atos
the smart electrohydraulics

USER MANUAL

S-MAN-S-EP

SMART SERVOPUMP SYSTEM SERIES 20
PROGRAMMING INSTRUCTIONS
PROFINET RT/IRT PROTOCOL

INDEX

1 GENERAL

1.1 About this user manual

This manual describes the required information to operate Atos Smart ServopPump system (SSP) using PROFINET 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 PROFINET communication.

The purpose of this manual is not to cover all the details or variations of PROFINET 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
- 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

1.3 Trademarks

1.4 Abbreviations

Abbreviation	Description
API	Application Process Instance
AR	Application Relation
ADU	Application Data Unit
CR	Communication Relation
CC-A	Conformance Class A
CC-B	Conformance Class B
CC-C	Conformance Class C
DCP	Discovery and Basic Configuration Protocol
DHCP	Dynamic Host Configuration Protocol
EDS	Electronic Data Sheet
EMC	ElectroMagnetic Compatibility
ESD	ElectroStatic Discharge
FTP	File Transfer Protocol
GSD	General Station Description
GSDML	GSD Markup Language
HICP	Anybus IP config protocol
HMI	Human Machine Interface
I&M	Identification & Maintenance
IEC	International Electrotechnical Commision
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standard Organization
IP	Internet Protocol
IRT	Isochronous Real Time
LAN	Local Area Network
MAC	Medium Access Control
MRP	Media Redundancy Protocol
NMT	Network Managment
PNIO	ProfiNet Input Output
PROFINET	Process Field Network
RT	Real Time
SNMP	Simple Newtwork Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
USB	Universal Serial Bus
	Active on rising edge

2 ABOUT PROFINET IO RT/IRT INTERFACE - ETHERNET FIELDBUS

PROFINET IO (Input - Output) enables the connection of decentralized field devices, such as I/Os, drives, transducers or analysis devices to a central automation device, such as a PLC, PC or process control system. It can be considered as the direct successor of PROFIBUS DP.

Data transfer is based on the Fast Ethernet standard transmission with 100 Mbit/s.

PROFINET IO follows the provider-consumer model for the data exchange.

Every PROFINET Device has a defined set of features which are mandatory based on its conformance class. The conformance classes are divided into three categories:

Conformance Class A (CC-A)

Is the most basic and provides basic functions for PROFINET IO with RT communication provides real time and acyclic real time, as well as support for standard TCP/IP and basic functions such as topology information.

Conformance Class B (CC-B)

Adds in simple network management protocol (SNMP) support to make it possible to read statistics with standard SNMP tools.

Conformance Class C (CC-C)

Is the most advanced and it should be noted that they build upon each other, so Class C also contains all functions from Class B and A. Conformance Class C has support for motion control applications with a jitter of less than a microsecond and distributed clock synchronization protocol (allowing deterministic Ethernet) based on hardware-supported bandwidth reservation and synchronization (IRT communication). Conformance Class C is the basis for isochronous applications.

	CC-A	CC-B	CC-C
Real-Time Data Exchange - cycle times down to 1ms	●	●	●
Alarm and Diagnostic	●	●	●
Network Topology Support	●	●	●
SNMP Support		●	●
Real-Time Data Exchange - cycle times down to 31.25us			●

The PROFINET IO connection can be configured for both acyclic and cyclic I/O data:

Acyclic Data

Are sent only when it is needed. It is useful for non time critical parameters , or for very large data transfers. For example, when you need to send or read some setting parameter from the PLC that doesn't need any type of time synchronization behaviour.

Cyclic Data

Are defined to as "I/O" and are time-critical are used for real-time data exchange (speed and low latency are important) Cyclic I/O data is always sent between the PLC and I/O device at the specified update time. For example, status information from the Atos IO Device, would typically be part of the cyclic data. Cyclic I/O data is defined by the Incoming (Input) and Outgoing (Output) Cyclic I/O Data.

3 ETHERNET - PHYSICAL LAYER

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

3.1 PROFINET Network

Following a short description of the terms used in a PROFINET network, into this guide:

IO Controller: master (e.g. PLC) or controlling device (e.g. HMI) that initiates implicit communication with IO devices.

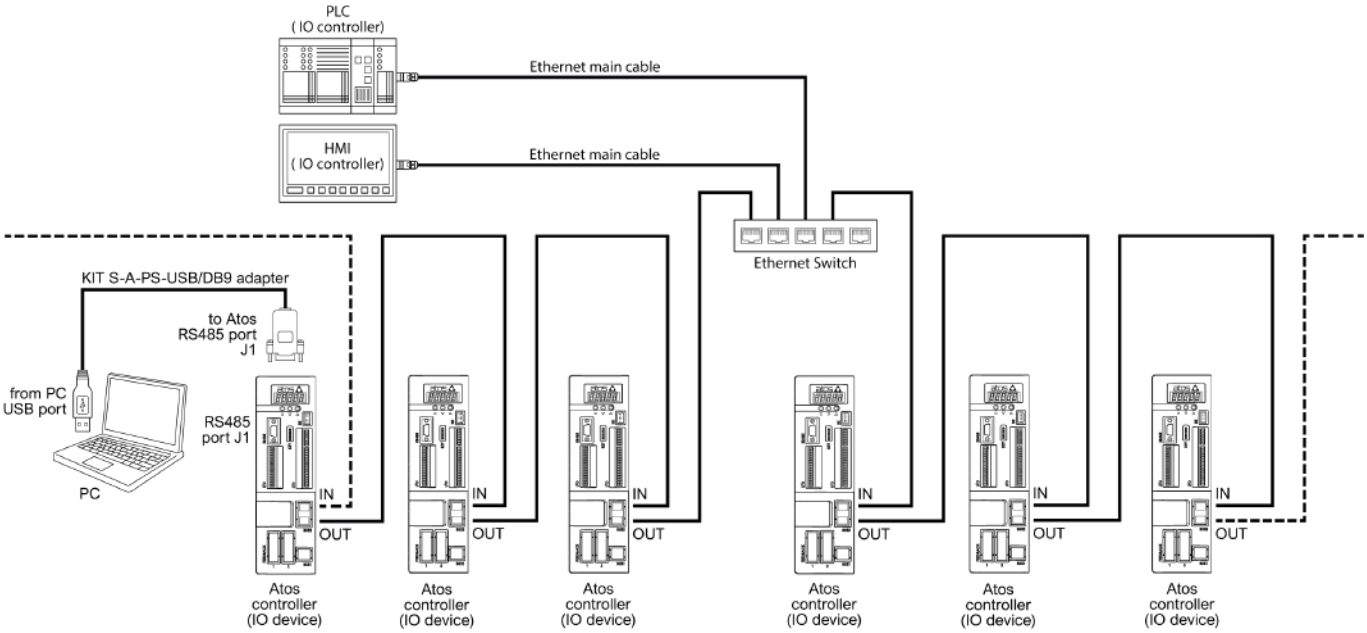
IO Device: device that receives a connection request to PROFINET network. One or more IO Devices can be connected to the IO Controller on a PROFINET network, also through Ethernet switches (see 3.1.1). IO Devices are also an explicit messages server.

