Bulletin MSG11-5715-708/UK

Installation Manual

Supplement for Valves with EtherCAT Interface

Parker Hannifin
Manufacturing Germany GmbH & Co. KG
Industrial Systems Division Europe
Gutenbergstr. 38
41564 Kaarst, Germany
E-mail: isde.kaarst.support@support.parker.com

WARNING — USER RESPONSIBILITY

FAILURE OR IMPROPER SELECTION OR IMPROPER USE OF THE PRODUCTS DESCRIBED HEREIN OR RELATED ITEMS CAN CAUSE DEATH, PERSONAL INJURY AND PROPERTY DAMAGE.

This document and other information from Parker-Hannifin Corporation, its subsidiaries and authorized distributors provide product or system options for further investigation by users having technical expertise.

The user, through its own analysis and testing, is solely responsible for making the final selection of the system and components and assuring that all performance, endurance, maintenance, safety and warning requirements of the application are met. The user must analyze all aspects of the application, follow applicable industry standards, and follow the information concerning the product in the current product catalog and in any other materials provided from Parker or its subsidiaries or authorized distributors.

To the extent that Parker or its subsidiaries or authorized distributors provide component or system options based upon data or specifications provided by the user, the user is responsible for determining that such data and specifications are suitable and sufficient for all applications and reasonably foreseeable uses of the components or systems.

1 General information

This document is a supplement to the general valve manuals. Its sole purpose is to describe the optional EtherCAT® field bus interface.

It is assumed the reader is already familiar with the EtherCAT® field bus. Vast information about EtherCAT® and the CANopen Protocol is available at the below references and in the documents listed below. All parameters follow the common EtherCAT® communications profile IEC 61158-x-12, CiA 301 (CANopen Application layer and communication profile) and CiA 408 (CANopen Device profile fluid power technology proportional valves and hydrostatic transmissions).

1.1 Further documentation for the digital valves

This manual is part of the complete documentation for digital valves existing of the following documents:

- ProPxD parameter description
- User manual valve

Software:

- ProPXD
- ParkerDFPEtherCAT.xml

Visit http://www.parker.com/euro_hcd to download.

1.2 References

Below organisations provide information about standards for EtherCAT®, CANopen and the device profile used.

ETG	EtherCAT Technology Group Ostendstrasse 196, DE-90482 Nuernberg, Germany http://ww.ethercat.org
ISO	International Organization for Standardization 1, ch. De la Voie-Creuse, Case postale 56, CH-1211 Geneva 20 http://www.iso.org
IEC	International Engineering Consortium 233 S. Wacker Drive, Suite 8400, Chicago, IL 60606-6338 USA http://www.iec.org
CiA	CAN in Automation Kontumazgarten 3, DE-90429 Nuernberg, Germany http://www.can-cia.org
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V. Lyoner Strasse 18, 60528 Frankfurt/Main, Germany http://www.vdma.org

Documents of importance are listed below.

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

1.2.1 EtherCAT® Feldbus

The EtherCAT® field bus interface provides a 100 Base full duplex real time Ethernet connection to the digital valves using standard EtherCAT® frames according to IEEE 802.3.

1.2.2 Object dictionary (CANopen over EtherCAT®)

CiA 301	CANopen application layer and communication profile
CiA 303-1	Additional specification

1.2.3 Device profile

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

1.2.4 Initiator of EtherCAT®

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany. EtherCAT® was developed in 2003 by Beckhoff Automation GmbH and propagated as an open standard. To further develop the technology, the user association “EtherCAT® Technology Group” (ETG) was established.

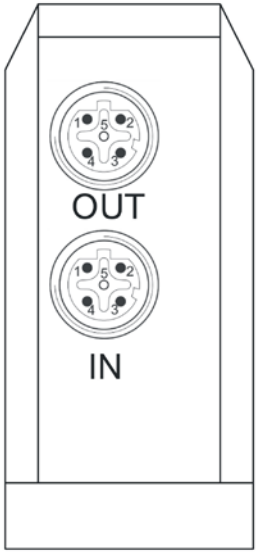
Beckhoff’s product TwinCAT® was used to verify the functionality of the EtherCAT® interface.

2. Scope EtherCAT® interface

- Diagnostics (actual value feedback, temperature, fault, voltage, time of operation)
- EtherCAT®-status LEDs (Link/Active)
- Valve status LED
- Reduced Device State Machine (DSM) as per device profile CiA DSP-408
- Bus-cycle time down to 0.250 ms
- 2 x EtherCAT® – Connector M12x1 / 4-pin, d-coded (female)
- Maximum operating temperature 85 °C

3 Ethernet connection

The EtherCAT® vales are equipped with two M12x1 socket 4-pin (female), d-coded connectors, one EtherCAT® IN and one EtherCAT® OUT to implement the common bus topologies. Cables in various lengths are available from Phoenix contact, Binder, etc.



Location depends on case (Figure DFP).

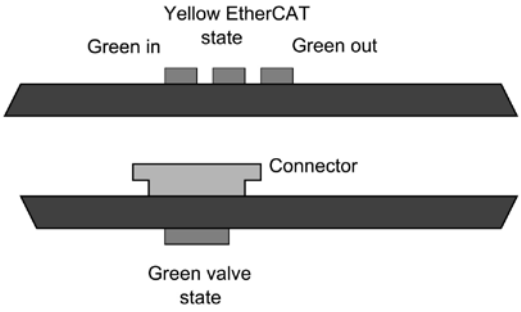
Connector (M12, 4pole, d-coded)

Signal	Comment	RJ45	M12	Wire colour TIA-568B	Wire colour EN61918
TD+	Transmit data +	1	1	WHS /OG	YE
TD-	Transmit data -	2	3	OG	OG
RD+	Receive data +	3	2	WH /GN	WH
RD-	Receive data -	6	4	GN	BU
-	3. pair +	4	-	BU	-
-	3. pair -	5	-	WH /BU	-
-	4. pair +	7	-	WH /BN	-
-	4. pair -	8	-	BN	-
Shield	Screening	Housing	Housing	n.def.	n.def.

