output connector which relays the data to the next slave device in the line.

To build up a tree structure a switch has to be used.

To address a slave device, EtherCAT uses two different addressing modes.

- Auto Increment Addressing
- Using the Station Alias that means using a fixed device node address

For more details about device node addressing, see chapter “Device Addressing”.

2.3 EtherCAT Protocol

Das Protokoll wird in erster Linie als Master - Slave Kommunikation verwendet. Abhängig vom Master ist auch eine direkte Slave to Slave Kommunikation möglich. Näheres dazu in den Beschreibungen der Master und der Protokollbeschreibung der EtherCAT Organisation.

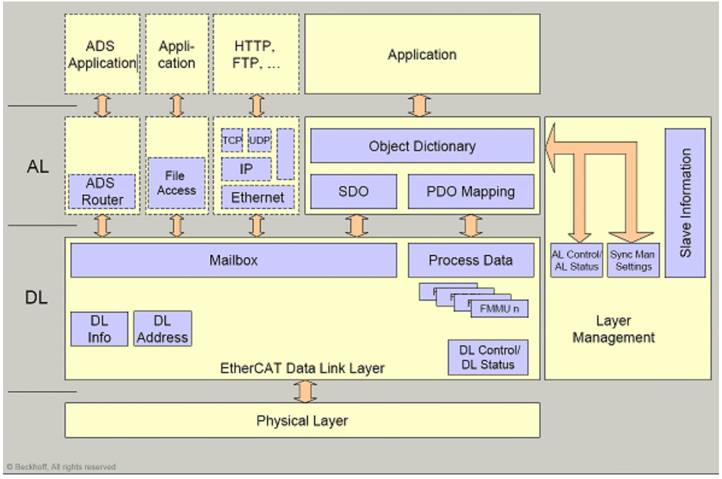
Die Netzwerkknoten stellen so genannte Datenobjekte zur Verfügung, die auch zum knotenübergreifenden Datenaustausch verwendet werden. Diese Datenobjekte werden in einem Objektverzeichis zur Verfügung gestellt und haben eine eindeutige Kennung, die so genannte Objekt-ID gemäß CANopen.

EtherCAT ermöglicht sowohl eine synchrone als auch eine asynchrone Übertragung der Daten.

- Die so genannten Prozessdaten-Objekte (PDO) werden synchron/zyklisch übertragen und können vom Anwender frei konfiguriert werden. (siehe Kapitel Process Data Objects (PDO) / Process Data Channel)
- Als Servicedaten-Objekte (SDO) bezeichnet man einen festen Satz an Parametrierdaten. Diese werden bei Bedarf asynchron (nicht Echtzeit) übertragen. (siehe Kapitel Service Data Objects (SDO) / Parameter Channel)

EtherCAT nutzt Standard Ethernet Datenpakete. Bei Verzicht auf die Echtzeitfähigkeit können die Daten somit auch über das gewöhnliche UDP, und damit über Internet, übertragen werden.

Die Synchronisation und Aufbereitung der Datenpakete wird im Slave Device von der Field Management Unit (FMMU) übernommen, der die Position der Datenobjekte innerhalb des EtherCAT-Datenpaketes festlegt. Der Sync Managern (SM) koordiniert den Zugriff auf diese Daten in dem Slave Device.



Supported Protocols

- Cyclic Process Data Objects (PDO)
- Acyclic Service Data Objects (SDO)
- Emergency Messages
- Network Management
- Distributed Clock (currently not supported)
- PDO Watchdog
- Device Node addressing
- EtherCAT frame packed into a UDP data package

These protocols will be described in detail in the following sub-chapters.

2.3.1 Process Data Objects (PDO) / Process Data Channel

The Process Data Objects (PDO) will be used for cyclic transmission of the process data.

One Receive- and one Transmit-PDO can be configured each containing 8 free configurable 8-32 bit parameter of the valve.

This configuration is done using the following Service Data Objects (SDOs).