The Atos drives are IO Devices.

3.1.1 Wiring PROFINET Network

Atos recommends, like PROFINET network topology, the classic daisy chain. Atos IO Devices are supplied with RJ-45 socket communication connectors (see 3.3) to manage two integrated Ethernet ports: this feature allows to use the two ports on the device interchangeably. Connect one port of the Atos IO Device to an port on the IO Controller (e.g. PLC or HMI) or to one port of Ethernet Switch as showed below.

Network configuration is for the most part performed automatically by the IO Controller (e.g. PLC). Please refer the PLC user guide for complete information on network configuration.



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 PROFINET devices only use cables that meet the Ethernet specifications.

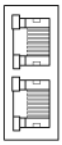
Ethernet patch or crossover cables in CAT5e quality can be used as the connection cable. CAT5e is an Ethernet network cable standard defined by the EIA/TIA. CAT5e is the fifth generation of twisted pair Ethernet technology and the most popular of all twisted pair cables in use today. CAT5e cable runs 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 EP (PROFINET) 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.

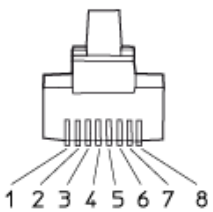
CONNECTOR	PIN	SIGNAL	TECHNICAL SPECIFICATIONS	NOTES
BUS2 	1	**TX+**	Transmitter (white/orange)	
	2	**RX+**	Receiver (orange)	
	3	**TX-**	Transmitter (white/green)	
	4	**NC**	-	Do not connect
	5	**NC**	-	Do not connect
	6	**RX-**	Receiver (green)	
	7	**NC**	-	Do not connect
	8	**NC**	-	Do not connect

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

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

The PROFINET 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.

PROFINET 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 PROFINET CONFIGURATION

The Atos drive is a modular PROFINET slave, classified as a generic I/O Device: this allow an extended compatibility with most PNIO controllers and require no specific device profile support.

Modular PROFINET devices have slots (MODULES) that can be populated with different kind of sub- slot (SUBMODULES) and so can be configured for different functions.

Specifically, the drive is defined as a RT and IRT capable device with 3 modules:

- MAIN module
- INPUTS module
- OUTPUTS module

The Main module contains 3 standard sub-modules for device I&M (Identification and Maintenance) with data and status about the device and link ports.

The Inputs module must be populated with 1 sub-module that can be choosed among a collection of different sized data INPUT, from 1 up to 10 words.

The Outputs module must be populated with 1 sub-module that can be choosed among a collection of different sized data OUTPUT, from 1 up to 10 words.

Input and Output are related to PNIO controller point of view, so input is data send by the slave to the master and output is data send by the master to the slave.

In order to accept CR (Communication Relation) for cyclic data drive requires that the size of input data matches the size of output data so Input and Output sub-modules have to be of same size. Also, the data size configured into the modules in PNIO controller configurator tool have to match to the size of data configured into drive by Cyclic Mapping.

PNIO controller network configuration tool allow also for setting the device name and IP address (by DCP services).

The PROFINET master uses device name and IP address to univocally identify the devices connected to the network and to establish AR (Application Relation) to them so the default name and IP address of drive have to be changed properly.

Please notice that while the device name assigned by master via DCP can be permanentely stored by drive, the IP address will be over-written at next power-up or reset so the IP address assigned into the network configurator have to be also setted into drive configuration by S-SW-SETUP (see IP Addressing configuration)

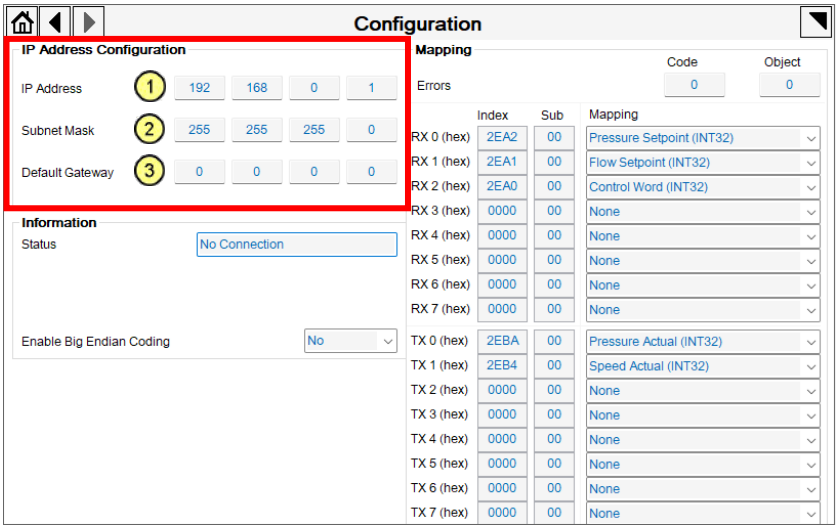
4.1 IP Addressing configuration

IP Address, Subnet Mask and Default Gateway are assigned by fieldbus master (e.g. Discovery and Configuration Protocol) and are used by the drive for communication. The IP address must be set univocally in the network in order to allow proper communication. The subnet mask and gateway allow identification of local sub-network boundaries and specify a path to reach outside hosts.

PNIO controller (PROFINET master) usually can set the device name, IP address, subnet mask and gateway of PNIO devices (PROFINET slaves) by DCP services.

Although all these can be temporary changed by the master via DCP, only the device name can be permanent stored by the drive: at power-up all other settings are over-written with the values saved into drive parameters.

Image below is referred at the “Network management > Configuration” S-SW-SETUP software page.



IP Address Configuration

IP Address (1), Subnet Mask (2) and Default Gateway (3) are assigned by fieldbus master (e.g. Discovery and Configuration Protocol) or IPconfig.

IPconfig is a software directly provided by Atos that allows to identify only PROFINET Atos IO Device: it can be downloaded from My Atos.