WH=white, OG=orange, GN=green, BU=blue, BN=brown, YE=yellow

4 Function indication LEDs

The LEDs are located around the RS232 connector behind the screw cap at the top of the valve box (DFplus). Directly at the connector is the operation mode LED of the valve. On another PCB there are 3 LEDs for EtherCAT® monitoring.



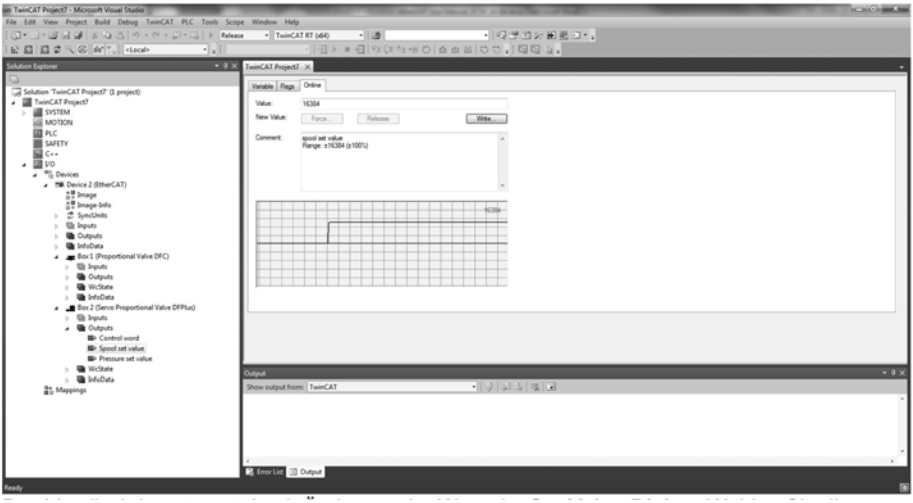
Green valve LED	
Slow blinking (1000 ms)	Enabled
Faster blinking (500 ms)	Disabled
Very fast blinking (50 ms)	Fault
Yellow (or green EtherCAT® state machine LED	
Off	Initialization or no power
Fast blinking (500 ms)	Pre-operational
Slow blinking (1000 ms)	Safe-operational
On	Operational
Grün EtherCAT® in Port LED und Grün EtherCAT® out Port LED	
Off	No connection
On	Connection but no data traffic
Fast blinking	Data traffic

5 Adding the device in TwinCAT

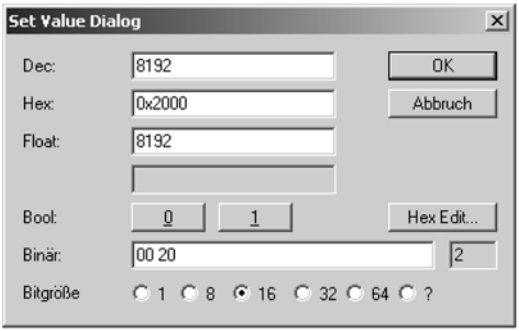
Any EtherCAT® master should be able to communicate with the valve. The configuration is described for the TwinCAT® System Manager from Beckhoff Automation GmbH. For offline configuration (application design), the slave configuration file ParkerHCDEtherCATValves.xml should be available in the directory C:\TwinCAT\Io\EtherCAT (TwinCAT® 2) or C:\TwinCAT\3.1\Config\Io\EtherCAT (TwinCAT® 3).

Online the devices can be scanned with the **Scan Devices…** menu command. The device has to be connected to the master. Select **I/O Devices**. Now press **F5** or the **magic wand** button or the right mouse button and from the drop down **Scan Devices**…

Follow the instructions of TwinCAT® and when the devices are successfully configured, one should see something like the below:

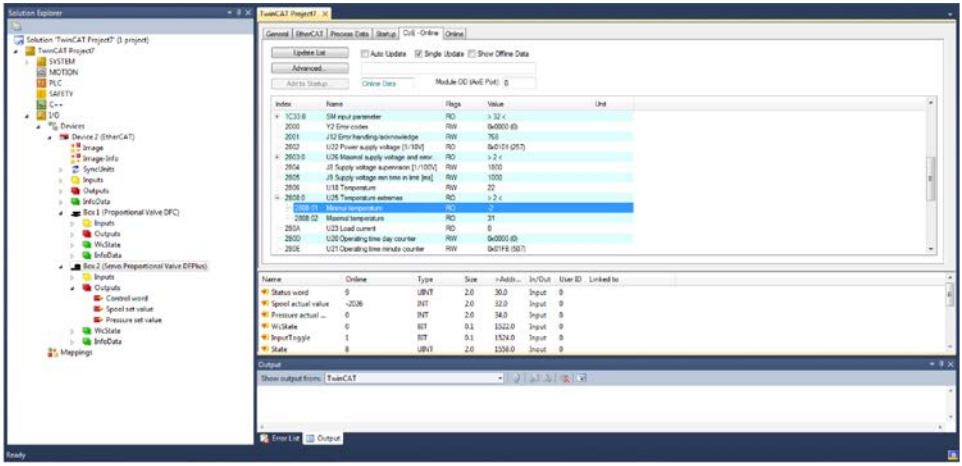


The valve can be operated by changing the values with the **Set Value Dialog**. Select tab **Online**, button **Write…**



These are the I/O's to be connected to your control software. On how to accomplish this, see the Twin-CAT® documentation or the documentation for your EtherCAT® master. The I/O's are described by the objects 6040_h , 6041_h , 6300_h , 6301_h , 6380_h and 6381_h , see their descriptions further on.

