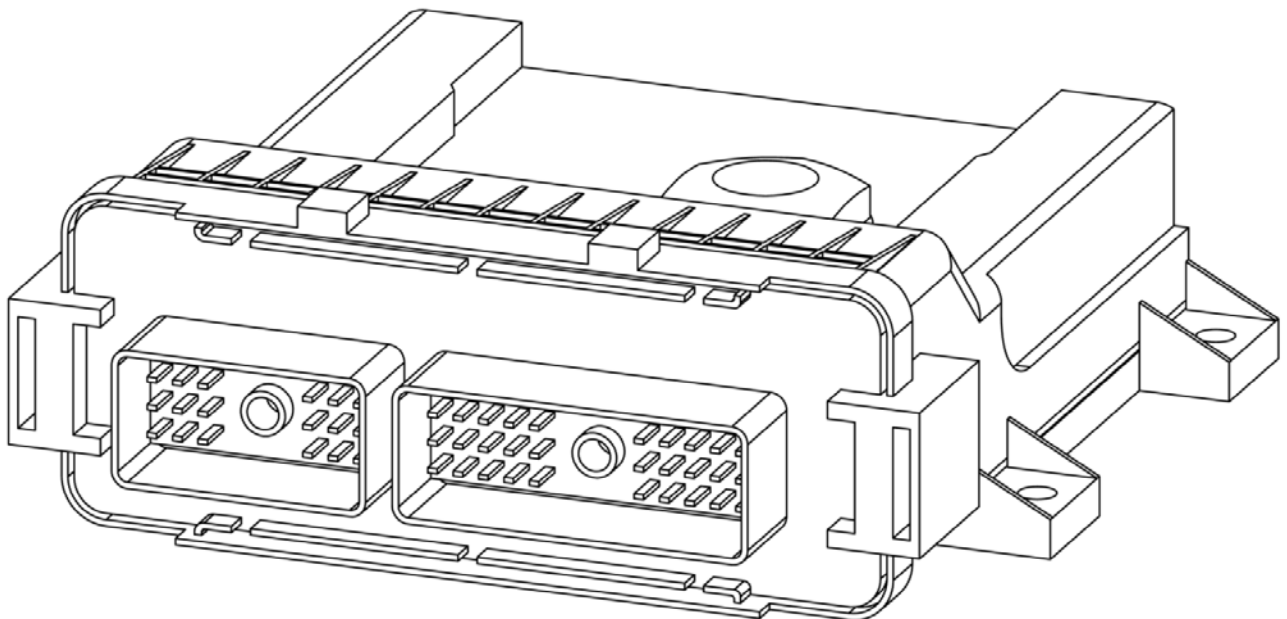


OPERATING INSTRUCTIONS MD2

CANopen - Protocol with Device Profile in accordance with CiA DSP 408

Revision 3



0 Contents

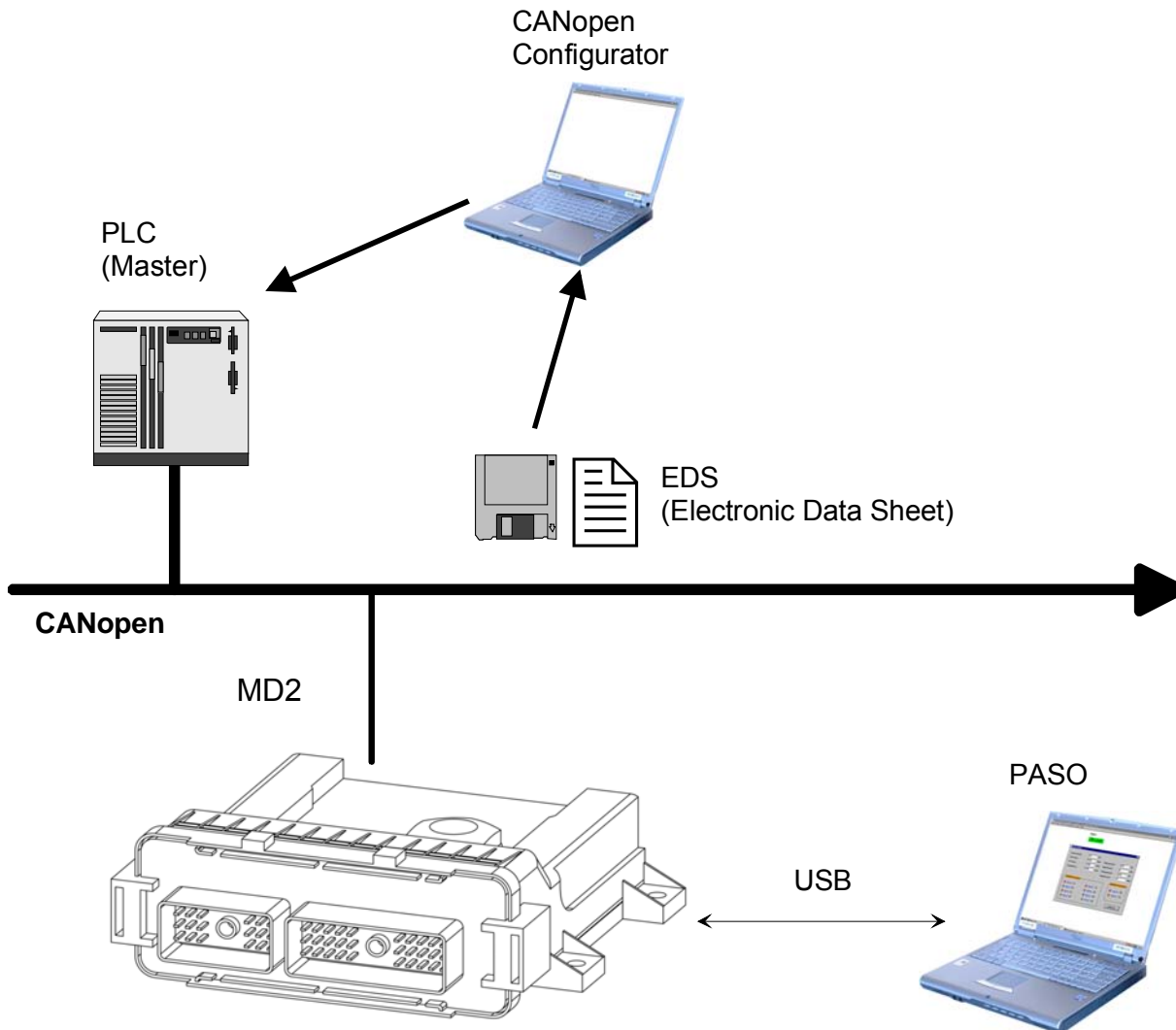
0	Contents	2
1	CANopen Technology	4
1.1	EDS Files	4
2	Product Description	5
2.1	General	5
2.2	Technical Data	5
2.2.1	Physical Layer	5
2.2.2	Fieldbus Wiring	6
2.2.3	Transmission Speeds	6
2.2.4	Other connectors	7
2.3	Fieldbus Settings	7
2.4	Fieldbus Diagnostics	8
2.5	CAN-Bus Parameters	9
2.6	Connection Example	9
3	Representation of a CAN Message	10
3.1	Data Coding with CAN / CANopen	10
4	Communication Profile	11
4.1	Default- Identifier-Distribution	11
4.2	Process Data Communication (PDO)	13
4.2.1	PDO Communication Parameters	13
4.2.2	Example for MD2 Electronics PDO:	14
4.3	Service Data Communication (SDO)	16
4.3.1	Upload SDO Protocol	16
4.3.2	Download SDO Protocol	17
4.3.3	Abortion of a Parameter Communication	17
4.4	Emergency Objects (EMCY)	18