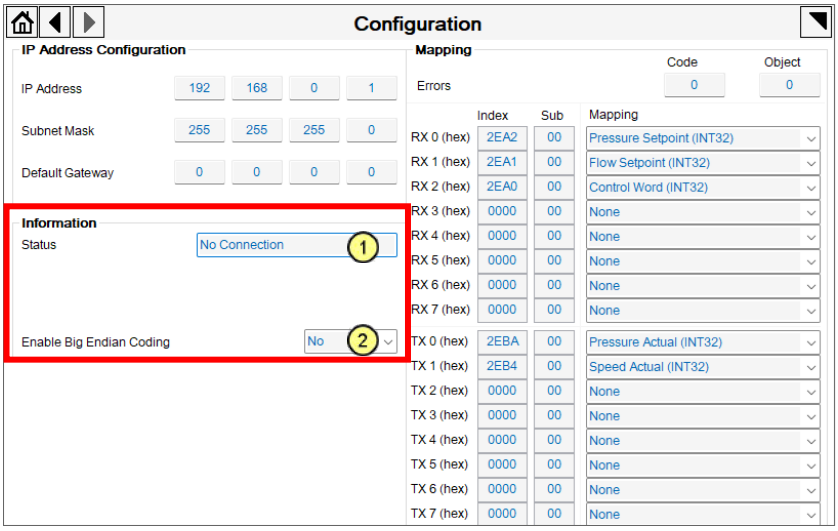
Atos IO device are supplied by factory default with an unique MAC Address (shown on Atos device label). This allows to communicate with the Atos IO device in order to configure the IP address for the network.

4.2 Fieldbus status

This read-only parameter monitor the state of filedbus interface:

Setup	Fieldbus setup in Progress
Init	Fieldbus interface in Initialization
Wait Process Data	AR/CR with PNIO controller (Profinet Master) are ON, but no cyclic data are exchanging
No Connection	The network is idle. This happens if there are no connection or if no Profinet Master has established an AR/CR with the drive
Process Data Exchange	AR/CR with PNIO controller (Profinet Master) are ON and cyclic data are exchanging
Error	Fieldbus Error
Exception	Fatal Error on the PROFINET Module

Image below is referred at the “Network management > Configuration” S-SW-SETUP software page.



Informations - Status

The indicator (1) allows to display the status of the fieldbus communication.

Informations - Enable Big Endian Coding

The control (2) allow to set the order in which a sequence of bytes is stored.

Selecting "Option 2 (Yes)" each character is sent in reverse order (LSB to MSB) and each word is sent from high to low byte (Big Endian coding is enabled).

Option1	No
Option2	Yes
Default	*No*

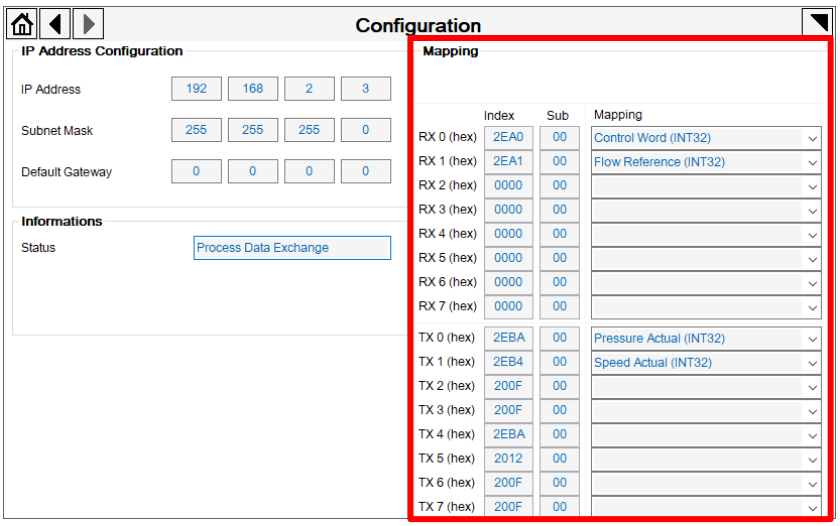
5 PROFINET SERVICES

5.1 Cyclic Communication

Cyclic communication consists of few data (maximum 10 words for each direction) exchanged quickly and frequently (from some ms down to hundreds µs cycle): I/O, diagnostic, set point, internal values… the so called "PROCESS DATA".

5.1.1 Mapping

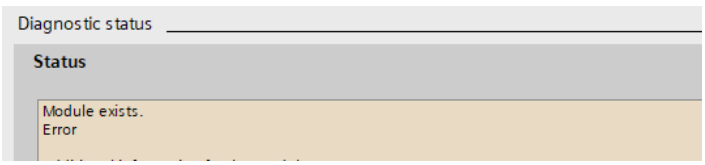
It's possible to map the desired objects with S-SW-SETUP at the "Network management > Configuration" software page (see image below) that allow mapping up to 8 objects both in transmission and reception.



For the process area configuration:

- Programme the objects being received and transmitted by indicating the index and sub-index of the objects in the CANopen Dictionary (the sub-index is 0 for VAR objects or the array index for ARRAY objects). Every mapped object can be 16 or 32 bit wide and then use 1 or 2 data words. The max number of words can be 10 in RX and 10 in TX.
  RX and TX direction are related to the drive point of view so:
  - **RX** refers to data sent by the master to the slave (**master data output**)
  - **TX** refers to data sent by the slave to the master (**master data input**)
- Store parameters: use STORE USER button of the S-SW-SETUP (see S-MAN-SW manual)
- Switch the drive off and then on again

To reach the "process Data Online" status, the number of cyclic data exchange WORDs set in the PLC must correspond to the number of WORDs configured in the DRIVE. Otherwise the data exchange cannot take place and the PLC will show the Module Error status.



Pay attention that the UINT32 or INT32 parameters are to be considered as modules of 2 WORDs each.

5.1.2 Input data mapping (RX)

Name	Description
RX0_INDEX	Receive Object0 Index
RX0_SUB_INDEX	Receive Object0 Sub-Index
RX1_INDEX	Receive Object1 Index
RX1_SUB_INDEX	Receive Object1 Sub-Index

RX7_INDEX	Receive Object7 Index
RX7_SUB_INDEX	Receive Object7 Sub-Index

5.1.3 RX mapping

Parameter	TX Mapping		Manufacturer specific profile area		Data Type
	Index	SubIndex	Index	SubIndex	
Control Word	2EA0	00 h	0134	00 h	UNSIGNED32
Flow Setpoint	2EA1	00 h	0132	00 h	INTEGER32
Pressure Setpoint	2EA2	00 h	0144	00 h	INTEGER32

RX mapping example:

Name	Index	Description
RX0_INDEX	2EA0	Control Word 32bit
RX0_SUB_INDEX	00	
RX1_INDEX	2EA1	Flow Setpoint Fieldbus
RX1_SUB_INDEX	00	

5.1.4 Output data mapping (TX)

Name	Description
TX0_INDEX	Transmit Object0 Index
TX0_SUB_INDEX	Transmit Object0 Sub-Index
TX1_INDEX	Transmit Object1 Index
TX1_SUB_INDEX	Transmit Object1 Sub-Index