The device parameters are changed in the **Set Value Dialog** of the **CoE-Online** list. See below.



6 Watchdogs

The EtherCAT® Slave Controllers (ESCs) support up to two internal watchdogs (WD), a Process Data watchdog used for monitoring process data accesses, and a Process Data Interface (PDI) watchdog monitoring Process Data Interface activity. They can be useful when implementing data intensive applications.

The timeout for both watchdogs can be configured individually, but they share a single Watchdog Divider (WD_DIV), register 0400_h :0401_h). The watchdog timeout is calculated from the Watchdog Divider settings multiplied with the Watchdog Time settings for PDI (WD_PDI, register 0410_h :0411_h) or Process Data (WD_PD, register 0420_h :0421_h). Base time unit is 40 ns. The watchdog timeout jitters, the jitter depends on the Watchdog Divider settings. I.e., selecting smaller Watchdog Divider settings results in smaller jitter.

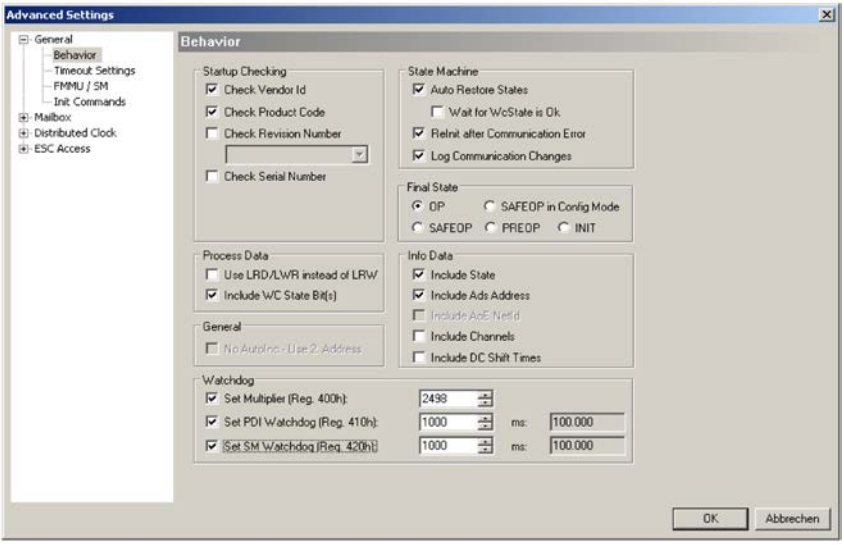
The following equations are used for a quick estimation of the watchdog timeout (they are not exact in terms of nanoseconds):

$$t_{WD\_Div} = (WD\_DIV + 2) * 40\ ns$$
$$t_{WD\_PDI} = [t\ WD\_Div\ * WD\_PDI\ ;\ t\ WD\_Div\ * WD\_PDI + t\ WD\_Div\ ]$$
$$t_{WD\_PD} = [t\ WD\_Div\ * WD\_PD\ ;\ t\ WD\_Div\ * WD\_PD + t\ WD\_Div\ ]$$

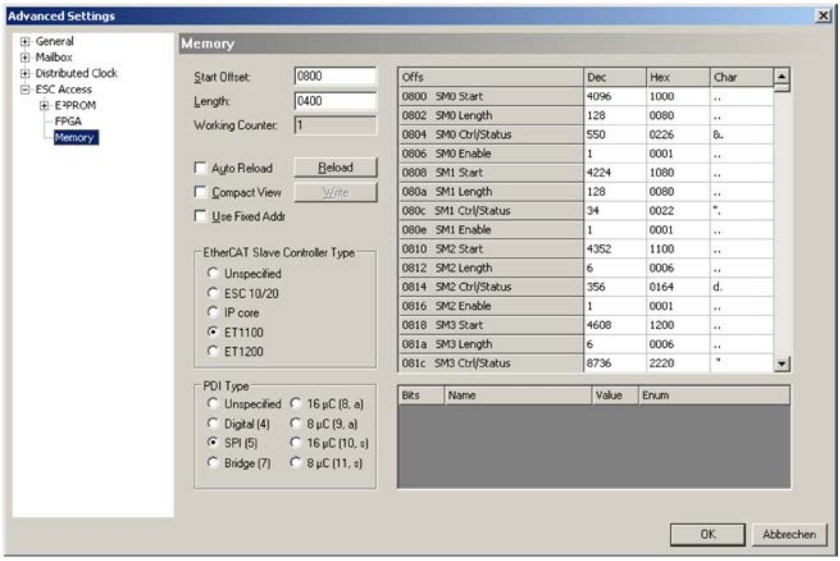
Registers used for watchdogs are described below.