4.4.1	Emergency Object Data	18
4.5	Network Management Objects (NMT)	19
4.5.1	Device Control Services	19
4.5.2	Error Control Services	21
4.5.3	Bootup Service	21
4.6	Synchronous Transmission (SYNC)	22
5	The Device Profile DSP-408 (in accordance with CiA)	23
5.1	Device architecture	23
5.2	Device Control	24
5.2.1	Local control	24
5.2.2	Device State Machine	26
5.3	Program Control	29
5.4	MD2 Object Dictionary	31
5.4.1	Scaled parameter	31
5.4.2	Internal bus resolution	31
5.4.3	Overview MD2 Object Dictionary	32
5.4.4	Device Type	34
5.4.5	Error register (Flag)	34
5.4.6	Predefined Error Field	34
5.4.7	COB-ID SYNC	35
5.4.8	Guard Time	35
5.4.9	Life Time Factor	36
5.4.10	Save Parameter	36
5.4.11	Restore Defaults	37
5.4.12	COB-ID Emergency	38
5.4.13	Producer Heartbeat Time	39
5.4.14	Identity Object	39
5.4.15	Communication parameter RxPDO1 – RxPDO8	40
5.4.16	Mapping RxPDO1 – RxPDO8	41
5.4.17	Communication parameter TxPDO1 – TxPDO8	42