TX7_INDEX	Transmit Object7 Index
TX7_SUB_INDEX	Transmit Object7 Sub-Index

5.1.5 TX mapping

Parameter	TX Mapping		Manufacturer specific profile area		Data Type
	Index	SubIndex	Index	SubIndex	
Flow Setpoint Analog	2EB0	00 h	03C6	00 h	INTEGER32
Flow Demand	2EB3	00 h	03D8	00 h	INTEGER32
Speed Actual	2EB4	00 h	03E0	00 h	INTEGER32
Speed Error	2EB5	00 h	03FC	00 h	INTEGER32
Pressure Setpoint Analog	2EB6	00 h	03C4	00 h	INTEGER32
Pressure Demand	2EB9	00 h	03EA	00 h	INTEGER32
Pressure Actual	2EBA	00 h	03D2	00 h	INTEGER32
Pressure Error	2EBB	00 h	03FA	00 h	INTEGER32
Pressure PID Feed Forward	2EBC	00 h	n.a.	n.a.	INTEGER16
Pressure PID Proportional	2EBD	00 h	n.a.	n.a.	INTEGER16
Pressure PID Integral	2EBE	00 h	n.a.	n.a.	INTEGER16
Pressure PID Derivative	2EBF	00 h	n.a.	n.a.	INTEGER16
Pressure PID Output	2EC0	00 h	03DC	00 h	INTEGER32
Speed Demand	2EC1	00 h	03DE	00 h	INTEGER32
Status Word	2EC2	00 h	03EE	00 h	UNSIGNED32
Q Input Actual	2EC3	00 h	03C2	00 h	INTEGER16
P Input Actual	2EC4	00 h	03C0	00 h	INTEGER16
TR1 Actual	2EC5	00 h	03CC	00 h	INTEGER32
Drive IGBT Temperature	2EC6	00 h	03AD	00 h	INTEGER16
Drive Radiator Temperature	2EC7	00 h	0399	00 h	INTEGER16
Drive CPU Temperature	2EC8	00 h	03A8	00 h	INTEGER16
Motor Temperature	2EC9	00 h	039A	00 h	INTEGER16
Drive DC Bus Voltage	2ECA	00 h	0398	00 h	INTEGER16
Stator Voltage	2ECB	00 h	n.a.	n.a.	INTEGER16
Power Actual	2ECC	00 h	0381	00 h	INTEGER16
Flux Current PID Output	2ECD	00 h	0396	00 h	INTEGER16
Torque Current PID Output	2ECE	00 h	0394	00 h	INTEGER16
Torque Current Demand	2ECF	00 h	0387	00 h	INTEGER16
Torque Current Actual	2ED0	00 h	038F	00 h	INTEGER16
Flux Current Demand	2ED1	00 h	0388	00 h	INTEGER16
Flux Current Actual	2ED2	00 h	0390	00 h	INTEGER16
Motor Total Current Actual	2ED3	00 h	038B	00 h	INTEGER16

TX mapping examples:

Name	Index	Description
TX0_INDEX	2EBA	Pressure actual
TX0_SUB_INDEX	00	
TX1_INDEX	2EB4	Speed actual
TX1_SUB_INDEX	00	

5.2 Acyclic Communication

In addition to the cyclic I/O data exchange PROFINET allows also for acyclic communication from master to slaves using Record-Data Read and Write services that act like a mailbox system.

There are several Data Records pre-defined by PROFINET standard and used for device configuration and management.

Atos drive support an additional custom Data Record for asyncronous accessing to all the converter parameters and run-time data: this is Data Record with index 502 (0x1F6), see 5.2.1.

Using this Data Record, the PROFINET master can access all the data of the D-MP drive that is accessible from Modbus (exactly as with the S-SW-SETUP software) and in particular parameters that cannot be mapped for cyclic data exchange.

The data exchanged must be encoded according to the Modbus protocol and therefore consist of:

Node address + PDU

Node address is the same as used for RS485 communication (default = 1)

The PDU (Process Data Unit) consists of the Modbus Function Code and the related Data.

Function code:

Code	Function	Description
0x03	Read Holding Registers	Read 16 Bit single or multiple register contents
0x06	Write Single Register	Write single 16 Bit register
0x10	Write Multiple Register	Write multiple 16 Bit registers

Acyclic communication example

Parameter reading: **Device.DevLocal index 0x012F (303 dec)**

For more information about the Modbus parameter addresses required to configure the D-MP drive, see 6.

The PROFINET master writes the following 6 bytes to the data record with index 502 (0x1F6):

REQUEST		
Byte	Field Name	Value (hex)
01	Node	**0x01**
02	Function Code	**0x03**
03	Address HI	**0x01**
04	Address LO	**0x2F**
05	Number of Register HI	**0x00**
06	Number of Register LO	**0x01**

The Modbus request to node 0x01 is a Read Holding Registers (0x03) starting from register address 0x012F and involving a single register 0x0001.

The response returned by the D-MP drive in the Data Record with index 502 (0x1F6) will consist of the following 5 bytes:

RESPONSE		
Byte	Field Name	Value (hex)
01	Node	**0x01**
02	Function Code	**0x03**
03	Number of BYTE **(8 bit !)**	**0x02**
04	Parameter value HI	**0x00**
05	Parameter value LO	**0x01**

The response comes from node 0x01 for a Read Hold Register (0x03) request with a total data length of 0x02 bytes (1 16-bit register) whose value is 0x0001.

So the **Device.DevLocal index 0x012F (303 dec)** has a value of 1 (Reference via FieldBus).

5.2.1 Main Parameter list

For more information on how to view all parameter addresses, see 6.

Flow Conditioning

Parameter	Data Type	Profinet Acyclic access Parameter	Modbus Index for Profinet R/W	Scaling
Device.VpocUpperLimitA1	INTEGER 32	0x01F6	0x2060	RAQ_mrpm (e.g. 300rpm --> 300000)
Device.VpocLowerLimitA1	INTEGER 32	0x01F6	0x2062	RAQ_mrpm (e.g. 300rpm --> 300000)
Device.VpocDmdValGenRampTypeA1	INTEGER16	0x01F6	0x2155	0x00 NoRamp 0x01 SingleRamp 0x02 DoubleRamp
Device.VpocDmdValGenRampAccTimePositiveA1	UNSIGNED16	0x01F6	0x2156	UINT16_msec_to_sec (e.g. 400mSec --> 400)
Device.VpocDmdValGenRampDecTimePositiveA1	UNSIGNED16	0x01F6	0x2157	UINT16_msec_to_sec (e.g. 400mSec --> 400)