Register address	Name	Description
0110_h .1	ESC DL status	Status PDI watchdog
0400_h :0401_h	Watchdog divider	Watchdog divider (WD_DIV)
0410_h :0411_h	Watchdog time PDI W	atchdog time PDI (WD_PDI)
0420_h :0421_h	Watchdog time process data	Watchdog time process data (WD_PD)
0440_h :0441_h	Watchdog status process data	Status process data watchdog
0442_h	Watchdog counter process data	Watchdog expiration counter process data
0443_h	Watchdog counter PDI	Watchdog expiration counter PDI

Timing for the watchdogs can be set in TwinCAT® as below. First select **Box1 (Servo Valve DFplus)**, tab **EtherCAT®**, button **Advanced Settings…**



The SyncManagers and other registers can be accessed in TwinCAT as below.



6.1 Process Data Watchdog

The Process Data Watchdog is rewound (triggered) by a write access to a SyncManager buffer area, if the SyncManager is configured to generate a watchdog trigger signal (SyncManager Control register 0804_h .6 for SyncManager 0, etc.). The watchdog trigger signal is generated after the buffer was completely and successfully written (similar to the Interrupt Write of a SyncManager).

The Process Data watchdog can be disabled by setting the Process Data Watchdog Time to 0.

A timeout of the Process Data watchdog has the following consequences:

- Watchdog Status register for Process Data (0440_h .0) reflects the watchdog status.
- The device is disabled (could be useful to implement user defined behaviour here).
- The Watchdog Counter Process Data (0442_h) is incremented.
- A bus communication fault is generated.

6.2 PDI Watchdog

The PDI watchdog is rewound (triggered) by any correct read or write access by the Process Data Interface. The PDI watchdog can be disabled by setting the PDI Watchdog Time to 0.

A timeout of the PDI watchdog has the following consequences:

- ESC DL Status register (0110_h .1) reflects the watchdog status. This can be mapped to the ECAT interrupt to inform the EtherCAT® device.
- The Watchdog Counter PDI (0443_h) is incremented.

7 CAN over EtherCAT® object dictionary

The below is extensively described by the CAN in Automation (CiA®) e.V. in their documents CiA Draft Standard Proposal 301 (CANopen Application layer and communication profile) and CiA Draft Standard 408 (CANopen Device profile fluid power technology proportional valves and hydrostatic transmissions. Further the valve is equipped with an RS232 interface. Several parameters of this interface are mapped on EtherCAT® objects. For example U22, power supply voltage [1/10V] is mapped to object 2802_h. U22 indicates that it is originally an RS232 parameter. See the ProPxD documentation for these RS232 parameters.

8 Synchronisation manager (SM)

Several synchronisation managers allow consistent and secure data exchange between the EtherCAT® master and the slave device. The SMs inform both sides about data exchange status.

The SM settings in the object dictionary have read only access. These settings are read by the master at start up and are needed to configure the process data and mailbox communication.

8.1 Object $1C00_h$: SM communication type

Determines the direction and type of the communication for the specified SM channel.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
$1C00_h$	0	SM communication type Nr. of channels	UINT8	ro	-	0...4	4
$1C00_h$	1	SM0 communication type	UINT8	ro	-	UINT8	01_h (mailbox receive)
$1C00_h$	2	SM1 communication type	UINT8	ro	-	UINT8	02_h (mailbox send)
$1C00_h$	3	SM2 communication type	UINT8	ro	-	UINT8	03_h (process data receive)
$1C00_h$	4	SM3 communication type	UINT8	ro	-	UINT8	04_h (process data send)

8.2 Object $1C12_h$: SM2 receive PDO assignment

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
$1C12_h$	0	SM2 Nr. of assigned receive PDOs	UINT8	ro	-	0...4	1
$1C12_h$	1	SM2 receive PDO1 assignment	UINT16	ro	-	UINT16	1600_h

8.3 Object $1C13_h$: SM3 transmit PDO assignment

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
$1C13_h$	0	SM3 Nr. of assigned transmit PDOs	UINT8	ro	-	0...4	1
$1C13_h$	1	SM3 transmit PDO1 assignment	UINT16	ro	-	UINT16	$1A00_h$

9 Process data object (PDO) communication

These objects define the parameters communicated over the real time communication channels of SM2 and SM3. The communication channels exist of 3 buffers so writing and reading can take place simultaneous. Mailbox communication uses only one buffer so reading and writing takes place alternating.

9.1 Process data object mapping (PDO mapping)

The object 1600_h represents the RxPDO mapping and the object $1A00_h$ represents the TxPDO mapping. Sub-index 00_h contains the number of valid mapping entries within the mapping object. Sub-indeces 01_h to 08_h contain the references to the mapped application parameters. The application parameters are referenced by their index, sub-index and length. The length contains the length of the application parameter in bits. This may be used to verify the mapping.

Parameter addressing				
Byte	3	2	1	0
Description	Index MSB	Index LSB	Sub-index	Bit length: 10_h

The pointer contains a combination of index, sub-index and length of the parameter to be used. It may only refer to parameters with a bit length of 10_h.
Remapping is not implemented. The objects are configured to be read only.

9.2 Object 1600_h: RxPDO mapping (CiA 301/408)

The parameter no. of mapped application objects in PDO (1600_h) sets the number of real-time application parameters to be received. The application parameters are mapped by combining its index, sub-index and length to a 32 bit number which is written to one of the index positions within the PDO object.