Receive-PDO:

0x1600#1..8, pdrmap List of Receive Process data / Set Values
0x1600#0, pdrmapnum Number of configured receive process data

Transmit-PDO:

0x1a00#1..8, pdtmap List of Transmit Process Parameter / Actual Values

0x1a00#0, pdtmapnum Number of configred transmit process data

The Process Data Objects (PDOs) can be mapped by the user to a wide range of the valves process data.

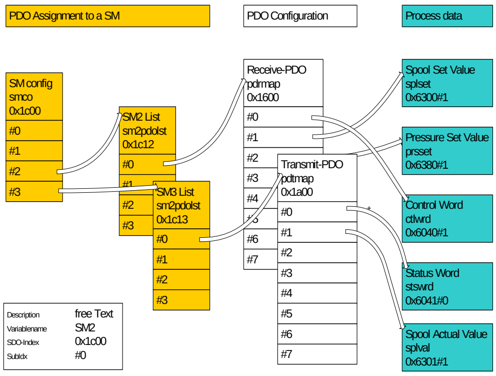
The default PDO mapping is as follows:

Default Parameter Mapping for the Receive PDO

Parameter Index (CANopen)	**Parameter Sub-Index**	**Parameter Description**
0x6040	0	Control Word
0x6300	1	Spool Setvalue
0x6380	1	Pressure Setvalue

Default Parameter Mapping for the Transmit PDO

Parameter Index (CANopen)	**Parameter Sub-Index**	**Parameter Description**
0x6041	0	Status Word
0x6301	1	Spool Actual Position
0x6381	1	Pressure Actual Value



The configuration in the valve (CANopen SDO Index 0x1600 and 0x1a00) should be consistent with the configuration in the master which will be defined in the Slave Description File (XML-Format, e.g. D636E1001-0001.xml). The format of the Slave Description File (also called “EtherCAT XML Device Description File”) is described by an XML-Schema EtherCATConfig.xsd defined by the EtherCAT organisation. For more details about the syntax of the XML-Slave Configuration File see chapter “EtherCAT XML Slave Device Description file” in the appendix.

It is possible to configure the PDO in the master only and to download the configuration into the valve during the network startup (during network status transition from Prepared to Save-OP).

The cycle time of both PDOs have to be configured in the network master.

Jedes PDO ist über die SDO Index 0x1c12, 0x1c13 einem Sync Manager (SM) zugeordnet. Diese Konfiguration sollte nicht verändert werden.

The PDO mapping is defined in the CANopen DS408 standard. The additional mapping with the objects 0x1cxx is defined in IEC61158-6-12_20060210.

2.3.2 Service Data Objects (SDO) / Parameter Channel

The EtherCAT interface does provide the CANopen configuration protocol (SDO) which is standardised in the DS408 profile. This profile describes the functionality of interconnectable proportional valves and other hydraulic components. It defines a vendor independent common set of configuration parameters as well as the communication protocol itself.

This SDO protocol is used for

- configuration, calibration and control parameter
- diagnostic functions, error states, internal process values, data logger, function generator,
- device identification, device capability, device description

SDO communication is asynchrony and always initiated by the network master. The slave responds always on request of the master. If the valve is not capable of processing the SDO request from the master, the valve responds with an SDO-Abort error message.

The SDO protocol is tunnelled in the EtherCAT-Mailbox protocol which can be accessed directly via the EtherCAT protocol or encapsulated in a standard internet UDP frame.

For more information about the available configuration parameter of the valve see [1].

To store the parameter inside the non volatile memory area a write command to the SDO “Store Parameter” parameter (Index 0x1010) has to be performed (see [1])

Most configurations can be done while the application is running. Anyway, configuration during operation has some risk and is not recommended for critical applications.

2.3.3 Emergency Message (EMCY)

The valve can send Emergency Message via CANopen over EtherCAT (CoE) to the master. These messages can be used by the master program for fault reactions. The Emergency Message is send if an internal error occurs. Each alert is only sent once.