5.4.18	Mapping TxPDO1 – TxPDO8	44
5.4.19	Device control word	46
5.4.20	Device status word	47
5.4.21	Device mode (Command value mode)	47
5.4.22	Device mode (actual value mode)	49
5.4.23	Actual value mode (Controller only).....	49
5.4.24	Speed positive	49
5.4.25	Speed negative.....	50
5.4.26	Device control mode (Control mode).....	50
5.4.27	Device local (Operating mode)	51
5.4.28	Device version	51
5.4.29	Device serial number	51
5.4.30	Device model description.....	51
5.4.31	Device vendor name.....	52
5.4.32	Device capability.....	52
5.4.33	Device temperature	53
5.4.34	Device node address	53
5.4.35	Device node baudrate.....	53
5.4.36	Command value	55
5.4.37	Command value for n-point control (Controller only)	55
5.4.38	Actual value input 16 bits (MD2-controller only).....	56
5.4.39	Actual value input 32 bits (MD2-controller only).....	56
5.4.40	Actual value (Controller only).....	56
5.4.41	Control deviation.....	57
Commissioning.....		58
5.5	Step by step instructions for the first commissioning.....	58
5.5.1	Test the hydraulic system	58
5.5.2	Adjust the mode of operation.....	58
5.5.3	Test the fieldbus	59
5.5.4	Test the control via the fieldbus	59
5.6	Presupposition for the DP-Slave controller card.....	59
5.7	Presupposition and information for the Fieldbus master	60
5.8	Delivery state	60
5.9	Settings	60
5.10	CAN Bus demand value.....	60
5.11	Starting after an error.....	60
6	Diagnostic and error detection	61
6.1	Diagnostic about the Fieldbus	61
7	Version Index.....	61

1 CANopen Technology

CANopen is an independent of the manufacturer, open fieldbus standard with a wide range of application in manufacturing- and process automation. Independence of the manufacturer and openness are guaranteed by the physical interface in accordance with ISO 11898 and CiA DS-102. With this, the basis is given for a worldwide CAN (Controller Area Network) distribution embracing all manufacturers.

CANopen provides functionally graded communication protocols (communication profiles), WANDFLUH for the MD2 Electronics uses the device profile DSP-408 "Device Profile Fluid Power Technology".



1.1 EDS Files

The characteristic communication features of a CANopen – device are defined in the form of an electronic data sheet (Electronic Data Sheet, EDS file). WANDFLUH makes available the corresponding EDS – file for the MD2 Electronics.