Pressure Conditioning

Parameter	Data Type	Profinet Acyclic access Parameter	Modbus Index for Profinet R/W	Scaling
Device.ManufActualValConditioning_Unit_4	INTEGER16	0x01F6	0x0140	0x00 Voltage 0x01 Current
Device.ManufActualValConditioning_MinInterface_4	INTEGER 32	0x01F6	0x2044	INT32_to_V_mA
Device.ManufActualValConditioning_RawLowLimit_4	INTEGER 32	0x01F6	0x202A	INT32_to_V_mA
Device.ManufActualValConditioning_MaxInterface_4	INTEGER 32	0x01F6	0x2046	INT32_to_V_mA
Device.ManufActualValConditioning_RawHighLimit_4	INTEGER 32	0x01F6	0x202C	INT32_to_V_mA
Device.ManufActualValConditioning_MinReference_4	INTEGER 32	0x01F6	0x2048	VALUE_to_Phisical
Device.ManufActualValConditioning_MaxReference_4	INTEGER 32	0x01F6	0x204A	VALUE_to_Phisical (e.g. 400Bar --> 400000)
Device.ManufActualValConditioning_T1_4	UNSIGNED32	0x01F6	0x2014	RAW_µs
Device.VprcSetPointCondUpperLimit32bitA1	INTEGER 32	0x01F6	0x205C	RAW_mbar (e.g. 350Bar --> 350000)
Device.VprcSetPointCondLowerLimit32bitA1	INTEGER 32	0x01F6	0x205E	RAW_mbar
Device.VprcDmdValGenRampTypeA1	INTEGER16	0x01F6	0x214D	0x00 NoRamp 0x01 SingleRamp 0x02 DoubleRamp
Device.VprcDmdValGenRampAccTimePositiveA1	UNSIGNED16	0x01F6	0x214E	UINT16_msec_to_sec (e.g. 400mSec --> 400)
Device.VprcDmdValGenRampDecTimePositiveA1	UNSIGNED16	0x01F6	0x214F	UINT16_msec_to_sec (e.g. 400mSec --> 400)

Pressure PID

Parameter	Data Type	Profinet Acyclic access Parameter	Modbus Index for Profinet R/W	Scaling
Device.VprcPress1CtrlFuncPropAction	UNSIGNED16	0x01F6	0x2134	PercInt16 (16384-->100%)
Device.VprcPointX1PropPress1	INTEGER16	0x01F6	0x2137	PercInt16 (16384-->100%)
Device.VprcPointX2PropPress1	INTEGER16	0x01F6	0x2138	PercInt16 (16384-->100%)
Device.VprcPointX3PropPress1	INTEGER16	0x01F6	0x2139	PercInt16 (16384-->100%)
Device.VprcPointX4PropPress1	INTEGER16	0x01F6	0x213A	PercInt16 (16384-->100%)
Device.VprcPointY1PropPress1	INTEGER16	0x01F6	0x213B	PercInt16 (16384-->100%)
Device.VprcPointY2PropPress1	INTEGER16	0x01F6	0x213C	PercInt16 (16384-->100%)
Device.VprcPointY3PropPress1	INTEGER16	0x01F6	0x213D	PercInt16 (16384-->100%)
Device.VprcPointY4PropPress1	INTEGER16	0x01F6	0x213E	PercInt16 (16384-->100%)
Device.VprcPositivePropPress1	INTEGER32	0x01F6	0x200A	32BitScaling1 (NUM / DEN) 2147450879 --> 1,00
Device.VprcNegativePropPress1	INTEGER32	0x01F6	0x200C	32BitScaling1 (NUM / DEN) 2147450879 --> 1,00
Device.VprcPress1CtrlFuncIntegralAction	INTEGER16	0x01F6	0x2142	PercInt16 (16384-->100%)
Device.VprcIntegralTimeP1	INTEGER16	0x01F6	0x2141	decmsec_to_µsec (5000 -->500000)
Device.VprcIntegralAWP1	INTEGER16	0x01F6	0x2143	0x00 Disabled 0x01 Enabled
Device.VprcIntegralThresP1	INTEGER16	0x01F6	0x2144	PercInt16 (16384-->100%)
Device.VprcAntiWUTimeP1	UNSIGNED16	0x01F6	0x2145	decmsec_to_µsec (5000 -->500000)

Aternated p/Q control

Parameter	Data Type	Profinet Acyclic access Parameter	Modbus Index for Profinet R/W	Scaling
Device.VprcPQ_QtoP_PressForceTh_P1	UNSIGNED16	0x01F6	0x215E	PercInt16 (16384-->100%)
Device.VprcPQ_PtoQ_PressForceTh_P1	UNSIGNED16	0x01F6	0x215F	PercInt16 (16384-->100%)

Sources

Parameter	Data Type	Profinet Acyclic access Parameter	Modbus Index for Profinet R/W	Scaling
Device.DevLocal	INTEGER16	0x01F6	0x0137	0x00 Fieldbus 0x01 Local
Device.DevMode	INTEGER16	0x01F6	0x012F	0x01 Fieldbus 0x02 Analog 0x03 Internal
Device.DevModePress	INTEGER16	0x01F6	0x012E	0x01 Fieldbus 0x02 Analog 0x03 Internal

Responses to exceptions

If the slave device receives the request correctly, it returns a normal response.

If the slave receives the request without a communication error, but is unable to handle it (for example, if the request is to read a non-existent parameter), the slave returns an exception response that informs the master of the nature of the error.

The exception response message has two fields that differentiate it from a normal response:

Function Code Field

In a normal response, the slave echoes the function code from the original request into the function code field of the response. All function codes have a most significant bit (MSB) of 0 (their values are all less than 80 hex). In an exception response, the slave sets the MSB of the function code to 1. This causes the value of the function code in an exception response to be exactly 80 hex higher than the value a normal response would have.

With the MSB of the function code set, the master application program can recognize the exception response and examine the data field for the exception code.

Data field:

In a normal response, the slave can return data or statistics in the data field (any information requested in the request). In an exception response, the slave returns an exception code in the data field. This defines the slave condition that caused the exception.

Example:

Master request and Slave exception response to a non-existent parameter

REQUEST		
Byte	Field Name	Value (hex)
01	Node	**0x01**
02	Function Code	**0x03**
03	Address HI	**0x99**
04	Address LO	**0x99**
05	Number of Register HI	**0x00**
06	Number of Register LO	**0x01**

RESPONSE		
Byte	Field Name	Value (hex)
01	Node	**0x01**
02	Function Code	**0x83**
03	Exception Code	**0x02**

In this example:

- the master sends a read request to the slave, function code **03**
- the requested parameter address is **0x9999 (hex)**
- the slave returns an error response with the indicated Exception Code (**02**)
- this indicates a non-existent parameter address for the slave

Exception Code List

0x01 WRONG FUNCTION CODE

The received function code is not an allowed action for the slave. This may be because the requested function code is invalid. It could also indicate that the slave is in the wrong state to process such a request, e.g. because it is not configured and is being asked to return register values.

0x02 WRONG ADDRESS

The received parameter address is not an allowed slave address.

0x03 WRONG VALUE IN DATA FIELD

A value in the data field (in writing) is not an allowed value for the slave.

