atos
the smart electrohydraulics

USER MANUAL

S-MAN-S-EH

SMART SERVOPUMP SYSTEM
PROGRAMMING INSTRUCTIONS
EtherCAT PROTOCOL

INDEX

1 GENERAL

1.1 About this user manual

This manual describes the required information to operate Atos Smart ServopPump system (SSP) using EtherCAT fieldbus communication: always refer to the specific drive manual (see 1.2) for a complete description of the available function and of the parameters settings.

To speed up the fieldbus startup operations it is always recommended to use the Atos S-SW-SETUP programming software for PC before connecting Atos SSP to the fieldbus: S-SW-SETUP programming software allows a fast identification of the functions and parameters that would be included in the EtherCAT communication.

The purpose of this manual is not to cover all the details or variations of EtherCAT fieldbus, Atos drive and software, and it does not provide complete details for all possible working conditions; if any further information or technical support are required, please contact the Technical Sales Support of Atos Electronic Division (ele-support@atos.com).

In addition please follow up all the current regulations of the country/community where the drives will be used.

A basic skill in using personal computers and Windows® operating system is required.

For information about mechanical and electrical installation of a complete SSP system (drives, motors, pumps, fuses, inductances and wiring cable) please refer dedicated manual S-MAN-HW - see 1.2

1.2 Documentation

Additional information about electronic drives, motor, pump and Atos software can be found into the Atos web site or in the Atos Download Area.

Related documentations

- S-MAN-S-SW SSP programming software – user manual
- S-MAN-HW SSP system installation - user manual
- S-MAN-STO Safety Torque Off instruction – user manual
- AS050 Basics for Smart Servopumps - SSP - technical table
- AS100 Smart Servopumps - SSP- technical table
- AS200 Sizing criteria for Servopumps - technical table
- AS300 PGI - Cast iron internal gear pumps for SSP servopumps- technical table
- AS350 PGIL - Aluminium internal gear pumps for SSP servopumps - technical table
- AS400 PMM – Electric motors for SSP servopumps - technical table
- AS500 D-MP – Digital electronic drives for SSP servopumps - technical table
- AS800 Programming tools for pumps & servopumps – technical table
- AS810 Accessories for SSP servopumps - technical table
- AS910 Operating and maintenance information for SSP servopumps - technical table
- GS510 Fieldbus features

Other standards

- 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 DS 301 v4.02 CANopen – Application Layer and Communication Profile for Industrial Systems
- CiA DR 303-1 v1.7 Cabling and connector pin assignment
- CiA DSP 305 v2.2 CANopen – Layer Setting Services and Protocol

1.3 Trademarks

Windows® is a registered trademark of Microsoft Corporation
Pentium® is a registered trademark of Intel Corporation
EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany
Beckhoff® and TwinCAT®, are registered trademarks of and licensed by Beckhoff Automation GmbH
PROFIBUS™, PROFINET™ and PROFIsafe™, as well as the relevant logos, are registered trademarks of PROFIBUS Nutzerorganisation e.V. (PNO) SIMATIC is a registered trademark of SIEMENS AG

1.4 Abbreviations

Abbreviation	Description
AL	Application Layer
APRD	Auto Increment Read
APRW	Auto Increment Read Write
APWR	Auto Increment Write
ARMW	Auto Increment Read Multiple Write
FRMW	Configured Read Multiple Write
BRD	Broadcast Read
BRW	Broadcast Read Write
BWR	Broadcast Write
CAL	CAN Application Layer
CAN	Controller Area Network
CANopen	ISO/OSI Layer 7 protocol specified by CAN in Automation
CiA	CAN in Automation
CoE	CANopen over EtherCAT
DL	Data Link Layer
DR	Draft Recommendation
DRP	Draft Recommendation Proposal
DS	Draft Standard
DSP	Draft Standard Proposal
EEPROM	Electrically Eresable Programmable Read Only Memory
EMC	ElectroMagnetic Compatibility
EMCY	Emergency
EDS	Electronic Data Sheet
ESC	EtherCAT Slave Controller
ESD	ElectroStatic Discharge
ESI	EtherCAT Slave Information
ESM	EtherCAT State Machine
ETG	EtherCAT Technology Group
EtherCAT	Ethernet for Controller and Automation Technology
FCS	Frame Check Sequence
FIFO	First Input First Output
FMMU	Fieldbus Memory Management Unit
FoE	File access over EtherCAT
FPRD	Configured Address Read
FPRW	Auto Increment Read Write
FPWR	Auto Increment Write
IEC	International Electrotechnical Commision
ISO	International Standard Organization
IP	Internet Protocol
IRQ	Interrupt Request
ISO	Internation Standard Organization
LRD	Logical Read
LRW	Logical Read Write
LWR	Logical Write
LSB	Least Significat Byte
LSS	Layer Setting Services
LVL	Level
MSB	Most Significat Byte
NOP	No Operation
OSI	Open Systems Interconnection
PDO	Process Data Object
PWM	Pulse Width Modulation

Res	Reserved
RO	Read Only
RPDO	Receive Process Data Object
SDO	Service Data Object
SM	SyncManager
SYNC	Synchronization
TPDO	Transmit Service Data Object
UDP	User Datagram Protocol
USB	Universal Serial Bus
VLAN	Virtual LAN
WD	Watchdog
WKC	Working Counter
XML	Extensible Markup Language

2 ABOUT ETHERCAT - ETHERNET FIELDBUS

EtherCAT is a real-time industrial Ethernet communication fieldbus interface. In 2003 Beckhoff Automation GmbH has developed EtherCAT (Ethernet for Controllers and Automation Controllers).

EtherCAT Technology Group (ETG) is a research group of associated user and was established to further develop this technology.

EtherCAT was developed using a lot of protocol definitions of CANopen. In this way for drive configuration and analysis (SDO and EMCY) EtherCAT uses the same requirements DSP402.

A flexible topology (line, tree or star) and an easy configuration, allow EtherCAT to be managed as a traditional fieldbus.

EtherCAT protocol is directly transferred in the Ethernet frame according to IEEE 802.3 standard (Broadcast, Multicast and Slave−to−Slave communication types are supported).

For data exchange via EtherCAT the slave uses a hardware system dedicated to process the Ethernet telegram according to the EtherCAT protocol.

The master instead uses a software solution therefore only an Ethernet standard connection is required.

As for other Industrial fieldbus systems, the use of EtherCAT interface on the drives introduces the following advantages:

- **Lower installation costs**: standard 4-wires connection allows drastic cost reduction in comparison to the conventional ”one to one” wiring of standard analog components.
- **Improved Safety**: an elevated immunity to the electromagnetic interferences is performed due to the small number of the electric connections and the galvanic insulation between the fieldbus and the power devices.
- **Improved expandability**: the adding of new components in fieldbus network requires only wiring to the bus and software configuring - no change on the control panel and no addition of cables on the machine are required.
- **Standardization**: all the connected devices talk with the control unit “speaking the same language”: devices of different builders with the same function are easily interchangeable.

2.1 EtherCAT - Atos drives

This manual will provide all information necessary to communicate and operate with Atos EtherCAT drives.

CANopen Application

The CANopen communication objects integrated in the EtherCAT protocol are transmitted at the CANopen Application. These CANopen communication objects are processed as standard CANopen communication objects.

CANopen Application is implemented in all Atos EtherCAT drives.