After detection of a fault, the device sends out an emergency message as described in chapter "Emergency Message" in [1]. In the valves Status Word an error bit is set. Depending on the set fault reaction, the device changes into a fault state. In order to get out of the fault state the fault must be acknowledged. This is achieved by sending the Control Word (0x6040) to the device with the fault reset bit set (Bit 3 of the Control Word). The error will not be send again until it has been acknowledged.

The Emergency Message and the Message Codes are standardised in the VDMA Device Profile and IEC61158-6-12_20060210 Chapter 5.6.4.2. or [1], chapter “Emergency Message”.

For the EtherCAT fieldbus additional messages are added which are defined in the IEC61158-6-12_20060210, Chapter 5.6.4.3 :

- Transmit PDO Konfigurationsfehler FAUECTTPD 0x8245
  Der Fehler wird gemeldet/gelöscht beim der erfolgreichen/erfolglosen Transmit-PDO Initialisierung. Diese findet beim Übergang des AL-Status (Netzwerkstatus) von Save-Operational – nach Operational statt.
- Receive PDO Konfigurationsfehler FAUECTRPD 0x8241
  Der Fehler wird gemeldet/gelöscht beim der erfolgreichen/erfolglosen Receive-PDO Initialisierung. Diese findet beim Übergang des AL-Status (Netzwerkstatus) von
  Pre-Operational – nach Save-Operational statt.
- Feldbuskarten-Initialisierungs- bzw. Hardwarefehler FAUECTCOM 0x8100
  Die Feldbuskarten-Initialisierung findet beim Einschalten oder bei einem Neustart des Ventils (z.B. nach SW-Reset) statt.
- Receive PDO Watchdog timeout FAUECTRPT 0x8231
  Der Fehler wird ausgelöst wenn das Ventil im Netzwerkstatus OP ist und kein Receive PDO empfangen wird. Er wird gelöscht, sobald wieder PDOs empfangen werden.
  Der Netzwerkstatus selbst wird durch die Fehlermeldung nicht verändert.
  Zur Aktivierung dieses Watchdog Fehlermeldung muss sowohl das Stauts-Bit im Sync-Manager gesetzt sein, als auch eine entsprechende Fehlerreaktion aktiviert werden.

2.3.4 Network State Machine

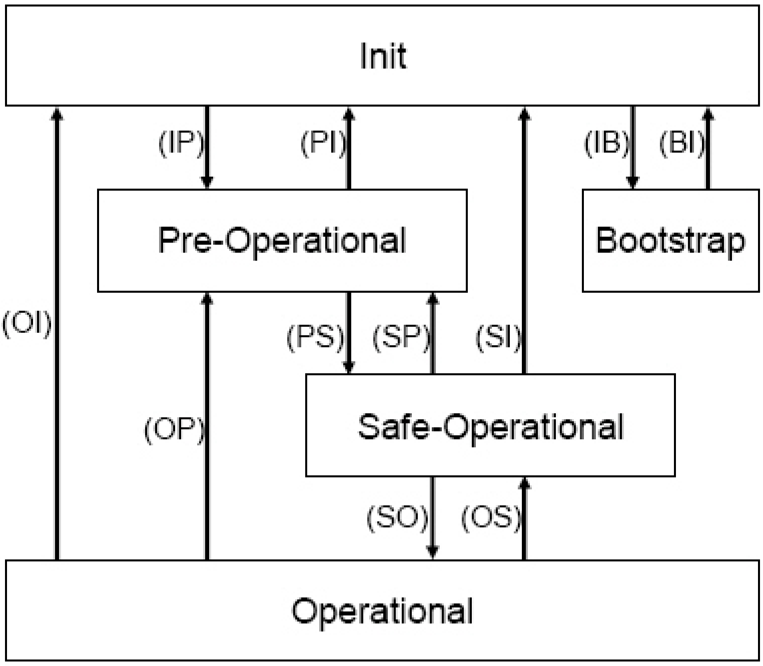
The EtherCAT State Machine is responsible for the coordination of master and slave during operation and device and network start-up.

The Network State Machine is also called Application Layer Status Machine.