The EDS files expand the open communication right to the user level. All modern planning tools make it possible to read-in the EDS files during the configuration. As a result, the integration into the CANopen - system becomes simple and user friendly.

2 Product Description

2.1 General

The present operating instructions represent a CANopen-specific extension of the MD2 Electronics operating instructions.

Remark: Please read the operating instructions of the MD2 Electronics beforehand.

First of all it is important to familiarize oneself with the CANopen-terms, which occur time and again in these operating instructions. On principle, two CANopen profiles are referred to:

- The CANopen-communication profile regulates the "how" of the communication. It specifies elements for the exchanging of real-time data and parameter data as well as a simplified network management.
- The CANopen-device profile describes the "what" of the communications. The objective of device profiles is to define the data contents independent of the manufacturer, so that the basic functionality of the different device classes can be uniformly addressed.

2.2 Technical Data

The physical interface corresponds to the standard ISO 11898.

The CAN-protocol in accordance with ISO 11898 corresponds to the data connection layer Layer 2 in the ISO/OSI-reference model and supports a multi-master operation, i.e., every participant can request the communication through the serial bus.

Used as application layer is the protocol CANopen, which is standardized in the International Manufacturers and Users Association CAN in Automation (CiA).

Device Profile	DSP-408 (CiA)
Application Layer <ul style="list-style-type: none"> • CANopen communication profile (DS-301) • Timing, communication services, network management 	ISO / OSI Layer 7
These layers are not used	Layer 3 ... 6
Data Link Layer <ul style="list-style-type: none"> • Message validation • Bus-arbitration • Build-up of the message frame • Receipt acknowledgement (Acknowledge) • Error identification, signaling and suppression • Transmission speed and bus timing 	Layer 2
Physical Layer <ul style="list-style-type: none"> • Signal level and bit representation • Transmission medium 	Layer 1

2.2.1 Physical Layer

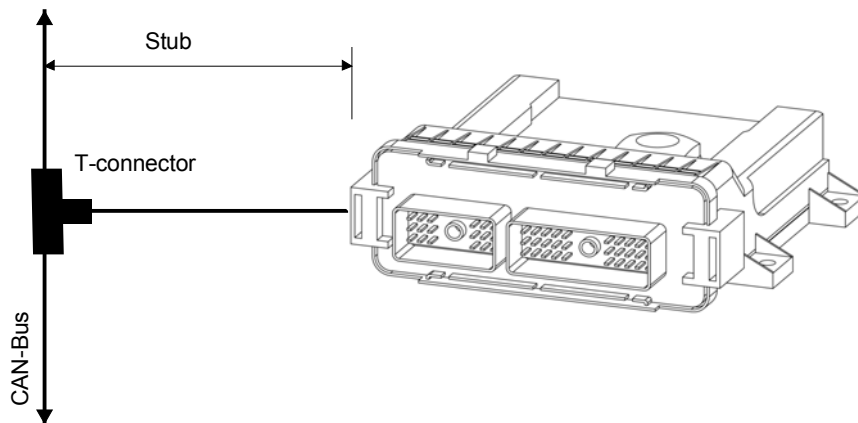
The transmission medium is an electric two-wire. The signal transmission takes place differentially (3.5/1.5V dominant //2.5V recessive).

The CAN- bus has to be terminated at both ends with a 120 Ohm resistor. The MD2 itself does not have a termination resistor. After the installation of the CAN-bus and the termination resistors, the resistance between the two lines should be measured with an ohmmeter; it has to be within the range of 60...70 Ohm.

2.2.2 Fieldbus Wiring

The bus wires may be routed parallel, twisted and/or shielded, depending on the EMC requirements. The wiring topology should be as close as possible to a single line structure, in order to minimize reflections. The cable stubs for connection of the bus nodes should be as short as possible, especially at high bit rates. At 1Mbit/s, the length of the cable stubs should not exceed 0,3m (20kBit/s=7,5m, 125kBit/s=3,7m, 500kBit/s=0,7m).