EtherCAT Slave Controller (ESC)

For data exchange via EtherCAT, the Atos EtherCAT drive uses a dedicated hardware system.

These devices are called EtherCAT Slave Controller (ESC).

EtherCAT protocol and addressing

EtherCAT protocol supports the auto-configuration, auto-addressing on all the modules connected in the network.

Mailbox protocol

Mailbox protocol telegram is used for the acyclic data communication (SDO CANopen communication objects).

CANopen over EtherCAT (CoE)

For EtherCAT communication all Atos EtherCAT drives use CANopen over EtherCAT (CoE).

CoE is recognized and processed by means of the Mailbox protocol and Process Data protocol.

The EtherCAT interface for Atos EtherCAT device uses the ETG.1000.6 “EtherCAT Application Layer protocol specification”, ETG.1020 “Protocol Enhancements”.

Further information of EtherCAT physical layer and CoE implementation, please refer ETG.1000 and ETG.6010 EtherCAT specifications.

EtherCAT transmits and integrates several protocols using the "Tunneling" procedure. This procedure defines the data transmission of a network protocol (CANopen objects communication according to the CiA301 protocols) integrated in other network protocol (CoE protocol).

Process Data protocol

Process Data protocol telegram is used for the cyclic data exchange (PDO CANopen communication objects).

SDO

SDO are managed by 2 SyncManager (SM0 and SM1) in the Mailbox mode

PDO

PDO are managed by 2 SyncManager (SM2 and SM3) in the Buffered mode

Extended Markup Language (XML) – see section 9

To connect easily the Atos EtherCAT drive is used the XML file.

The XML file is downloadable from MyAtos.

Object Dictionary

CoE not includes all objects present of the dictionary CANopen communication area because due to the wide difference between these data link layer, many functionality must be handled in differently mode (e.g. PDO object configuration are not included in CoE).

EtherCAT not manages others functions as Node Guarding and Timestamp.

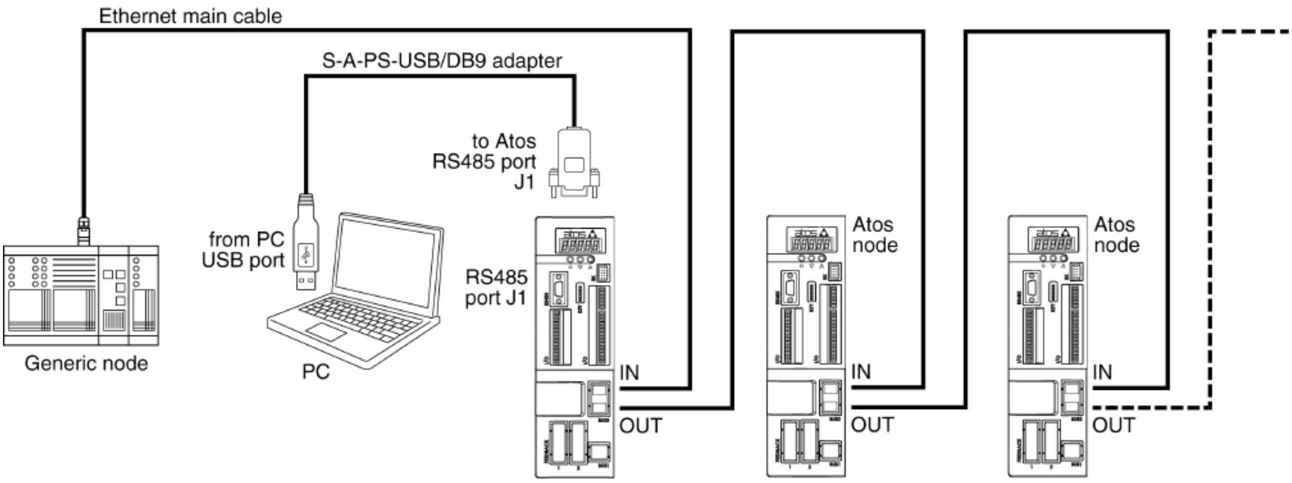
3 ETHERNET - PHYSICAL LAYER

Ethernet physical layer defines all the relevant aspects data signals transmission between devices connected to the network.

3.1 Topology

Atos recommends, like EtherCAT network topology, the classic daisy chain. In daisy chain topology the EtherCAT slave (drive) has an IN and an OUT RJ45 socket. The EtherCAT cable (coming from the direction of the master) is plugged into the IN socket. The OUT socket is connected to the next station.

For fieldbus versions, the software permits drive's parameterization through serial RS485 communication port also if the drive is connected to the central machine unit via fieldbus.



3.2 Cables

To connect the EtherCAT devices only use cables that meet the Ethernet specifications.

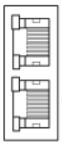
Ethernet patch or crossover cables in CAT5e quality can be used as the connection cable. CAT5e is an Ethernet network cable standard defined by the EIA/TIA. CAT5e is the fifth generation of twisted pair Ethernet technology and the most popular of all twisted pair cables in use today. CAT5e cable runs are limited to a maximum recommended run length of 100m.

Also Atos recommends shielded cables for environments where proximity to power cable, high power or RF equipments may introduce crosstalk.

Type	Signal		Distance between 2 nodes	Category
Ethernet on	100 BASE-TX	Transformer Coupling	0,2 - 100 m	CAT5 or greater

3.3 Communication connectors

For EH (EtherCAT) executions two fieldbus communication connectors are always available for digital drive. To connect the drive into the Ethernet network use dedicated RJ45 – 8 pin connectors.

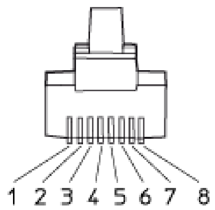
CONNECTOR	PIN	SIGNAL	TECHNICAL SPECIFICATIONS	NOTES
BUS2 	1	**TX+**	Transmitter (white/orange)	
	2	**RX+**	Receiver (orange)	
IN	3	**TX-**	Transmitter (white/green)	
	4	**NC**	-	Do not connect
OUT	5	**NC**	-	Do not connect
	6	**RX-**	Receiver (green)	
	7	**NC**	-	Do not connect
	8	**NC**	-	Do not connect

Note: for EtherCAT perform the cables connection following the IN and OUT indications

Connector to EtherCAT network (RJ45 – 8 poles - male)

The EtherCAT module incorporates two 10/100 Base TX RJ45 interfaces. The individual contacts of the RJ-45 socket are allocated as per the "T 568-B" standard. In table below are shown the pins and the colour codes of the T 568-B standard.