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
1600_h	0	Nr. of mapped application objects in PDO	UINT8	ro	-	UINT8	03_h
1600_h	1	1st application object	UINT32	ro	-	UINT32	60400010 $_h$
1600_h	2	2nd application object	UINT32	ro	-	UINT32	63000110 $_h$
1600_h	3	3rd application object	UINT32	ro	-	UINT32	63800110 $_h$

The following three parameters from the object dictionary are mapped as process data parameters by default according to CiA 408:

Device state machine (DSM) Control Word (6040_h)
Spool position setpoint value (6300_h sub-index 01_h)
Pressure setpoint value (6380_h sub-index 01_h) (D*FP only)

9.3 Object $1A00_h$: TxPDO mapping (CiA 301/408)

The parameter Nr. of mapped application objects in PDO ($1A00_h$) sets the number of real-time application parameters to be transmitted. The application parameters are mapped by combining its index, sub-index and length to a 32 bit number which is written to one of the index positions within the PDO object.

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
$1A00_h$	0	Nr. of mapped application objects in PDO	UINT8	ro	-	UINT8	03_h
$1A00_h$	1	1st application object	UINT32	ro	-	UINT32	60410010_h
$1A00_h$	2	2nd application object	UINT32	ro	-	UINT32	63010110_h
$1A00_h$	3	3rd application object	UINT32	ro	-	UINT32	63810110_h

The following three parameters from the object dictionary are mapped as process data parameters by default according to CiA 408:

Device state machine (DSM) Status Word (6041_h)
Spool position actual value (6301_h sub-index 01_h)
Pressure actual value (6381_h sub-index 01_h) (D*FP only)

10 Device identification

10.1 Object 1000_h: device type (CiA 301)

The LS 16 bit word (bits 15...0) describes the device profile number that is being used. 408 is the value for CiA408, the profile for hydraulic devices. The MS 16 bit word for additional information is not used.

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
1000_h	0	Device type	UINT32	m ro	-	UINT32	408

10.2 Object 1008_h: M1 manufacturer device name (CiA 301)

ProPXD parameter number M1 Manufacturer short text.

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
1008_h	0	Manufacturer device name	STRING	ro	-	-	Name

10.3 Object 1009_h: Manufacturer Hardware Version (CiA 301)

Indicates the current hardware version of the device electronics.

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
1009_h	0	Manufacturer hardware version	STRING	ro	-	-	XXX-XXX

10.4 Object $100A_h$: manufacturer software version (CiA 301)

Indicates the current software version of the device.

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
$100A9_h$	0	Manufacturer software version	STRING	ro	-	-	XXX

10.5 Object 1018_h: identity object (CiA 301)

These parameters represent a unique identification for the device.

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
1018_h	0	Nr of	UINT8	m ro	-	UINT8	04_h
1018_h	1	Vendor-ID	UINT32	m ro	-	UINT32	8000089_h

The vendor ID assigns the device to a manufacturer and department. Assigned uniquely to manufacturers by CiA.

31	24	23	0
Department 08_h		Company 000089_h	
MSB			LSB

Department HCD = 08_h
Company Parker Hannifin = 000089_h
Further identification by the ProPXD parameters M2, M1 and M4 in objects 6052_h, 1008_h and 6053_h.

10.6 Object 6052_h: Device serial number (CiA 408)

The unique serial number of the device.

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
6052_h	0	Device serial number	STRING	ro	-	-	XXXXXXXXXX

10.7 Object 6053_h: M4 device description (CiA 408)

ProPXD parameter number M4 date code.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
6053_h	0	Device description	STRING	ro	-	-	XXXXXXXXXX

10.8 Object 6055_h: Device model URL (CiA 408)

The internet address where to find additional information about the device.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
6055_h	0	Device model URL	STRING	ro	-	-	www.parker.com

10.9 Object 6057_h: Device vendor name (CiA 408)

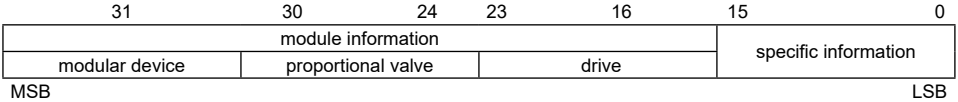
Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
6057_h	0	Device vendor name	STRING	ro	-	-	Parker Hannifin, Gutenbergstrasse 38, D-41564 Kaarst, Germany

10.10 Object $605F_h$: Device capability (CiA 408)

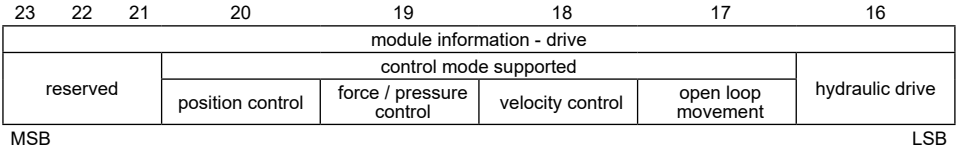
Contains information on the capabilities of a device.