0x04 SLAVE DEVICE FAULT

An unrecoverable error occurred while the slave was attempting to perform the requested action.

5.2.2 Alarms

Alarms can be acquired through cyclic data exchange and/or through Acyclic communication.

Cyclic communication

- **Running Alarms Parameter**
  To receive runtime alarms using the cyclic channel, it is possible to map in the PLC, the RUNNING_ERROR parameter (present in the list of GSDML mappable variables)
- **Bit Warning on the Status Word.**
  With this bit present on the status word, it is possible to be informed if there is an alarm configured with reaction >= Warning
  This bit can be used as a cyclic trigger for reading the Acyclic Alarm words

Acyclic communication

- **Running Alarms**
  The running alarms parameter can also be read through the acyclic channel [Modbus 0x03D4]
- **Alarms Status Parameter 1 e 2**
  No. 2 parameters at 32-bit (binary coded) that contain the status of each individual alarm

Running Alarm parameter

Parameter	Index				Type
Running Alarm	0x03D4				**UINT32**

Byte 3	Byte 2	Byte 1	Byte 0
Number of active alarms	Code list (see Code Error List)		

If more than 1 alarm is present, this parameter contains the most critical active alarm.

Code Error List:

Value dec	Value Hex	Description
0	00000000	No alarm
21041	00005231	AT0 Flow Analog Setpoint Out of Limits
21042	00005232	AT1 Pressure Analog Setpoint Out of Limits
21044	00005234	AT2 Pressure Transducer Out of Limits
33538	00008302	AT3 Pressure Control Error
33537	00008301	AT4 Flow Control Error
127120	0001F090	AT5 Pressure Setpoint Limits Touched
192656	0002F090	AT6 Flow Setpoint Limits Touched
65348	0000FF44	AT7 Autotuning Procedure Not Completed
61585	0000F091	AT8 Motor Check Procedure Not Performed
61587	0000F093	AT10 Emergency Parameters Loaded
28960	00007120	A0.0 Motor Anisotropy too Low
1070384	00105530	A1.0 Default Parameters loaded
1135920	00115530	A1.1 EEPROM Reading Failed
1201456	00125530	A1.2 EEPROM Writing Failed
1266992	00135530	A1.3 EEPROM Reading and Writing Failed
1332528	00145530	A1.4 Data Store Incomplete
2126592	00207300	A2.0 Sensor Channels Mismatch During Connection Check
2192128	00217300	A2.1 Sensor Channels Values too Low
3167232	00305400	A3.0 Drive Output Current Value too High
5259264	00504000	A5.0 Motor Temperature too High
5324800	00514000	A5.1 Radiator Temperature too High
5390336	00524000	A5.2 Braking Resistor Adiabatic Energy too High
5455872	00534000	A5.3 Power Dissipation on Braking Resistor too High
5521408	00544000	A5.4 Motor Thermal Probe Not Connected
5521408	00544000	A5.5 System Running with Radiator Temperature too High
6308352	00604200	A6.0 I2T Motor Thermal Model Current too High
7375104	00708900	A7.0 Motor Autotuning Test Not Completed
7440640	00718900	A7.1 Speed Error During Motor Autotuning Test
8425472	00809000	A8.0 External Enable Configured on Digital Input - Not Active
9466640	00907310	A9.0 Firmware and Resolver Card are Not Compatible
9532176	00917310	A9.1 Resolver Cable Not Connected
9597712	00927310	A9.2 Motor Speed Reading too High
9663248	00937310	A9.3 Zero Top Speed of Sensor Not Found
9794320	00957310	A9.4 SinCos Incremental Channel Problems
9859856	00967310	A9.6 Speed Control Lost
10498560	00A03200	A10.0 DC Bus Voltage too Low
10564096	00A13200	A10.1 Emergency Breaking - Lost Power Supply
11612928	00B13300	A11.1 DC Bus Voltage too High - Hardware Detection
11678464	00B23300	A11.2 DC Bus Voltage too High - Software Detection
11744000	00B33300	A11.3 DC Bus Voltage too High – Hardware/Software Detection
12607744	00C06100	A12.0 Software Drive Enabled
12673280	00C16100	A12.1 Run without Soft Start
13660416	00D07100	A13.0 DC Bus Charging Time Too Low
13725952	00D17100	A13.1 STO Function Enable

13791488	00D27100	A13.2 DC Bus Ripple too High
13857024	00D37100	A13.3 Only One STO Channel Active
13922560	00D47100	A13.4 Fault on at Least One STO Channel
13988096	00D57100	A13.5 Internal Communication STO Channel Fault
14053632	00D67100	A13.6 STO Diagnostic Test Required
14692656	00E03130	A14.0 Motor Phase Reversed
14758192	00E13130	A14.1 Motor Power Cable Not Properly Connected
15754016	00F06320	A15.0 Motor Poles Number Wrong
15950624	00F36320	A15.3 Incorrect Settings of Speed Sensor
16016160	00F46320	A15.4 Sensor Test Failed During Connection Check

Alarm Status Parameter 1 [Err 0- 31]

Parameter	Index	Type
Detailed Error 0-31	0x03E2	UINT32

Bit	Error
00	Flow Analog Setpoint Out of Limits
01	Pressure Analog Setpoint Out of Limits
02	Pressure Transducer Out of Limits
03	Pressure Control Error
04	Flow Control Error
05	Pressure Setpoint Limits Touched
06	Flow Setpoint Limits Touched
07	Autotuning Procedure Not Performed
08	*- reserved*
09	Motor Check Procedure Not Performed
10	*- reserved*
…	…
…	…
31	*- reserved*

Alarm Status Parameter 2 [Err 32 - 63]

Parameter	Index	Type
Detailed Error 32-63	0x03E4	UINT32

Bit	Error
00	Motor Anisotropy too Low
01	EEPROM Alarms
02	Absolute Sensor Channels Alarms
03	Drive Output Current Value too High
04	*- reserved*
05	Temperature Alarms
06	I2T Motor Thermal Model Current too High
07	Motor Autotuning Test Not Completed
08	External Enable Configured on Digital Input - Not Active
09	Speed Sensor Alarms
10	DC Bus Voltage too Low Alarms
11	DC Bus Voltage too High Alarms
12	Soft Start Alarms
13	STO and DC Bus Charging/Ripple Alarms
14	Motor Power Connection Alarms
15	Sensor Test Alarms
16	*- reserved*
…	…
…	…
31	*- reserved*