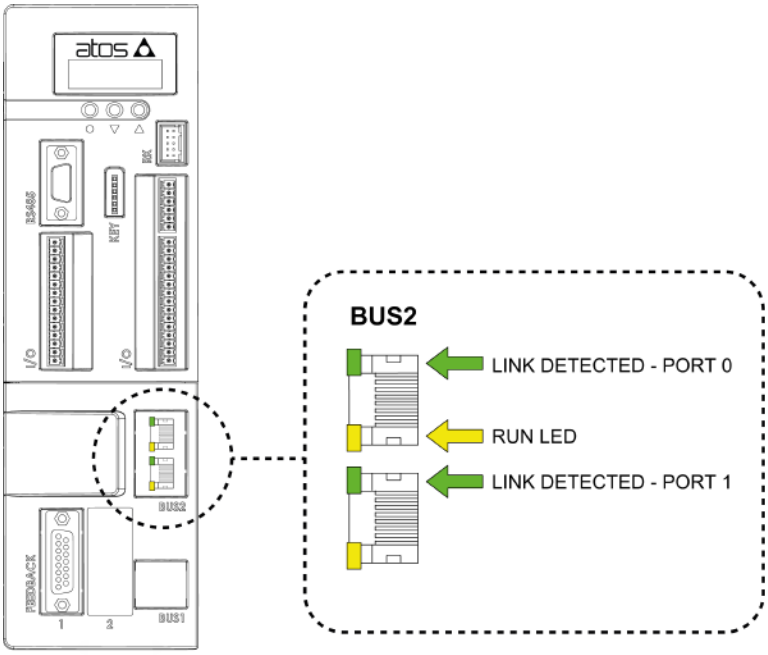
EtherCAT Input/Output drive connectors (RJ45 – 8 poles - female)

PIN	SIGNAL	COLOR (may change depending on cable)	TECHNICAL SPECIFICATIONS		RJ45 T 568-B
1	**TX+**	white/orange	Transmitter+	BI_DA+	
2	**TX-**	orange	Transmitter-	BI_DA-	
3	**RX+**	white/green	Receiver+	BI_DB+	
4		blue	(do not connect)	BI_DC+	
5		white/blue	(do not connect)	BI_DC-	
6	**RX-**	green	Receiver-	BI_DB-	
7		white/brown	(do not connect)	BI_DD+	
8		brown	(do not connect)	BI_DD-	

4 EtherCAT CONFIGURATION

The EtherCAT protocol supports the auto-configuration, auto-addressing on all the modules connected in the network, this meaning that the protocols does not require other settings in order to achieve communication.

To check that the Ethernet cable is connected to EtherCAT Module, verify if green LED (link detected LED) on the RJ45 connector is switched on. If green LED is switched off then check the cabling and also check that the master has started communications.



The EtherCAT Master scans the network to check the connected EtherCAT slaves. If the network is configured correctly the Atos drive should be visible on the master interface.

At this point, to begin the cyclic communications, set the input and output data to send cyclically.

The input and output data are the supported CAN Open objects (CoE objects). Atos drive supports the objects of Communication Profile Area (1000h – 1FFFh), Manufacturer Specific Profile Area (2000h – 5FFFh).

Cyclic data is implemented on CoE network by using "Process Data Objects". The Process Data Objects are data packets inserted in the EtherCAT frame.

The input and output data configuration can be shared in two steps:

- process data objects mapping
- process data objects assignment

5 ETHERCAT SERVICES

5.1 EtherCAT protocol

The EtherCAT is a open real-time Ethernet network. Protocol uses a special Ether type inside the Ethernet frame. The Ether type allows transport of control data directly within the Ethernet frame without redefining the standard Ethernet frame. The frame may consist of several sub-telegrams, each serving a particular memory area of the logical process images. Addressing of the EtherCAT terminals can be in any order because the data sequence is independent of the physical order.

The Atos drive supports the following EtherCAT protocol properties:

- SyncManager – see 5.2
- Distributed Clock – see 5.3
- EtherCAT State Machine – see 5.4
- CoE – see 5.5

5.2 Sync Manager

The memory of an ESC can be used for exchanging data between the EtherCAT master and application µ-controller (Atos drive µ-controller) without any restrictions. SyncManagers enable consistent and secure data exchange of these data. SyncManagers are configured by the EtherCAT master.

SyncManagers support two communication modes:

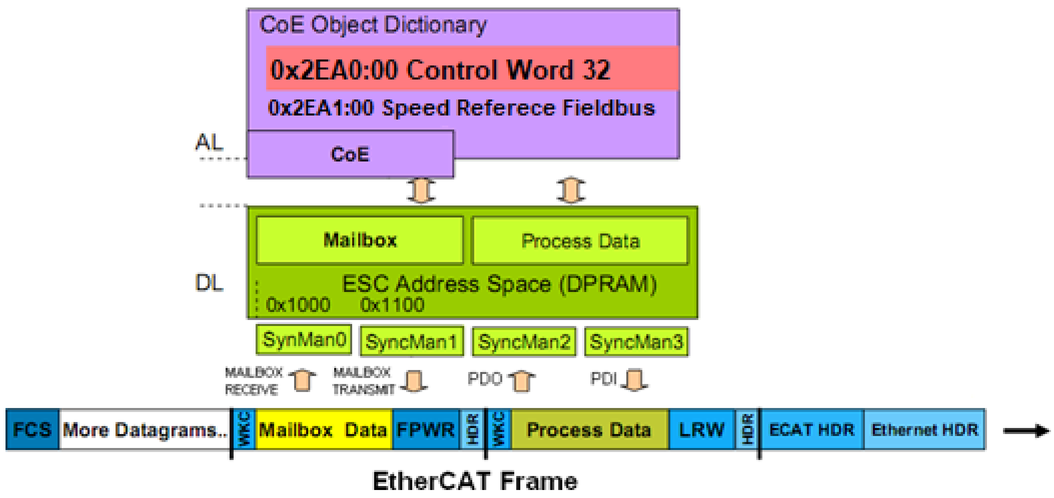
- Buffered Mode: EtherCAT master or Atos application can access to the communication buffer at any time. The buffered mode is typically used for cyclic process data.
- Mailbox Mode: EtherCAT master or Atos application can access to the communication buffer only after the other side has finished its access. In other words, if EtherCAT master is writing/reading on the communication buffer, the Atos application must wait that the communication buffer is free. The mailbox mode is typically used for application layer protocol.

Atos drive implements four SyncManagers:

two SyncManagers in mailbox mode and two SyncManagers in buffered mode.

- Sync Manager 0 is used like receive mailbox (master to slave)
- Sync Manager 1 is used like transmit mailbox (slave to master)
- Sync Manager 2 is used like process data output (master to slave)
- Sync Manager 3 is used like process data input (slave to master)

In picture below is shown the SyncManagers assigning.