In the following sentences AL stands for Application Layer.

The EtherCAT State Machine is represented in three register:

AL Status Ctrl	represents the requested state of the network state machine
AL Status Code	represents the actual state of the network state machine
AL Status Code Register	represents information about the error state of the device



2.3.4.1 State Machine States

State	Description
Init	Default state after initialisation of the device. No communication between the master and the slave on the application layer is possible.
Pre-Operational	In this state the mailbox is initialised and can be used e.g. for SDO communication. No PDO Communication is possible in this state.
Safe-Operational	In this state the PDOs input buffer is configured and actual “input” data are serviced by the valve and send to the master. The valves set point values (“output” data) are set to their “save state”.
Operational	In this state the actual outputs send from the master are used by the valves application.
Bootstrap	Not used.

2.3.4.2 State Machine State Transitions

The state machine transitions are requested by the network master.

State transition	Local management service
IP	Start Mailbox Communication
PI	Stop Mailbox Communication
PS	Start Input Update
SP	Stop Input Update
SO	Start Output Update
OS	Stop Output Update
OP	Stop Output Update, Stop Input Update
SI	Stop Input Update, Stop Mailbox Communication
OI	Stop Output Update, Stop Input Update, Stop Mailbox Communication
IB	Start Bootstrap Mode
BI	Restart Device

2.3.4.3 The AL Status Code Register

If AL Status indicates an error, the AL Status Code Register provides detailed information about the reason.

Code	Description	Current state or state change	Resulting state
0x0000	No error	Any	Current state
0x0001	Unspecified error	Any	Any + E
0x0011	Invalid requested state change	I→S,I→O, P→O, O→B, S→B, P→B	Current state + E
0x0012	Unknown requested state		Current state + E
0x0013	Bootstrap not supported	I→B	I + E
0x0014	No valid firmware	I→P	I + E
0x0015	Invalid mailbox configuration	I→B	I + E
0x0016	Invalid mailbox configuration	I→P	I + E
0x0017	Invalid sync manager configuration	P→S, S→O	Current state + E
0x0018	No valid inputs available	S, P→S	P + E
0x0019	No valid outputs	O, S→O	S + E
0x001A	Synchronisation error	O, S→O	S + E
0x001B	Sync manager watchdog	O,S	S + E
0x0020	Slave needs cold start	Any	Any + E
0x0021	Slave needs INIT	B,P,S,O	Any + E
0x0022	Slave needs PREOP	S,O	S + E, O + E
0x0023	Slave needs SAVEOP	O	O + E

Abbreviations: I=Init, P=Preop, O=Operational, S=Save-OP, B=Bootstrap, E=Error

AL Status Ctrl and AL Status Code Values (16 bit short integer variable):

Parameter	**Data Type,** **Bit Position**	**Value** Network State
State	Unsigned, 0..3	1: Init 2: Pre-Operational 3: Bootstrap 4: Save-Operational 8: Operational
Change (only AL Status Code)	Unsigned, 4	0: State transition successful 1: State transition unsuccessful
Reserved	Unsigned, 5..7	Shall be zero

2.3.5 Distributed Clock

The Distributed Clock Protocol provides a method to do a very precise synchronization of network devices to a global time base. This protocol is currently not supported in the valve.

2.4 Additional configurations

2.4.1 PDO Watchdog

For the Receive PDO a watchdog can be activated which produces a watchdog timeout error FAUECTRPT 0x8231. This error is thrown if the PDO is not received within a defined duration. For this error an appropriate fault reaction can be configured by the user (→chapter “Fault Reactions” in [1]).

To activate the watchdog the XML-Slave Configuration file has to be modified. Therefore the bit 0x40 of the Control Byte of the Sync Manager has to be set and the configuration has to be re-read from the master.