The MD2 has two connections for the CAN. This allows to grind the CAN-bus through the MD2, which eliminated the stub line.



The wiring of a CANopen "MD2" takes place through the plug X2 on the MD2 housing. The pin assignment corresponds to the following table.

Pin	Signal	Beschreibung
A3	CAN_High IN	Bus signal plus
B3	CAN_Low IN	Bus signal minus
C3	CAN_Gnd IN	Masse (CAN)
D3	CAN_High OUT	Bus signal plus
E3	CAN_Low OUT	Bus signal minus
F3	CAN_Gnd OUT	Masse (CAN)

2.2.3 Transmission Speeds

The transmission speed and the maximum admissible bus lengths mutually influence one another:

Fieldbus cable length	Line resistance	Bus-Line cross-section	Termination Resistance	Max. bit rates
0...40m	70mΩ/m	0,25 ... 0,34mm ² (AWG23, AWG22)	124Ω (1%)	1000 kBit/s at 40m
40...300m	< 60mΩ/m	0,34 ... 0,6mm ² (AWG22, AWG20)	127Ω (1%)	<= 500 kBit/s at 100m
300...600m	< 40mΩ/m	0,50 ... 0,6mm ² (AWG20)	150Ω...300Ω	<= 100 kBit/s at 500m
600...1000m	< 26mΩ/m	0,75 ... 0,8mm ² (AWG18)	150Ω...300Ω	<= 50 kBit/s at 1000m

To minimize the voltage drop on long distances the termination resistor should be higher than in the ISO 11898-2 standard.

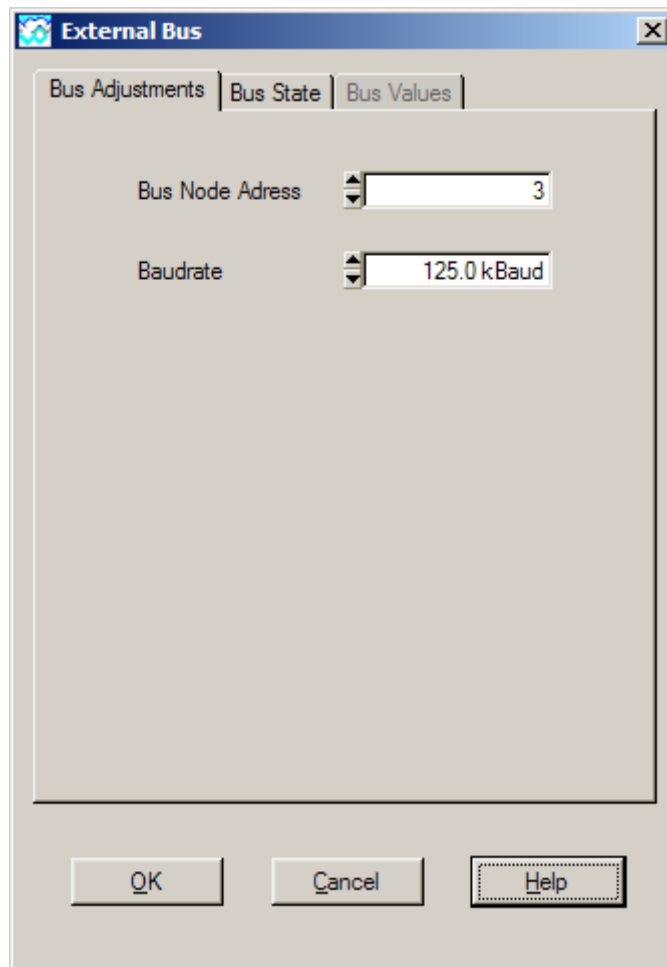
Remark: The transmission speed can be parameterized on the MD2 Electronics, resp. set via the parameterization software PASO (menu "Fieldbus-Fieldbus-Info").