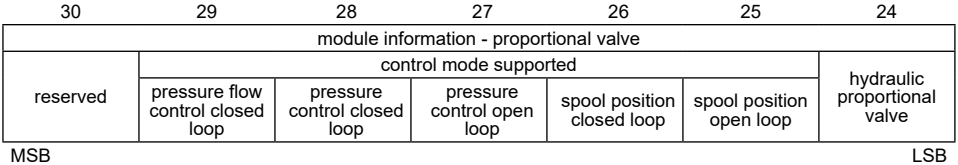
Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
$605F_h$	0	Device capability	UINT32	m ro	-	UINT32	15000000_h

Value description:



Module Information:





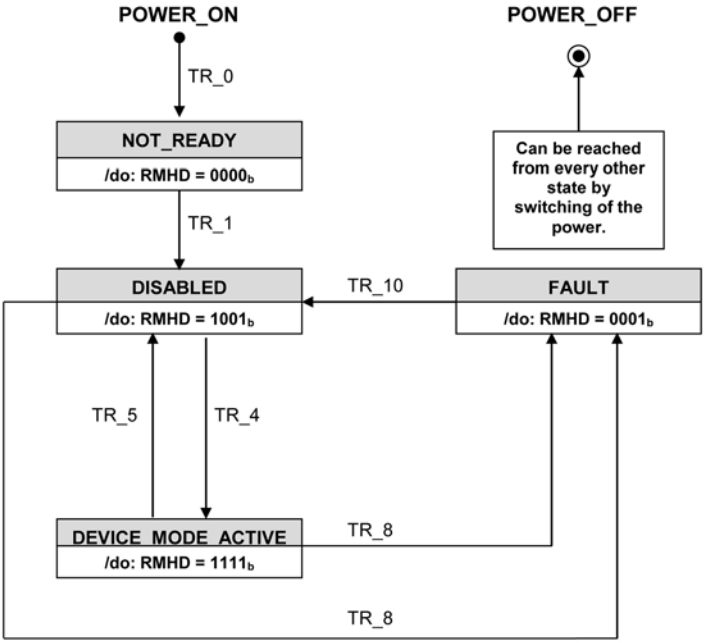
See also object 6043_h: Device control mode.

11 Device control

These objects represent the parameters used for device control.

11.1 Device State Machine (DSM)

Reduced Device State Machine as by device profile CiA DSP-408. The device state machine (DSM) describes the status of the device and the transitions between them. Any state represents a certain internal and external behaviour. Status changes are caused by the control word 6040_h or external events (like switching on the supply voltage or device faults). The current device status is presented in the status word (6041_h bits 0..3).



The table below describes the states and the actions performed in them:

Zustand	Beschreibung
NOT_READY	The electronic circuit has power. Device initialization is running (e.g. communication interface, hardware, software) Device function disabled, output stage deactivated.
DISABLED	Device parameters can be set. Device function disabled, output stage deactivated. Internal set value 0 V active.
DEVICE MODE ACTIVE	Device parameters can be set. Device function enabled, output stage activated. The set point values from the bus are active.
FAULT	Device function disabled, output stage deactivated.

The table below describes the transitions between the states:

Transition	Description	Control word bits 7 6 5 4 3 2 1 0							
		7	6	5	4	3	2	1	0
						R	M	H	D
TR_0	Power on.	Internal transition							
TR_1	Device initialization successfully completed.	Internal transition							
TR_4	Bit "Device mode active enable" active.	X	X	X	X	X	1	1	1
TR_5	Bit "Device mode active enable" inactive.	X	X	X	X	X	0	X	X
TR_8	Fault present.	Internal transition							
TR_10	Fault reset. It is mandatory that the bit changes from 0 to 1.	X	X	X	X	0	X	0	X
		X	X	X	X	1	X	0	X

11.2 Object 6040_h: Device control word PDO (CiA 408)

Controls the device status.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
6040_h	0	Device control word	UINT16	m rw	-	UINT16	-

Value description

15 14 13	12	11 10 9	8	7 6	5 4	3	2	1	0
manufacturer specific	reserved	device mode specific	control mode specific	switch parameter set	reserved	R	DM	H	D
-	-	-	-	0	-	m	m	m	m

MSB LSB

The lower 3 bits 0, 1, 2 represent the device control command. The 4th bit, bit 3, resets a fault.

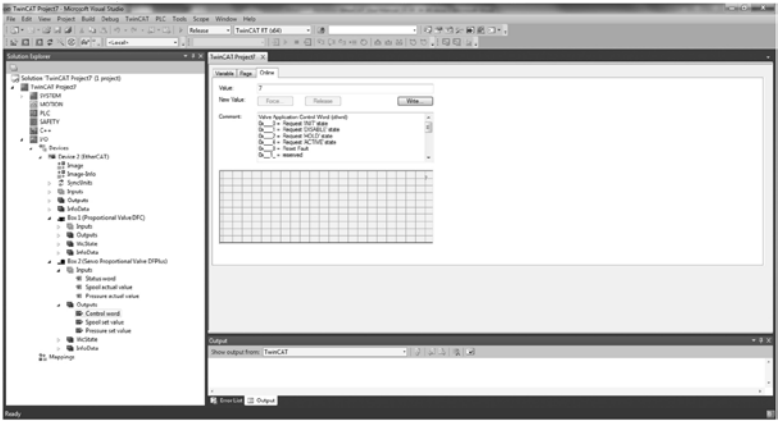
Bit 0: Disabled (D)

Bit 1: Hold enable (H)

Bit 2: Device mode active (DM)

Bit 3: Reset fault (R)

This parameter is present in the Outputs variables, see the TwinCAT view below. The button **Write…** presents the **Set Value Dialog**. For example write 7 to set the device to operational. The parameter is also present in the **CoE – Online** view. Changing it there has no use as it is kept to the value set in the **Outputs** list.



11.3 Object 6041_h: Device status word PDO (CiA 408)

Indicates the device condition.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
6040_h	0	Device control word	UINT16	m rw	-	UINT16	-

Value description:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
manufacturer specific			RT	control mode specific				reserved		W	L	R	DM	H	D
-			O	-				-		o	m	m	m	m	m

MSB LSB

The lower 4 bits 0, 1, 2, 3 show the actual state of the device state machine (DSM).