Line to be changed in the XML-Slave Configuration file:

```
<Sm StartAddress="#x1000" ControlByte="#x64" Enable="1">Outputs</Sm>
```

If the PDO Watchdog is enabled and the application program stops its PDO transmission, the watchdog with its corresponding error reaction is triggered. The watchdog time can be configured in the “PDI Watchdog” parameter setting of the Slave Configuration in the Network Master (Master | EtherCAT | Advanced Settings | General | Behavior | Set SM Watchdog (Reg. 410h).

For more details about the PDO Watchdog, see chapter “EtherCAT XML Slave Device Description file“.

2.4.2 Device Addressing

EtherCAT devices can be addressed using two different addressing modes.

- Position Addressing (typically used)
  also referred as Auto Increment Addressing
  Each slave device is addressed via its physical position within the network segment. Each slave device increments the 16-bit address field while the datagram passes through the slave devices. This mechanism has the advantage that no slave device addresses have to be set manually at the slaves.

- Device Node Addressing
  also referred as Fixed Node Addressing using a Station Alias Address for each device.
  With node addressing, the slaves are addressed via a configured node address called “Configured Station Alias”. This configured address can be set by the network master. This ensures that, even if the segment topology is changed or devices are added or moved, the slave devices can still be addressed via the same configured address.
  The node address can be configured by the master accessing the Slave Information Interface (SII, a set of configuration parameter in the configuration EPROM of the fieldbus interface) of the slave or by setting it using the CANopen SDO 0x3002.

One may ask how it could be possible to select a single device from a batch of other networking devices. Therefore, the valve has a unique identifier by which it can clearly be specified. Unique in this case means, that no other device in the world has the same identifier. This identifier can be obtained from our valve’s (main stage) nameplate. The displayed identifier (LSS address) has the following form:

LSS VendorID / ProductCode / RevisionNumber / SerialNumber

(All values are given as decimal numbers.)

This identifier is shown in the EtherCATs Slave Information Interface and the SDO index 0x1018.

For more details see

IEC61158-4-12, Chapter 6.1.2,

IEC61158-3-12, Chapter 7.1.1.1, 7.1.1.2

IEC61158-6-12, Chapter 5.4

2.5 Store/Restore

It is possible to save the changes of configuration inside a non volatile memory area of the electronic.

2.5.1 Store

To save the configuration changes a write access to the function “StoreParameters” has to be performed.

Slot	Index	Name	Data Type	Access	Persis-tence	Value Range	Default
0	51	Store Parameters	INT32	rw	N	INT32	0

Value table:

Value	Function
0x73617665 (‘save’)	“StoreParameters” function is called
...	all other values are rejected

Table 1: Store

2.5.2 Restore

To reset the configuration of the valve to factory default settings a write access to the function “RestoreDefaultParameters” has to be performed.

Slot	Index	Name	Data Type	Access	Persis-tence	Value Range	Default
0	52	Restore Default Parameters	INT32	rw	N	INT32	0

Value table:

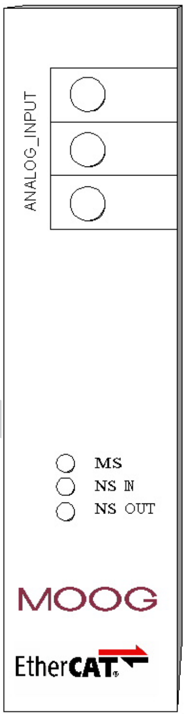
Value	Function
0x6C6F6164 (‘load’)	“RestoreDefaultParameters” function is called
...	all other values are rejected

Table 2: Restore

When the “RestoreDefaultParameters” function is started the configuration of the valve is reset to the factory default values. The new parameters are automatically stored in the internal nonvolatile memory of the valve. Because also the communication configuration of the valve is restored the valve performs a restart of the application program to reinitialize the communication interface. Therefore the “RestoreDefaultParameters” request is not acknowledged by the valve, because the communication configuration is changed immediately. The valve automatically restarts the PROFIBUS state machine and waits for the new parameterization and configuration from the master.

! A “RestoreDefaultParameters” may only be performed in a save machine state because all configuration changes like communication parameters or controller adjustments are set back to factory default.