5.3 Distributed Clock

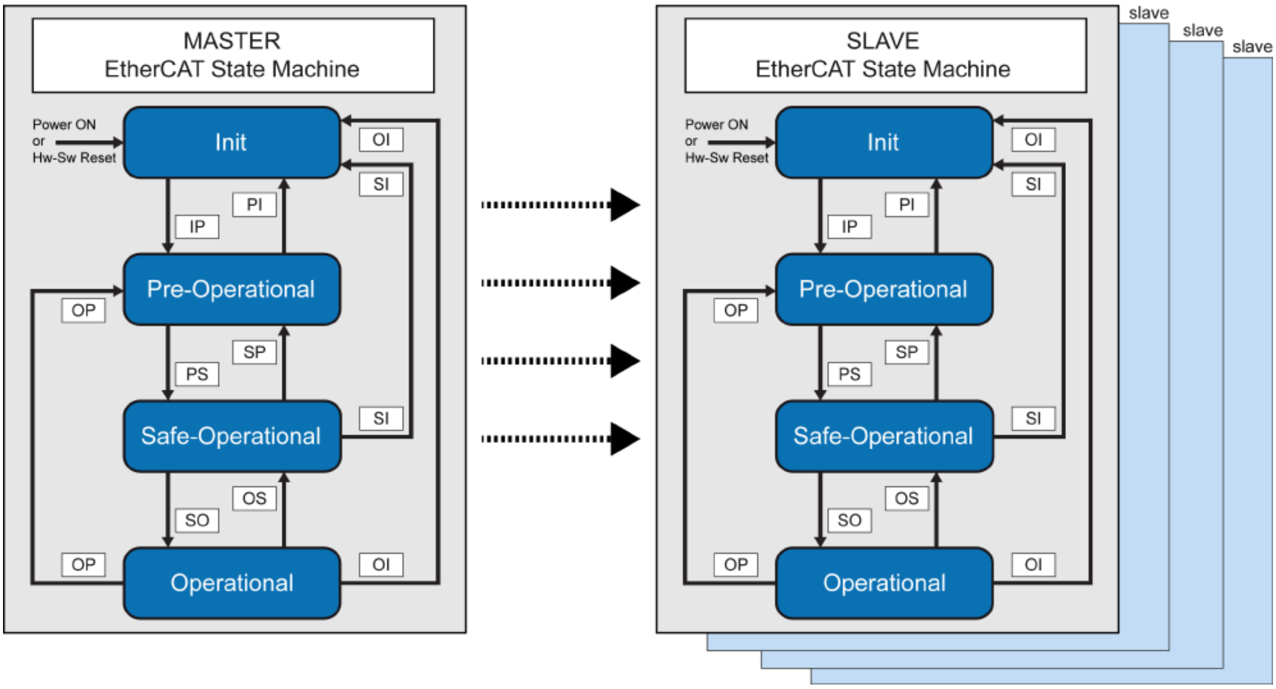
Distributed Clock is a technique of clock synchronization between the slaves and the master. DC clock synchronization enables all EtherCAT devices to share the same EtherCAT system time. In this way is possible to synchronize the local application of each device.

Accurate synchronisation is particularly important in cases where widely distributed processes require simultaneous actions. This may be the case, for example, in applications where several servo axes carry out coordinated movements simultaneously.

5.4 EtherCAT State Machine (ESM)

Each Atos EtherCAT device has implemented EtherCAT State Machine (ESM). The EtherCAT master controls every ESM of slave devices connected.

The EtherCAT State Machine (ESM) is not the Device State Machine.



State	Description
Init	Initialization phase (default state after power on) No communication (SDO / PDO) with the slave devices Device detection through the fieldbus scan
Pre-Operational	SDO communication (mailbox communication): enabled PDO communication: disable Fieldbus communication is active
Safe-Operational	SDO communication (mailbox communication): enabled PDO communication: - the input data of slave device are sent to the master and evaluated - the output data of master are not sent to the slave device - the output data remain in “Safe Operational”
Operational	SDO communication (mailbox communication): enabled PDO communication: enabled

ESM goes to the 'Safe-Operational' state if:
1. SM Watchdog is activated
2. SM Watchdog detects a PDO communication fault

Transition	Description	Status
IP (Init > Pre-Operational)	Mailbox communication	Enabled
PI (Pre-Operational > Init)	Mailbox communication	Disabled
PS (Pre-Operational > Safe-Operational)	Input update	Enabled
SP (Safe-Operational > Pre-Operational)	Input update	Disabled
SO (Safe-Operational > Operational)	Output update	Enabled
OS (Operational > Safe-Operational)	Output update	Disabled
OP (Operational > Pre-Operational)	Output update: Input update	Disabled
SI (Safe-Operational > Init)	Input update Mailbox communication	Disabled
OI (Operational > Init)	Output update Input update Mailbox communication	Disabled

⚠ The EtherCAT state machine has also the “Bootstrap” state: it is used to new firmware upload into the slave through the EtherCAT protocol. Firmware for Atos EtherCAT devices can be update through RS232 (“Bootstrap” state is not used).

EtherCAT State Machine and CANopen State Machine difference

CoE protocol use the EtherCAT state machine instead of the CANopen state machine.

Differences between state machines are:

- No direct transition to Pre-Operational after Power-On
- Reset transition to Init
- Additional state Safe-Operational (outputs in safe state)

Following are indicated the state machines differences:

EtherCAT State Machine	CANopen State Machine
Power-on	Power-on (Initialization)
Init	Stopped
Pre-Operational	Pre-Operational
Safe-Operational	/
Operational	Operational

5.5 CoE

EtherCAT provides the same communication mechanisms as the familiar CANopen mechanisms: object dictionary, PDO (process data objects) and SDO (service data objects) - even the network management is comparable.

Atos drive supports the following CoE services:

- SDO Download Expedited Request;
- SDO Upload Expedited Request;
- SDO Upload Expedited Response;
- Abort SDO Transfer Request;
- Emergency Request;
- RxPDO Transmission via mailbox;
- TxPDO Transmission via mailbox;
- RxPDO Remote Transmission Request;
- TPDO Remote Transmission Request;
- Process Data Input;
- Process Data Output;

5.5.1 Process Data Object Mapping

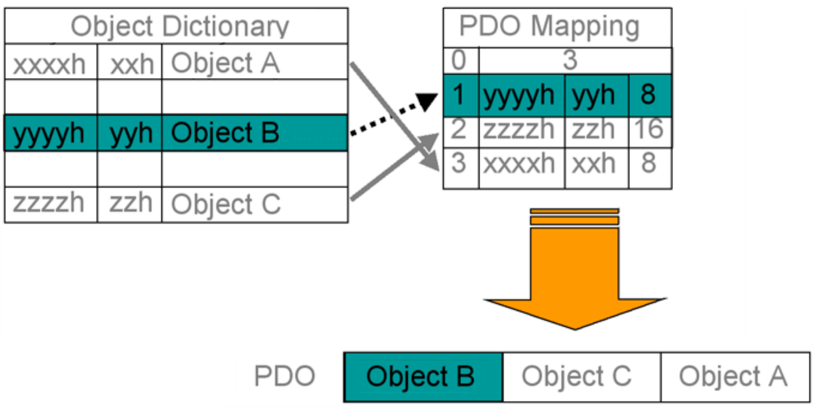
The Process Data Objects Mapping defines the content of the Process Data Objects. There are two Process Data Objects types:

- Receive Process Data Objects (RPDO)
- Transmit Process Data Objects (TPDO)

To execute the RPDO Mapping is necessary to insert in the RPDO Mapping Record the received objects. RPDO Mapping Records are the objects from 1600h to 17FFh.

To execute the TPDO Mapping is necessary to insert in the TPDO Mapping Record the objects to send. TPDO Mapping Records are the objects from 1A00h to 1BFFh.

In the picture below is shown the Process Data Objects Mapping.



5.5.2 RPDO mapping

Parameter	Index	SubIndex	Data Type
Control Word 32bit	2EA0	00 h	UNSIGNED32
Flow Setpoint Fieldbus	2EA1	00 h	INTEGER32
Pressure Setpoint Fieldbus	2EA2	00 h	INTEGER32

Examples:

- RPDO Mapping for the following objects:
  Index=2EA0h, sub-index=0h, size: 32bit – Control word 32;
  Index=2EA1h, sub-index=0h, size: 32bit – Flow Setpoint Fieldbus;
  RPDO 1 (1600h) is the Process Data Objects to set.