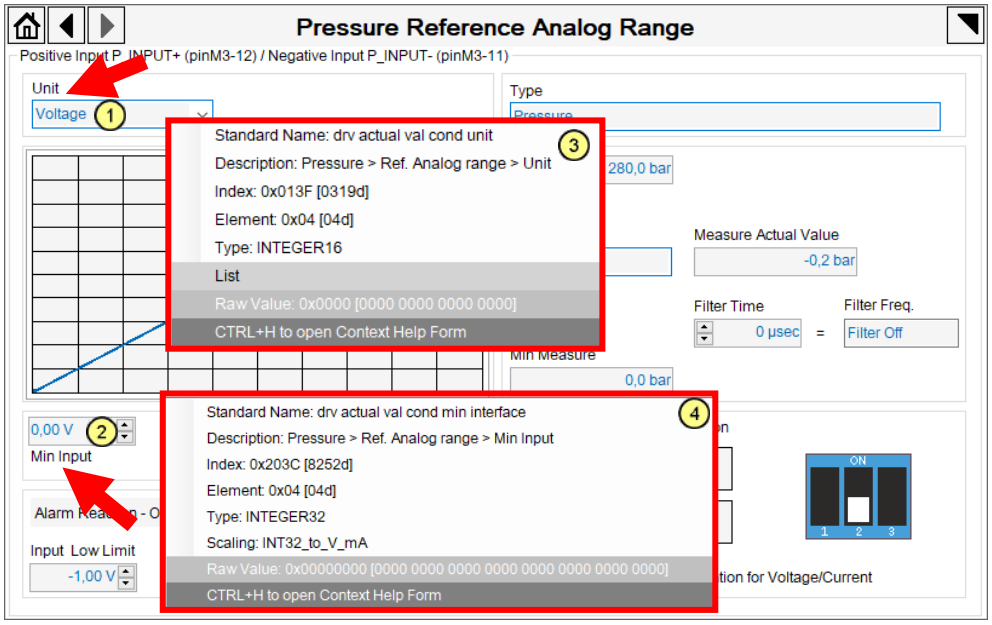
6 OBJECT DICTIONARY

6.1 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:

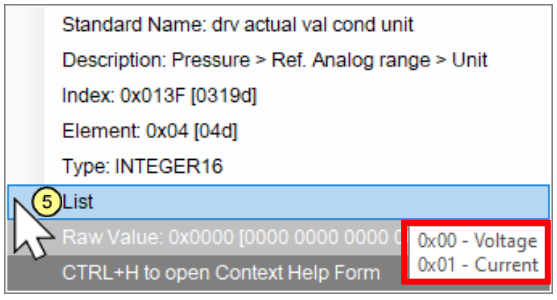
Click right button of the mouse on **Unit** control (1) or **Min Input** control (2) to open the related fieldbus communication windows ((3) and (4)).



Windows (3) 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:	Parameter address
Element:	
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".

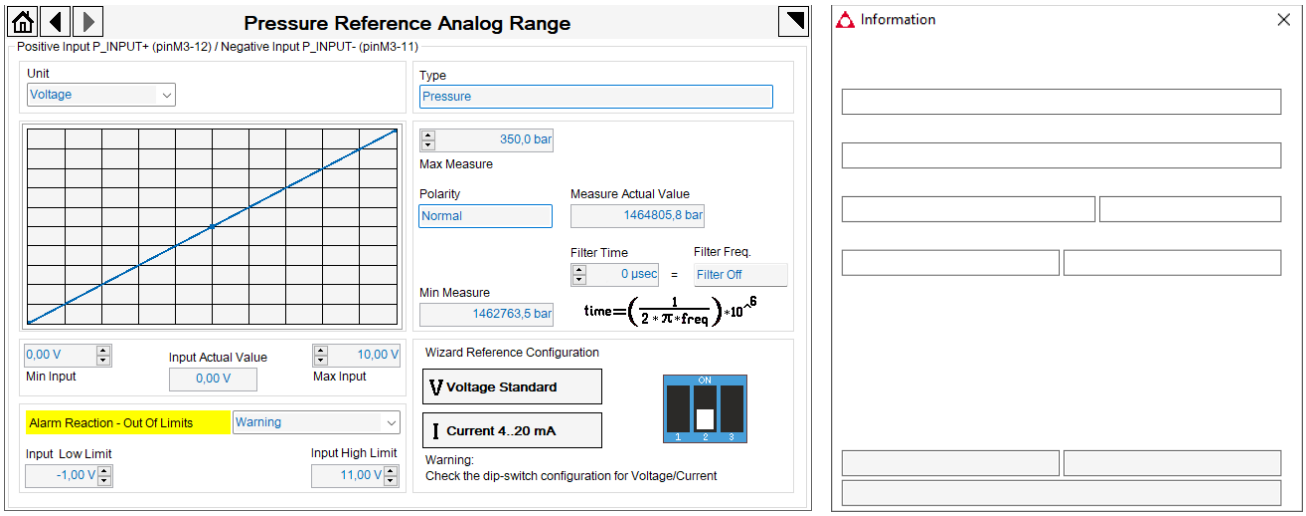
By pressing CRTL+H on PC keybord the Context Help Form windows opens.

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.

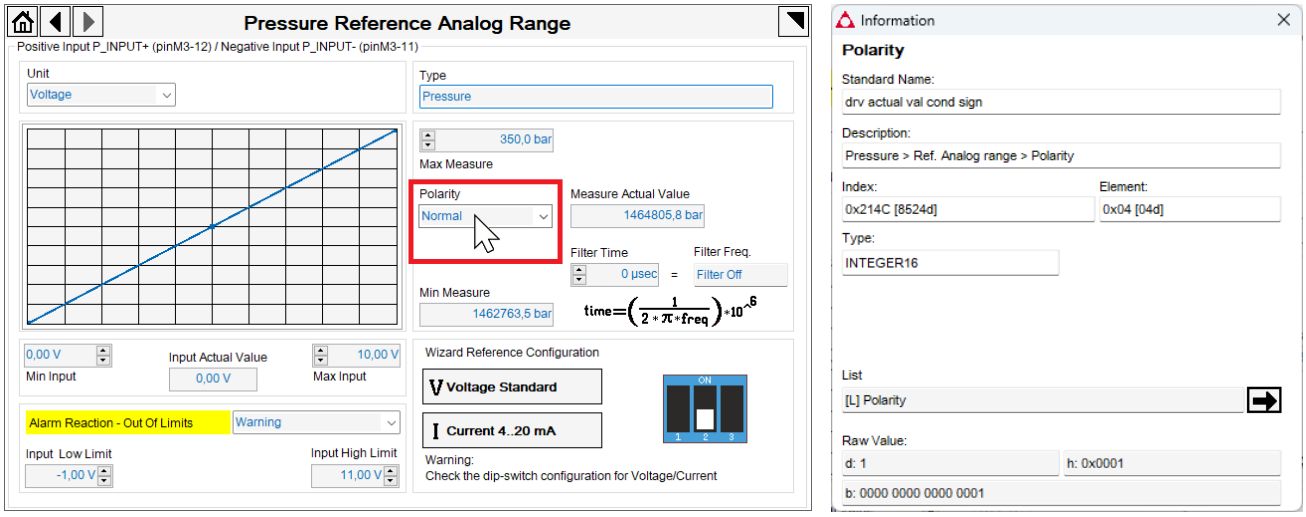
If no parameter is selected the fields of the window will be appears empty.