2.6 Status LEDs



The valve's operating mode and the network status are displayed on multi colour light emitting diodes (status display LEDs) on the front of the electronics housing of the valve.

After the valve's power supply is switched on, the valve electronics perform a self-test, indicated by red and green blinking LEDs.

2.6.1 Module Status LED (MS)

The module status LED displays the state of the power supply and possible operational and error states.

Module status LED (MS)	**Condition**
off	no supply power
green	**normal operation**
blinking green	valve standby mode
blinking red	correctable error
red	unrecoverable error
blinking red-green	self-test

2.6.2 Network Status LED (NS IN / NS OUT)

The network status LED displays the status of the networks state machine and the links connected on the fieldbus Input (NS IN) and Output (NS OUT) connectors.

Network status LED **(NS IN and NS OUT)**	**Condition**	**EtherCAT** **Network Status**
Off	No supply power for the valve or one of the connected devices or master network card not started	Not defined
Blinking red-green	Self-test	-
Orange	Hardware link is OK (Link up) and ready to operate. No data communication.	Init
Blinking green	Hardware link is OK. SDO communication is possible. No PDO communication.	PreOP, SaveOP
Green	**Normal operation.** **SDO communication is possible.** Cyclic PDO communication established.	**Operational**
Red	Unrecoverable error	-

2.7 CANopen Object Dictionary

For all fieldbus independent objects see [1].

The only EtherCAT fiedbus specific objects are the emergency objects, see chapter “Emergency Message” of this document.

2.8 XML - Slave Configuration File and Slave Information Interface (SII)

Die Konfiguration der Mailbox, der FMMU, der SM und der PDOs ist im Slave Information File gespeichert. Damit kann ein Nertzwerk-Master auch ohne angeschlossenes Ventil konfiguriert werden.

Manche Netzwerk-Master unterstützen das Lesen des Slave Information Files nicht. Diese lesen die Konfigurationen aus dem Slave Information Interface (SII) Konfigurationsspeicher in Form eines EPROMs) des Ventils und dem PDO Mapping in den SDO Objekten 0x1600ff aus. Für diese Master muss die Konfiguration entsprechend in das SII gespeichert werden. Das ist im Auslieferungszustand des Ventils nicht der Fall. Wird dieses Interface genutzt, so ist auf konsistente Daten in der SII-Konfiguration und der PDO Konfiguration in den Objekten 0x1600 ff zu achten.

3 Appendix

3.1 EtherCAT XML Slave Device Description File

The format of the EtherCAT XML Device Description File is described by an XML-Schema EtherCATConfig.xsd defined by the EtherCAT organisation.

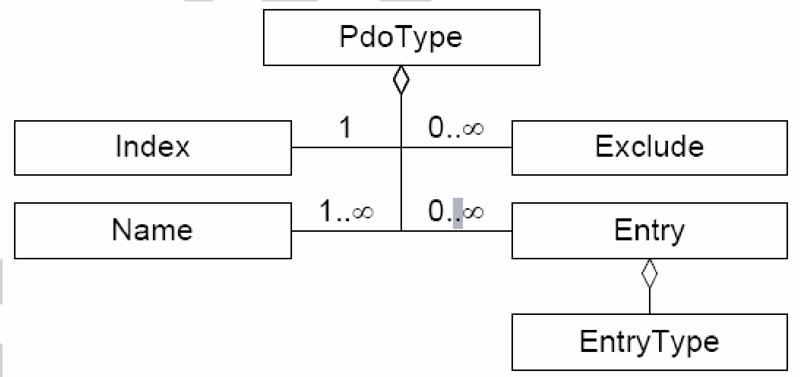
Example of one process data mapping in the XML-Slave Configuration File:

```
<RxPdo Fixed="1" Sm="2">
        <Index>#x1600</Index>
        <Name>Outputs</Name>
        <Entry>
                <Index>#x6040</Index>
                <SubIndex>0</SubIndex>
                <BitLen>16</BitLen>
                <Name>Control word (ctlwrd)</Name>
                <DataType>UINT</DataType>
        </Entry>
```

The following sub-chapters describe the tags relating to the PDO configuration within the EtherCAT's XML Slave Device Description File. For more details about the XML-Schema see [3].