Bit 0: Disabled (D)

Bit 1: Hold enable (H)

Bit 2: Device mode active (DM)

Bit 3: Ready (R)

Bit 4: Local control. Set when instead of the bus value a local control value is used as setpoint (bit 0 in object 604F h is set to 1 by user).

Bit 15 Indicates the setting of an optional hardware disable switch which supersedes the EtherCAT® control.

This value is present in the **Inputs** list and in the **CoE – Online** view.

11.4 Object 6043_h: Device control mode (CiA 408) (D*FP Only)

The control mode is set with the parameter 6043_h.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
6043_h	0	Control mode	UINT8	rw	-	UINT8	2

At the current version the device can be run in the following control modes.

Control mode	Meaning
2	Spool position control closed loop
4	Pressure control valve closed loop

See also object $605F_h$: Device capability.

11.5 Object $604F_h$: Local (CiA 408)

When set the local analogue setpoint value supersedes the EtherCAT® setpoint. Upon this Bit 4: Local control, in object 6041_h is set.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
$604F_h$	0	Local	INT8	Rw	-	0...1	0

12 Setpoint values

Set point, actual value, device control word and device status word are located in objects that are mapped on the input output structure as can be seen in the TwinCAT System Manager views.

12.1 Object 6300_h: Valve position control closed loop setpoint (CiA 408)

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
6300_h	0	vpoc set point	UINT8	m ro	-	1...3	1
6300_h	1	Value	INT16	Rw	-	±16384	

This object is present in the CoE – Online view and in the Outputs list. Change it in the Outputs list only as a change in CoE – Online is overwritten by the Outputs..

12.2 Object 6301_h: Valve Position Control Closed Loop Actual Value (CiA 408)

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
6301_h	0	vpoc actual value	UINT8	m ro	-	1...3	1
6301_h	1	Value	INT16	ro	-	±16384	

This object is present in the CoE – Online view and in the Inputs list.

12.3 Object 6380_h: Valve pressure control closed loop setpoint (CiA 408) (D*FP only)

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
6300_h	0	vprc set point	UINT8	m ro	-	1...3	1
6300_h	1	Value	INT16	rw	-	±16384	

This object is present in the CoE – Online view and in the Outputs list. Change it in the Outputs list only as a change in CoE – Online is overwritten by the Outputs.

12.4 Object 6381_h: Valve pressure control closed loop actual value (CiA 408) (D*FP only)

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
6301_h	0	vpoc actual valve	UINT8	m ro	-	1...3	1
6301_h	1	Value	INT16	ro	-	±16384	

This object is present in the CoE – Online view and in the Inputs list.

13 Error codes

The device internal errors are mapped on the CiA 301 defined object 1001_h. The correct error information is provided by (RS232) parameter Y2 mapped on object 2000_h. Error handling and acknowledge is provided by parameter J12 mapped on object 2001_h. Emergency messages are not supported yet.

13.1 Object 1001_h: Error Register (CiA 301)

The (CANopen) device maps internal errors into this object.

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
1001_h	0	Error register	UINT8	ro	-	UINT8	-

Value description

Error register bit	Description
0	Generic error (any error)
1	Current
2	Voltage
3	Temperature
4	Communication error
5	Device profile specific
6	Reserved (always 0_b)
7	Manufacturer-specific

If a specific error occurs, the corresponding error group bit is set to 1 b . The generic error shall be set at any error situation. The bits are cleared automatically when their corresponding errors have gone.

13.2 Object 1003_h: Pre-defined error field (CiA 301)

This object shall provide the errors that occurred on the (CANopen) device and were signalled via the emergency object (not implemented). In doing so it provides an error history.

Parameter U24, Error codes + operating counter, is mapped on this object. Only the last 10 errors + minute counter in the additional information field.

Value Definition:

- The object entry at sub-index 00_h shall contain the number of actual errors that are recorded in the array starting at sub-index 01_h. Note: If no error is present the value of sub-index 00_h is 00_h and a read access to sub-index 01_h is responded with an SDO abort message (abort code: 0800 0000_h).
- Every new error is stored at sub-index 01_h; older errors are moved to the next higher sub-index.
- Writing 00_h to sub-index 00_h deletes the entire error history (empties the array). Other values than 00 h are not allowed and lead to an abort message (error code: 0609 0030_h). Don't use this option when the valve is enabled as the EEPROM erase routine blocks the regulator interrupt and the valve does not operate properly for a second.
- The error numbers are of type UINT32 and are composed of a 16-bit error code and a 16-bit additional error information field, which is manufacturer-specific. The error code is contained in the lower 2 bytes (LSB) and the additional information is included in the upper 2 bytes (MSB).

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
1008_h	0	Number of errors	UINT8	m rw	Y	00_h to FE_h	0_h
1008_h	1	Standard error field	UINT32	m ro	Y	UINT32	-
1008_h	02_h to FE_h	Standard error field	UINT32	ro	Y	UINT32	-

31	16	15	0
Extra information		Fehler Code	
MSB			LSB

Fehler Codes: siehe das unten beschriebene Objekt 2000_h Y2 Error codes.