2.2.4 Other connectors

For all other connectors, please refer to the appropriate operating manual e.g. "Operating instructions MD2 Electronics".

2.3 Fieldbus Settings

The following settings can be made either via the parameterization software PASO (menu "Fieldbus") or directly from the master via the CANopen Master:

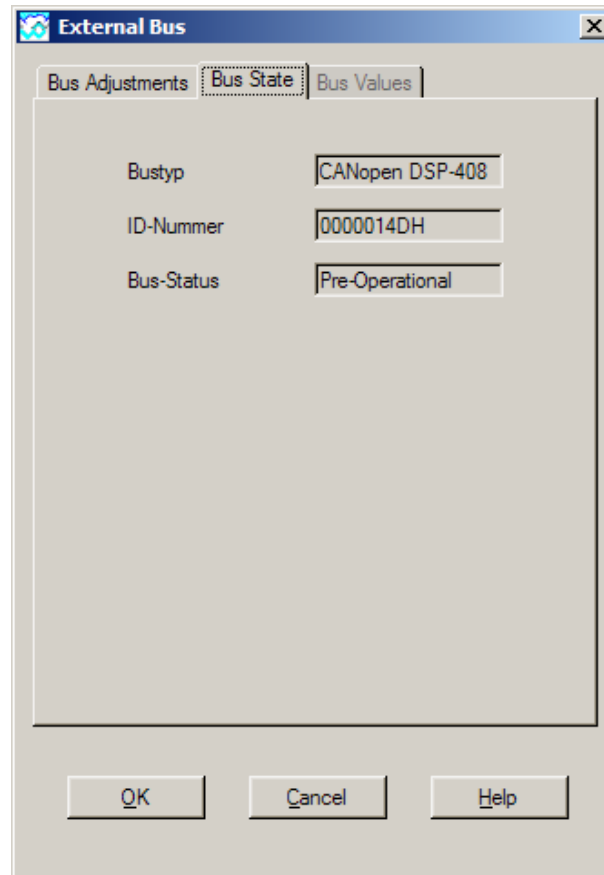


Field	Parameter Description	Display
Node address	With this parameter, the required node address for the CAN-Slave MD2 can be set. The value set is saved on the MD2 in the non-volatile memory.	1 ... 127
Baud rate	With this parameter, the required baud rate for the CAN-Slave MD2 can be set. The value set is saved on the MD2 in the non-volatile memory.	10, 20, 50, 125, 250, 500, 1000kbaud

Note: The factory setting for the node address is 1.
 The factory setting for the node baudrate is 20kBit/s

2.4 Fieldbus Diagnostics

A diagnosis of the fieldbus is possible at any time via the parameterization software PASO. This takes place through the menu point "Fieldbus_Fieldbus-Info".