3.1.1 PdoType

PdoType describes how an PDO has to be built. The following figure shows how the PdoType is composed.



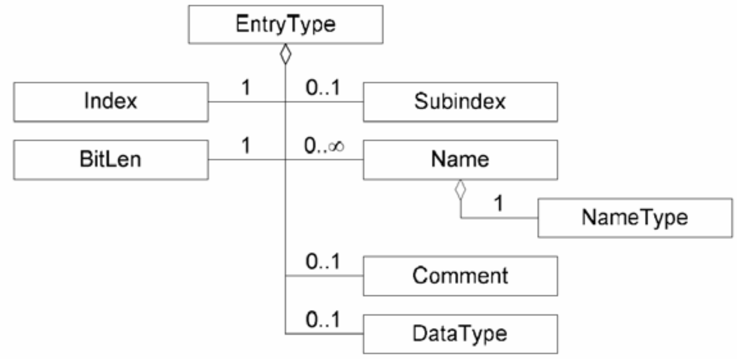
Attributes:

Attribute	Data Type	Use	Description
Fixed	xs:boolean	optional	PDO is not configurable
Mandatory	xs:boolean	optional	PDO must be configured in a SyncManager
Sm	xs:int	optional	Default SyncManager for this PDO (if set, this PDO will be included in the process Image

Element Name	Data Type	Description
Index	HexDecValue	Index of the PDO within the Object dictionary
Name	NameType	PDO name
Exclude	HexDecValue	List of PDO indices that are excluded if this PDO is in sync manager
Entry	EntryType	Description of all entries according to →EntryType

3.1.2 EntryType

EntryType describes an Entry of an Object within the Object Dictionary. The following figure shows how the EntryType element is composed.



Content of EntryType:

Element Name	Data Type	Description
EntryType:Index	HexDecValue	Index of this Entry within the PDO
EntryType:SubIndex	HexDecValue	Subindex of this Entry
EntryType:BitLen	xs:int	Bit Length of Entry
EntryType:Name	NameType	Name of this Entry
EntryType:Comment	xs:string	Description of this Entry
EntryType:DataType	xs:string	DataType of this Entry Only Base data types are allowed. See also Table 8. DataType is mandatory if Index != 0

MOOG.COM/INDUSTRIAL

For the location nearest you, contact
moog.com/industrial/globallocator

Argentina	+54	11 4326 5916	info.argentina@moog.com
Australia	+61	3 9561 6044	info.australia@moog.com
Austria	+43	664 144 6580	info.austria@moog.com
Brazil	+55	11 5523 8011	info.brazil@moog.com
China	+86	21 2893 1600	info.china@moog.com
Finland	+358	9 2517 2730	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 4120 8799	info.india@moog.com
Ireland	+353	21 451 9000	info.ireland@moog.com
Italy	+39	332 42111	info.italy@moog.com
Japan	+81	463 55 3615	info.japan@moog.com
Korea	+82	31 764 6711	info.korea@moog.com
Luxembourg	+352	40 46 401	info.luxembourg@moog.com
Netherlands	+31	252 462 000	info.netherlands@moog.com
Norway	+47	224 32927	info.norway@moog.com
Russia	+7	(8) 31713 1811	info.russia@moog.com
Singapore	+65	6773 6238	info.singapore@moog.com
South Africa	+27	12 653 6768	info.southafrica@moog.com
Spain	+34	902 133 240	info.spain@moog.com
Sweden	+46	31 680 060	info.sweden@moog.com
Switzerland	+41	71 394 5010	info.switzerland@moog.com
United Kingdom	+44	168 429 6600	info.unitedkingdom@moog.com
USA	+1	716 652 2000	info.usa@moog.com

**Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.
(B99226-DV010-B-400; Version 0.3, 11/07)**

WHAT MOVES YOUR WORLD

MOOG