13.3 Object 2000_h: Y2 Error Codes

These errors are also mapped on object 1001_h, see table.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
2000_h	0	Y2 Error codes	UINT16	rw	-	UINT16	-

Error Codes

Error codes (additional)	Error description	Mapped on object 1001_h bit
0000_h	No errors	-
0001_h	Overcurrent shut down	1
0002_h	Command signal cable break	7
0004_h	Feedback signal cable break	7
0008_h	Under voltage guard	2
0010_h	Bus communication error	4
0020_h	SPI bus communication	7
0040_h	External ADC timeout	7
0080_h	Free	-

When errors occurred concurrently the addition of their error codes is shown. Each error code is valued only once. Error bits, when not pending, can be reset by writing a 0 b to their location (don't set a bit when the valve is enabled, this will disable the valve as if a real error occurred.

Examples:

Error code 0019_h:
overcurrent shutdown (code 0001_h) + under voltage guard (code 0008_h) + bus communication error (code 0010_h)

Error code 0028_h:
undervoltage guard (code 0008_h) + SPI bus communication (code 0020_h)

13.4 Object 2001_h: J12 Error handling/acknowledge

Errors, for which the corresponding bits are set in this object, are acknowledged automatically, when not pending.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
2001_h	0	2001_h 0 J12 Error handling/acknowledge	INT16	rw	-	INT16	-

Value	Comment
0001_h	Auto acknowledge overcurrent shut down
0002_h	Auto acknowledge command signal cable break
0004_h	Auto acknowledge feedback signal cable break
0008_h	Auto acknowledge under voltage guard
0010_h	Auto acknowledge bus communication error
0020_h	Auto acknowledge SPI bus communication
0040_h	Auto acknowledge External ADC timeout
0080_h	Free
0100_h	On error +12V at diagnose A
0200_h	On error 2mA at diagnose B

14 Monitoring

14.1 Hardware Monitoring

14.1.1 Object 2802_h: U22 Power supply voltage [1/10 V]

Gives the actual supply voltage in 1/10V, normally around 240.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
2802_h	0	U22 Power supply voltage	INT16	ro	-	INT16	-

14.1.2 Object 2803_h : U26 Maximal supply voltage and error count

Gives the maximal supply voltage measured and the number of under voltage dips registered.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
2803_h	0	Maximal supply voltage and error count	UINT8	ro		1...2	2
2803_h	1	Maximal supply voltage [v]	UINT16	ro	-	UINT16	-
2803_h	2	Error count	UINT16	ro	-	UINT16	-

14.1.3 Object 2804_h: J8 Supply voltage supervision [1/100 V]

Used to set the under voltage limit in 1/100V. Use the object 1010_h to store this value in non-volatile.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
2802_h	0	U22 Power supply voltage	INT16	ro	-	INT16	-

14.1.4 Object 2805_h: J9 Supply voltage min time in limit [ms]

Used to set the time in milliseconds the supply voltage is allowed to drop below the supervision voltage without generating an error. Use the object 1010_h to store this value in non-volatile.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
2805_h	0	J9 Supply voltage min time in limit [ms]	INT16	rw	Y	0...3200	1000

14.1.5 Object 2806_h: U18 Temperature

Gives the actual temperature of the PCB in °C.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
2806_h	0	U18 Temperature	INT16	ro	-	INT16	-

14.1.6 Object 2808_h: U25 Temperature extremes

Minimal and maximal temperature measured in °C.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
2808_h	0	U25 Temperature extremes	UINT8	ro	Y	1...2	2
2808_h	1	Minimal temperature	UINT16	ro	Y	UINT16	-
2808_h	2	Maximal temperature	UINT16	ro	Y	UINT16	-

14.1.7 Object $280A_h$: U23 Load current

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
$280A_h$	0	U23 Load current	INT16	ro	-	INT16	-

14.1.8 Object $280D_h$: U20 Operating time day counter

Time in days the valve is operational and enabled. Can be changed by writing a value. Shows together with object $280E_h$ the operating time.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
$280D_h$	0	U20 Operating time day counter	UINT16	rw	Y	UINT16	-

14.1.9 Object $280E_h$: U21 Operating time minute counter

Time in minutes the valve is operational and enabled. Can be changed by writing a value. Shows together with object $280D_h$ the operating time.

Index	Sub index	Parameter name	Data type	Access	Persis-tence	Value range	Default
$280E_h$	0	U20 Operating time minute counter	UINT16	rw	Y	0...1439	-

15 Storing / restoring parameters

The electronics of the device provides a non-volatile memory which allows storing parameters. The current values of all parameters declared as non-volatile (persistence = Y) can be stored in non-volatile memory on the device. Three storing / restoring operations are possible:

- Parameters can be stored in non-volatile memory
- Parameters are loaded during a power on cycle
- Factory settings can be restored

During power on the user parameters stored in non-volatile memory are loaded in volatile memory to be worked with. The user can change parameters to adjust the behaviour of the device. After a power cycle these changes are gone unless they are stored in non-volatile. When messed up the factory settings have to be restored.

15.1 Object 1010_h: L2 Store parameters (CiA 301)

Storing is performed when the signature 65766173_h (“save”) is written to the parameter. The parameters are stored in user memory and loaded as working parameters each power cycle.

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
1010_h	0	Highest sub-index supported	UINT8	m ro	-	01_h to $7F_h$	01_h
1010_h	1	Save all parameters	UINT32	m rw	-	UINT32	-

15.2 Object 1011_h: L1 Restore default parameters (CiA 301)

The factory settings can be restored by writing the signature $64616F6C_h$ (“load”) to the parameter.

Index	Sub index	Parameter name	Data type	Access	Persistence	Value range	Default
1010_h	0	Highest sub-index supported	UINT8	m ro	-	01_h to $7F_h$	01_h
1010_h	1	Restore all default parameters	UINT32	m rw	-	UINT32	-