To configure the RPDO1 follow below steps:

- Index: 1600h Sub-index: 0 Deactivate the RPDO1 mapping
  Size: 1 byte
  Value: 0

- Index: 1600h Sub-index: 1 Insert first object of the RPDO1.
  Size: 4 byte
  To set the value, see the Object Dictionary description
  Value: 2EA00020h

- Index: 1600h Sub-index: 2 Insert second object of the RPDO1.
  Size: 4 byte
  To set the value, see the Object Dictionary description
  Value: 2EA10020h

- Index: 1600h Sub-index: 0 Set the number of mapped objects
  Size: 1 byte
  Value: 2

5.5.3 TPDO mapping

Parameter	Index	SubIndex	Data Type
Flow Setpoint Analog	2EB0	00 h	INTEGER32
Flow Demand	2EB3	00 h	INTEGER32
Speed Actual	2EB4	00 h	INTEGER32
Speed Error	2EB5	00 h	INTEGER32
Pressure Setpoint Analog	2EB6	00 h	INTEGER32
Pressure Demand	2EB9	00 h	INTEGER32
Pressure Actual	2EBA	00 h	INTEGER32
Pressure Error	2EBB	00 h	INTEGER32
Pressure PID Feed Forward	2EBC	00 h	INTEGER16
Pressure PID Proportional	2EBD	00 h	INTEGER16
Pressure PID Integral	2EBE	00 h	INTEGER16
Pressure PID Derivative	2EBF	00 h	INTEGER16
Pressure PID Output	2EC0	00 h	INTEGER32
Speed Demand	2EC1	00 h	INTEGER32
Status Word 32bit	2EC2	00 h	UNSIGNED32
Q_INPUT Actual	2EC3	00 h	INTEGER32
P_INPUT Actual	2EC4	00 h	INTEGER32
TR1 Actual	2EC5	00 h	INTEGER32
Drive IGBT Temperature	2EC6	00 h	INTEGER16
Drive Radiator Temperature	2EC7	00 h	INTEGER16
Drive CPU Temperature	2EC8	00 h	INTEGER16
Motor Temperature	2EC9	00 h	INTEGER16
Drive DC Bus Voltage	2ECA	00 h	INTEGER16
Stator Voltage	2ECB	00 h	INTEGER16
Power Actual	2ECC	00 h	INTEGER16
Flux Current PID Output	2ECD	00 h	INTEGER16
Torque Current PID Output	2ECE	00 h	INTEGER16
Torque Current Demand	2ECF	00 h	INTEGER16
Torque Current Actual	2ED0	00 h	INTEGER16
Flux Current Demand	2ED1	00 h	INTEGER16
Flux Current Actual	2ED2	00 h	INTEGER16
Motor Total Current Actual	2ED3	00 h	INTEGER16

Examples:

- TPDO Mapping for the following objects:
  Index=2EC2h, sub-index=0h, size: 32bit – Status word 32;
  Index=2EB4h, sub-index=0h, size: 32bit – Speed Actual;
  Index=2EC9h, sub-index=0h, size: 16bit – Motor Temperature;
  TPDO 2 (1A01h) is the Process Data Objects to set.

To configure the TPDO2 follow below steps:

- Index: 1A01h Sub-index: 0 Deactivate the TPDO2 mapping
  Size: 1 byte
  Value: 0

- Index: 1A01h Sub-index: 1 Insert first object of the TPDO2.
  Size: 4 byte
  To set the value, see the Object Dictionary description
  Value: 2EC20020h

- Index: 1A01h Sub-index: 2 Insert second object of the TPDO2.
  Size: 4 byte
  To set the value, see the Object Dictionary description
  Value: 2EB40020h

- Index: 1A01h Sub-index: 3 Insert third object of the TPDO2.
  Size: 4 byte
  To set the value, see the Object Dictionary description
  Value: 2EC90010h

- Index: 1A01h Sub-index: 0 Set the number of mapped objects
  Size: 1 byte
  Value: 3

5.5.4 Process Data Object Assignement

The Process Data Objects Assignment is used to control the transmission of Process Data Objects in the EtherCAT network. The Process Data Objects Assignment is supported by SyncManagers (see Protocols chapter for more information) There are two Process Data Objects Assignment types:

- PDO (master to slave)
- PDI

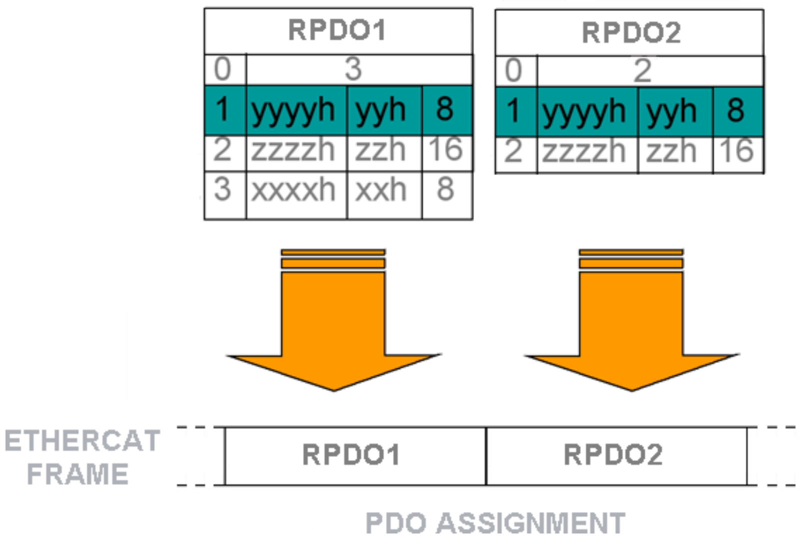
To execute the PDO assignment is necessary to insert in the Sync Manager Channel 2 the enabled RPDO.

Sync Manager Channel 2 is a record object with index 1C12h.

To execute the PDI assignment is necessary to insert in the Sync Manager Channel 3 the enabled TPDO.

Sync Manager Channel 3 is a record object with index 1C13h.

In picture below is shown the Process Data Objects Assignment.



Examples:

- PDO Assignment for the following RPDOs:
  Index=1600h – RPDO1;
  Index=1601h – RPDO2;

To configure the PDO (1C12h) follow below steps:

- Index: 1C12h Sub-index: 0 Deactivate the PDO assignment
- Size: 1 byte
- Value: 0

- Index: 1C12h Sub-index: 1 Insert the RPDO1 in the PDO.
- Size: 2 byte To set the value, see the Object Dictionary description
- Value: 1600h

- Index: 1C12h Sub-index: 2 Insert the RPDO2 in the PDO.
- Size: 2 byte To set the value, see the Object Dictionary description
- Value: 1601h

- Index: 1C12h Sub-index: 0 Set the number of assigned RPDO
- Size: 1 byte
- Value: 2

Examples:

- PDI Assignment for the following TPDO:
  Index=1A01h – TPDO2;

To configure the PDI (1C13h) follow below steps:

- Index: 1C13h Sub-index: 0 Deactivate the PDI assignment
- Size: 1 byte
- Value: 0

- Index: 1C13h Sub-index: 1 Insert the TPDO2 in the PDI.
- Size: 2 byte To set the value, see the Object Dictionary description
- Value: 1A01h

- Index: 1C13h Sub-index: 0 Set the number of mapped objects
- Size: 1 byte
- Value: 1

6 OBJECT DICTIONARY

6.1 Communication profile area

The following objects of the communication profile are supported:

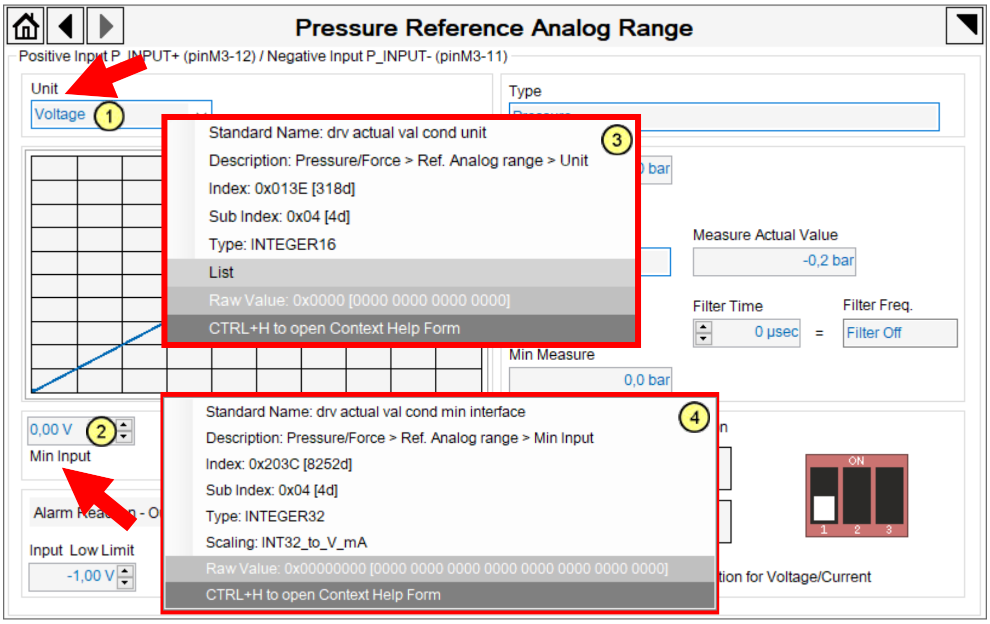
Index (hex)	Object	Name	Type	Access
1000	VAR	Device type	UNSIGNED32	Reading
1001	VAR	Error register	UNSIGNED8	Reading
1002		Reserved		
-	-	-	-	-
1007		Reserved		
1008	VAR	Manufacturer device name	Vis-String	constant
1009	VAR	Manufacturer hardware version	Vis-String	constant
100A	VAR	Manufacturer software version	Vis-String	constant
100B		Reserved		
-	-	-	-	-
1017		Reserved		
1018	RECORD	Identity Object	Identity (23h)	Reading
1019		Reserved		
-	-	-	-	-
15FF		Reserved		
1600	RECORD	1st receive PDO mapping	PDO Mapping	Reading/writing
1601	RECORD	2nd receive PDO mapping	PDO Mapping	Reading/writing
1602	RECORD	3rd receive PDO mapping	PDO Mapping	Reading/writing
1603	RECORD	4th receive PDO mapping	PDO Mapping	Reading/writing
1604		Reserved		
-	-	-	-	-
19FF		Reserved		
1A00	RECORD	1st transmit PDO mapping	PDO Mapping	Reading/writing
1A01	RECORD	2nd transmit PDO mapping	PDO Mapping	Reading/writing
1A02	RECORD	3rd transmit PDO mapping	PDO Mapping	Reading/writing
1A03	RECORD	4th transmit PDO mapping	PDO Mapping	Reading/writing
1A04		Reserved		
-	-	-	-	-
1BFF		Reserved		
1C00	ARRAY	Sync Manager Communication Type	UNSIGNED8	Reading
1C01		Reserved		
-	-	-	-	-
1C0F		Reserved		
1C10	ARRAY	Sync Manager 0 PDO Assignment	UNSIGNED16	Reading/writing
1C11	ARRAY	Sync Manager 1 PDO Assignment	UNSIGNED16	Reading/writing
1C12	ARRAY	Sync Manager 2 PDO Assignment	UNSIGNED16	Reading/writing
1C13	ARRAY	Sync Manager 3 PDO Assignment	UNSIGNED16	Reading/writing
1C14		Reserved		
-	-	-	-	-
1FFF		Reserved		

6.2 Manufacturer specific profile area

Directly from the graphical interface of the S-SW-SETUP software, it is possible to access information useful for the development of fieldbus communication by simply clicking with the mouse on a selected parameter or pressing CTRL+H on the PC keyboard.

Mouse click- example:

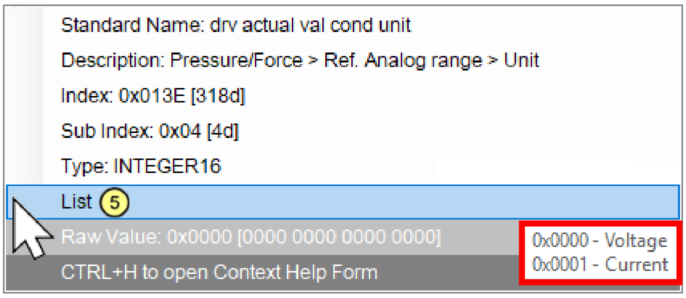
Click right button of the mouse on **Unit** control (1) or **Min Input** control (2) to open the related fieldbus communication windows ((3) and (4)).



Windows (1) and (4) shown all fieldbus information of selected parameter:

Standard Name:	Parameter name
Description:	Parameter fast reference to S-SW-SETUP software and drive manual descriptions
Index: Sub Index:	Parameter address
Type:	Parameter dimension and data type
Scaling:	Parameter scaling value (see 7 and 7.2)
List:	Parameter list value
Raw Values	Numeric parameter
CTRL+H	Press CRTL+H to open Context Help Form (see **CRTL+H - example**)

(1) Pass mouse arrow on **List** (5) to display the information



CRTL+H - example:

The information in Context Help Form window are the same as described above for the "mouse click - example".

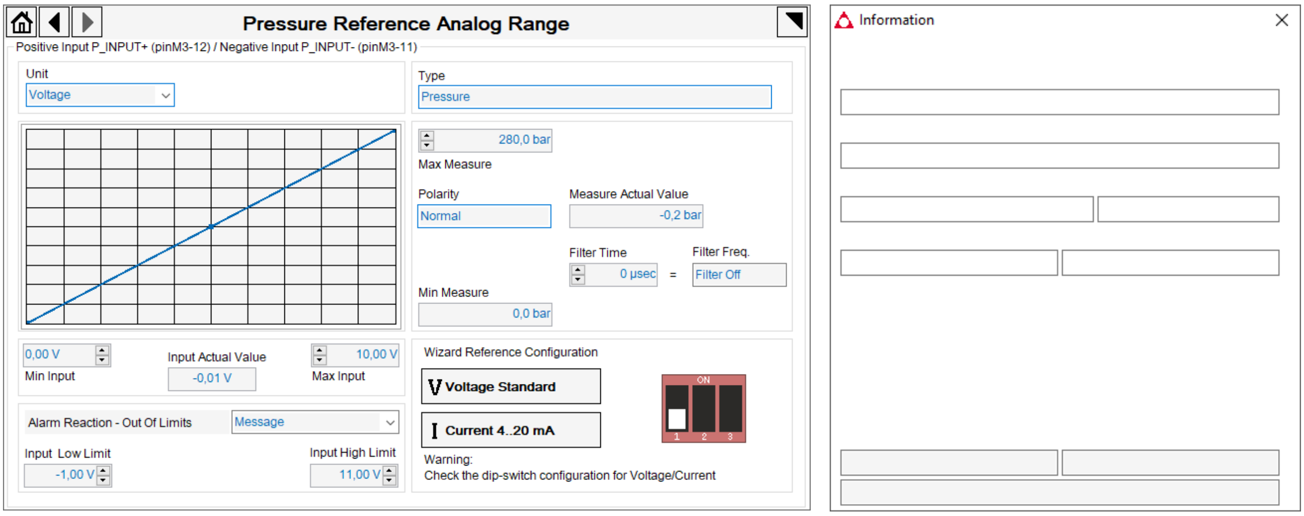
By pressing CRTL+H on PC keybord the Context Help Form windows opens.