Example: no parameter selected



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

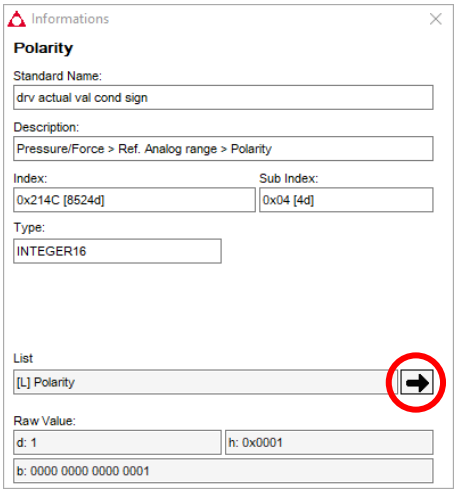
Example: “Polarity” parameter selected



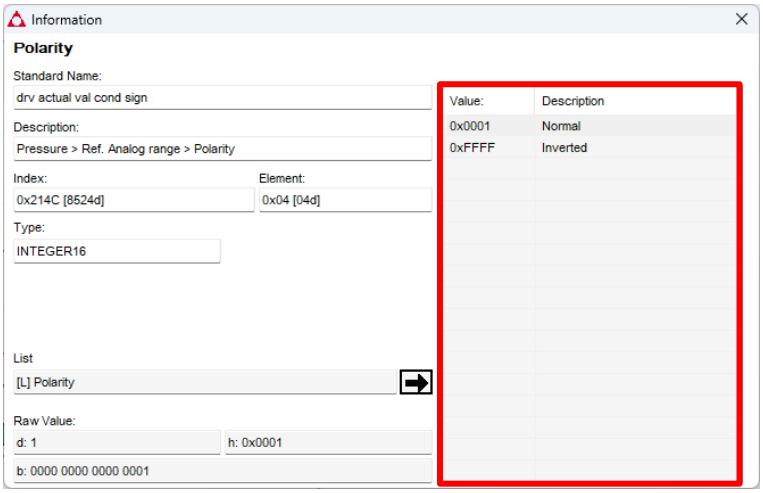
When in a parameter is present a “List” [L] press the black arrow to display the related information of the list (Value and Description).

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	**Element**	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**	**Smart Cooling 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**	**P/Q**	**Warning**	**Status**

Status

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

Status value (bits 6-0)	
Not reready 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

P/Q

Bit 8 indicates if the pressure control is active or not active:

P/Q (bit 8)	
Not 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

Smart Cooling active

Bit 24 indicates if the control (Smart Cooling) is active or not active:

Smart Cooling active (bit 24)	
Not active	0
Active	1

STO active

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

STO active (bit 24)	
Not active	0
Active	1

STO corrupted

Bit 26 indicates if the STO function is corrupted or not corrupted:

STO corrupted (bit 26)	
Not corrupted	0
Corrupted	1

STO test suggested

Bit 27 indicates if the STO test is suggested or not suggested:

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

8.2 Control Word - 32bit

Index	2EA0h	**Element**	00h

Conditions:

MSB *LSB*

Bit	**31-17**	**16**	**15-14**	**13-12**	**11**	**10-8**	**7-0**
Content	Res	**Alarm Reset**	**Smart Selection**	**Pressure PID Selection**	**Enable P/Q Control**	Res	**Control**

Control

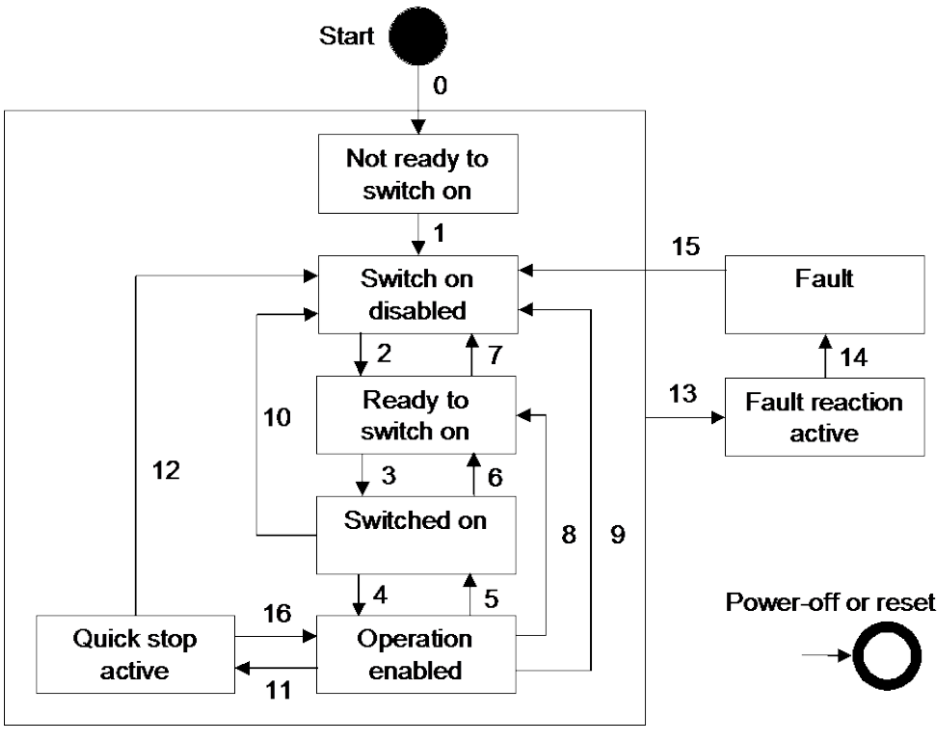
The first eight bits 7-0 allow to request the transition of the drive status to a defined condition:

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

(**) Automatic transition to switch ON disable after the quick stop ramp is elapsed

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



Enable P/Q Control

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

Enable P/Q Control (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:

Smart Tuning (bits 15-14)	
Dynamic	00
Balanced	01
Smooth	10

Alarm Reset

Bits 16 allows to clear all alarms present into the drive:

Alarm Reset (bits 16)	
Alarm reset	

9 CONFIGURATION FILE (GSDML)

The characteristics of an IO Device are described by the device manufacturer in a General Station Description (GSD) file. The language used for this purpose is the GSDML (GSD Markup Language) - an XML based language. For I/O data, the GSDML file describes the structure of the cyclic input and output data transferred between the Programmable Controller and the PROFINET IO Device. Any mismatch between the size or structure of the input and output data and the actual internal device structure generates an alarm to the controller.

GSDML files are included in MyAtos.


atos
the smart electrohydraulics