Once the window has been opened with CTRL + H, it always remains active until it is closed by the user.

In this way it is possible to view the fieldbus information of each single parameter present on the software page, simply by positioning the mouse over it.

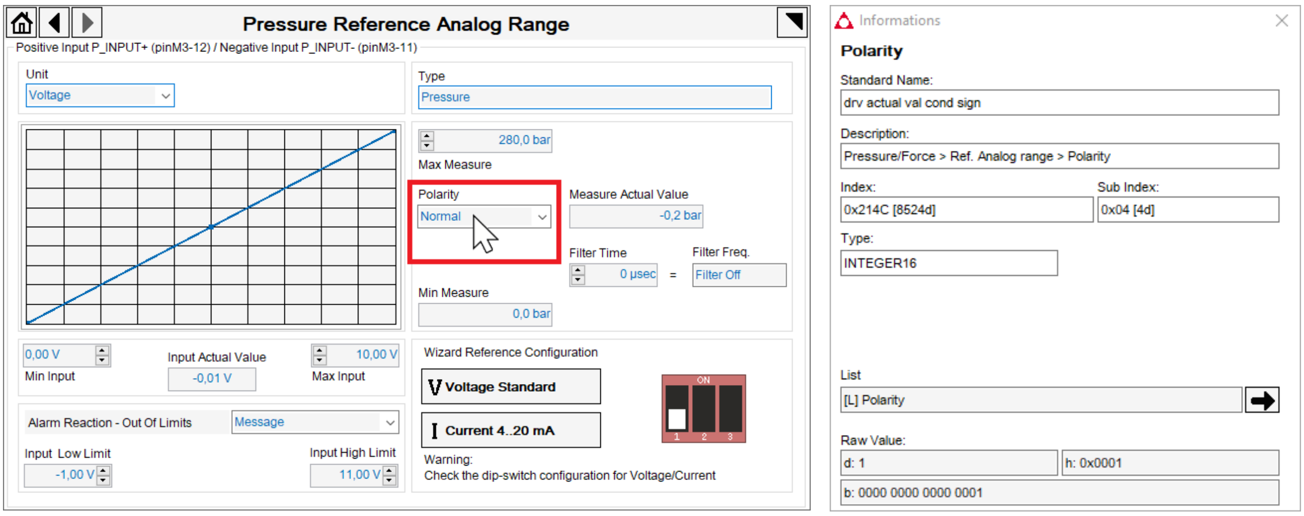
If no parameter is selected the fields of the window will be appears empty.

Example: no parameter selected



Pass the mouse on a parameter to display all fieldbus communication information related to it.

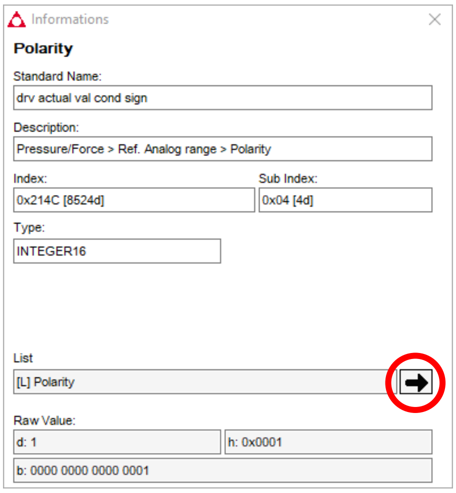
Example: “Polarity” parameter selected



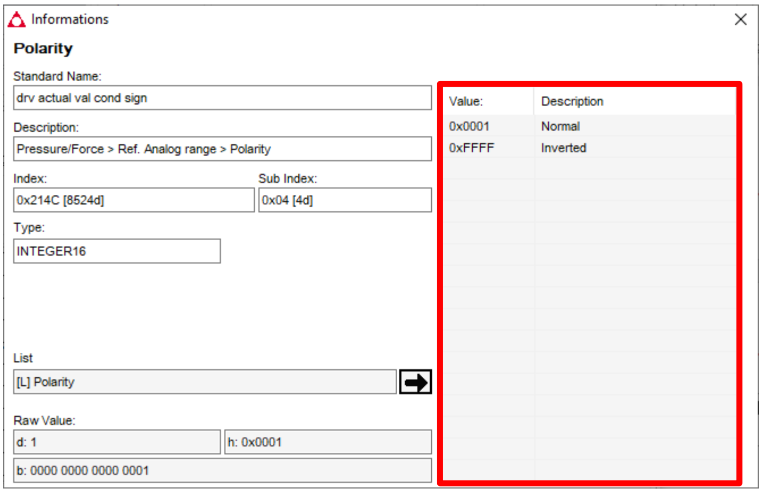
When in a parameter is present a “List” [L] press the black arrow to display the related information of the list (Value and Description).

Example: [L] Polarity

Press the black arrow



List information are displayed



7 SCALING DESCRIPTION

7.1 Internal resolution scaling

These scaling convert ‘Raw’ value with internal resolution formats in ‘Real’ value [Real Unit]:

Scale	Gain Raw to Real	Gain Real to Raw	Real Unit
Raw_mbar	1	1	mbar
Raw_mrpm	1	1	mrpm
Raw_µsec	1	1	µsec
Raw_mHz	1	1	mHz
Raw_msec	1	1	msec
Raw_mm	1	1	mm
INT16_to_A_X16	0,0625	16	A
INT16_to_V_X16	0,0625	16	V
INT16_to_°C_X16	0,0625	16	°C
INT16_to_kW_X16	0,0625	16	kW
INT16_to_Perc200	200 / 32767	32767 / 200	%
UINT16_msec_to_sec	0,001	1000	sec
INT32min_to_gg_hh_mm	1	1	min
INT16_to_V_mA	0,0001	10000	V
	0,0002	5000	mA
8192_to_200Perc	200 / 8192	8192 / 200	%

Real_Value [Real Unit] = Gain_Raw_to_Real ∗ Raw_Value

Raw_Value = Gain_Real _to_Raw ∗ Real_Value [Real Unit]

7.2 VALUE to Physical scaling

These scaling depend by the input interface. In the below table are described the input interface types:

Input interface	Scaling	Measure Unit
Pressure	RAW_mbar	mbar
Speed	RAW_mrpm	mrpm

7.3 Gain scaling

These scalings convert 32bits ‘raw’ values into a Gain factor with unit indication if required:

Scale	Real Full Scale	Real Unit
INT32_to_Gain	1	none

The function use the two words of value to calculate the gain real value:

Raw Value	**A_raw**: (Most Significant Word)	**B_raw**: (Less Significant Word)

where for Gain function:

Real_Gain = (A_raw / B_raw) ∗ Real_Fullscale

8 BITS PARAMETERS DESCRIPTIONS

8.1 Status Word - 32bit

Index	2EC2h

SubIndex	00h

Conditions:

MSB

Bit	31-28	27	26	25	24	23-22	21-20	19-16
Content	Res	**STO Test Suggested**	**STO Corrupted**	**STO Active**	**Pump Overheat Protection Active**	**Smart Selection**	**Pressure PID Selection**	Res

LSB

15	14	13-12	11	10	9	8	7	6-0
Drive Running	**Control Error**	Res	**Internal Limit Reached**	**Pressure Target Reached**	**Local**	**Power Limitation Active**	**Warning**	**Status**

Status

The first seven bits 6-0 indicate the functional status of the electronic drive:

Status value (bits 6-0)	
Not rerady to switch on	0xx 0000
Switch on disable	1xx 0000
Ready to switch on	01x 0001
Switched on	01x 0011
Operation enabled	01x 0111
Quick stop active	00x 0111
Fault reaction active	0xx 1111
Fault	0xx 1000

Warning

Bit 7 indicates the presence of alarm or error conditions:

Warning (bit 7)	
Normal working	0
Error/Alarm present	1

Power Limitation Active

Bit 8 indicates if the control (Power Limitation Active) is active or is not active:

Power Limitation Active (bit 8)	
No active	0
Active	1

Local

Bit 9 indicates if the drive status is actually controlled by fieldbus (see 8.2 – control word) or not:

Local (bit 9)	
Local (internal) control	1
Remote (fieldbus) control	0

Pressure Target Reached

Bit 10 indicates when the actual regulated pressure has reached the demanded value:

Pressure target reached (bit 10)	
Not reached	0
Reached	1

Limit Touched

Bit 11 indicates when the demanded pressure value is out of limit:

Limit Touched (bit 11)	
Limit not touched	0
Limit touched	1

Control Error

Bit 14 indicates when a pressure error is present:

Control Error (bit 14)	
Normal working	0
Control error present	1

Drive running

Bit 15 indicates when the drive is running:

Drive running (bit 15)	
Drive stop	0
Drive running	1

Pressure PID Selection

Bits 21-20 indicate which PID parameters set is active for pressure control:

Pressure PID selection (bits 21-20)	
PID1	00
PID2	01
PID3	10
PID4	11

Smart Selection

Bits 23-22 indicate which smart parameters set is active for smart control:

Pressure PID selection (bits 21-20)	
Dynamic	00
Balanced	01
Smooth	10

Pump Overheat protection active

Bit 24 indicates if the control (Pump Overheath protection) is active or is not active:

Pump Overheat protection active (bit 24)	
No active	0
Active	1

STO active

Bit 25 indicates if the STO function is active or is not active:

STO active (bit 24)	
No active	0
Active	1

STO corrupted

Bit 26 indicates if the STO function is corrupted or is no corrupted:

STO corrupted (bit 26)	
No corrupted	0
Corrupted	1

STO test suggested

Bit 27 indicates if the STO function is corrupted or is ok:

STO test suggested (bit 27)	
No test suggested	0
Test suggested	1

8.2 Control Word - 32bit

Index	2EA0h

SubIndex	00h

Conditions:

MSB						*LSB*
Bit	**31-16**	**15-14**	**13-12**	**11**	**10-8**	**7-0**
Content	Res	**Smart Selection**	**Pressure PID Selection**	**Pressure Control Enable**	Res	**Control**

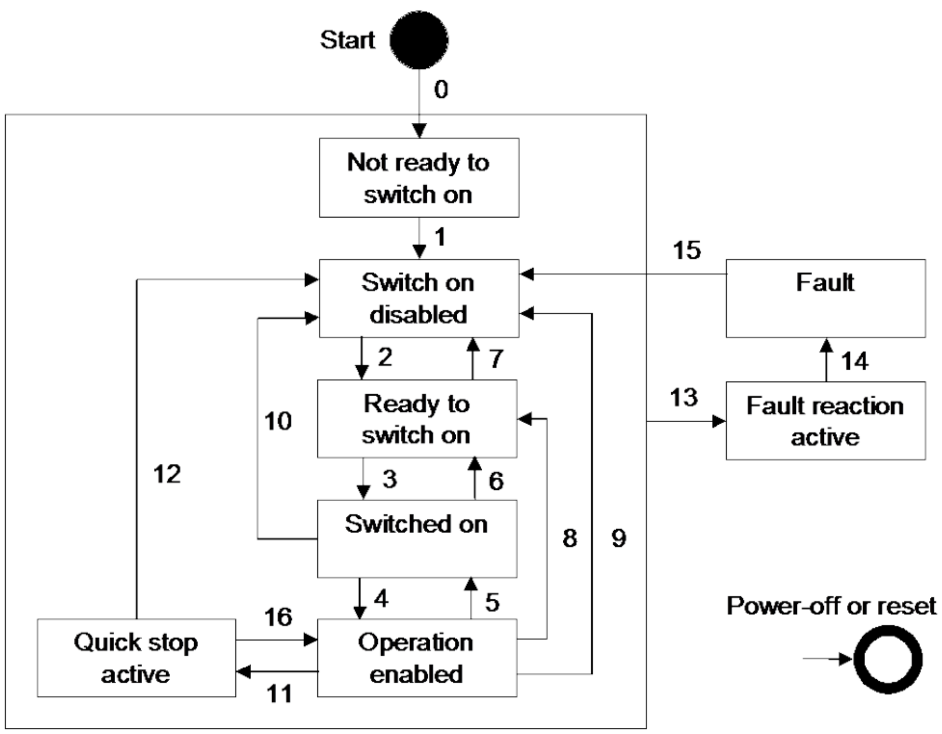
Control

The first eight bits 7-0 allow to request the transition of the drive status to a defined condition:

Command	Control Word Bits					Transition
	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	
Shutdown	0	X	1	1	0	2 , 6 , 8
Switch on	0	0	1	1	1	3
Switch on + enable operation	0	1	1	1	1	3 + 4 **(*)**
Disable voltage	0	X	X	0	X	7 , 9 , 10 , 12
Quick stop	0	X	0	1	X	7 , 10 , 11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4 , 16
Fault reset		X	X	X	X	15

(*) Automatic transition to enable operation state after executing switched on state functionality

Note: bits 6, 5, 4 of the controlword are not used.



Pressure Control Enable

Bit 11 allows to select the Enable when the alternated control is active:

Pressure Control Enable (bit 11)	
Speed control active	0
Alternated control active	1

Pressure PID Selection

Bits 13-12 allows to select the active Pressure PID parameters set:

Pressure PID selection (bits 13-12)	
PID1	00
PID2	01
PID3	10
PID4	11

Smart Selection

Bits 15-14 indicate which smart parameters set is active for smart control:

Pressure PID selection (bits 15-14)	
Dynamic	00
Balanced	01
Smooth	10

9 CONFIGURATION FILE (XML)

An electronic description of Atos drives EtherCAT characteristics is available through XML (Extensible Markup Language) files configuration. These files, included in MyAtos, list the communication features and the accessible parameters thus allowing to speed up configuration process of fieldbus master devices.

the smart electrohydraulics