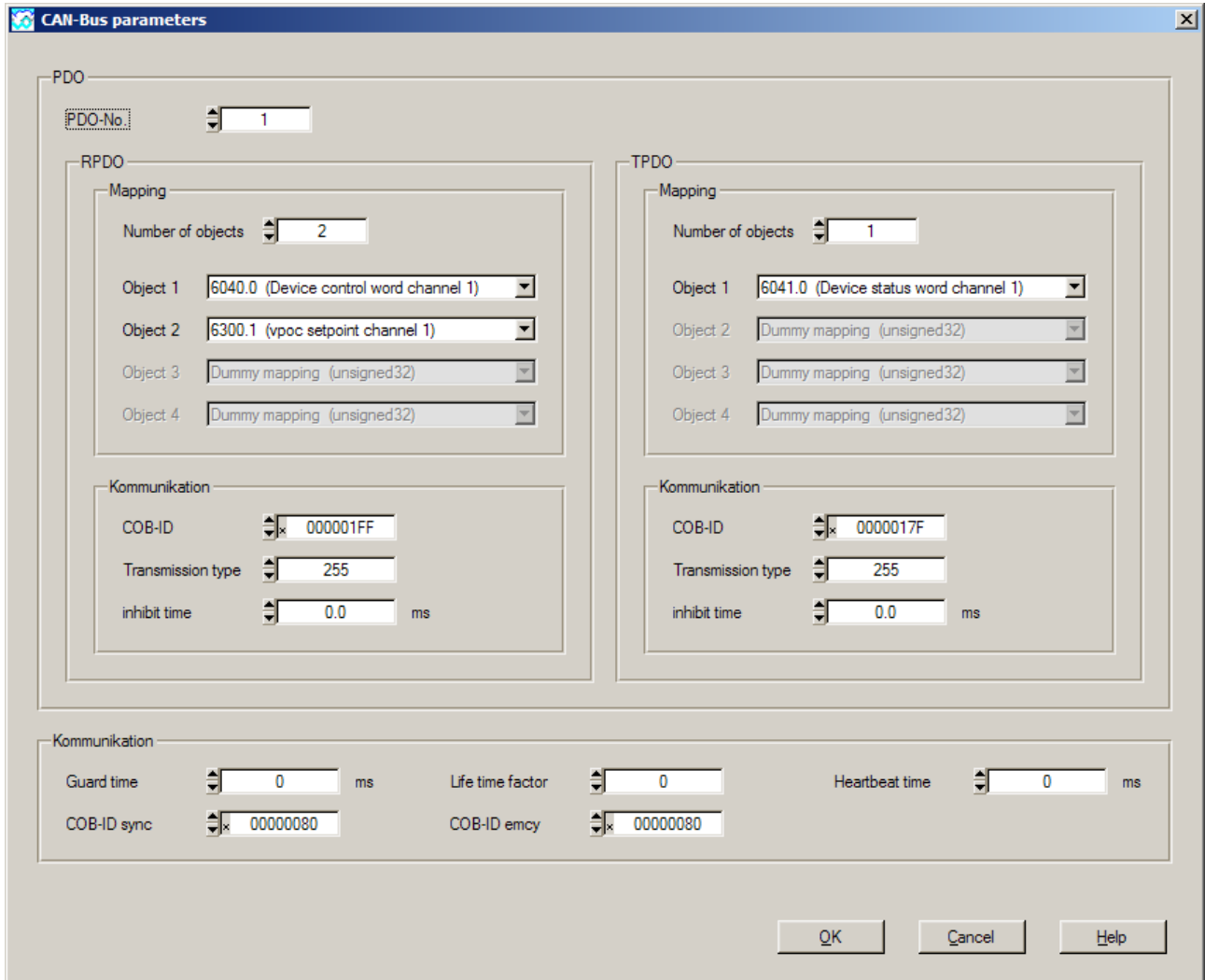


The following bus statuses are displayed:

Field	Parameter description	Display
Bustype	The type of fieldbus connected and the device profile used are displayed here.	CANopen DSP-408
ID - number	The identification number of the CAN-Slave MD2. This number is predefined fixed.	
Bus - status	The status of the communication state machine is displayed in this field. The corresponding description of the individual statuses can be found in the chapter "Start of the controlling (Boot-Up)".	Initialization Pre-Operational Operational Stopped

2.5 CAN-Bus Parameters

The menu item "Fieldbus - CAN-Bus parameters" in the parameterization software PASO allows the setting of the PDO communication and PDO mapping parameters (refer to "Communication parameter RxPDO1 – RxPDO" page 40). These parameters can only be changed in the bus status "Stop" or "pre-operational" (refer to "Device Control Services" page 19).



The screenshot shows the 'CAN-Bus parameters' configuration window. It is divided into several sections:

- PDO:** A dropdown menu for 'PDO-No.' is set to '1'.
- RPDO (Receive PDO):**
 - Mapping:** 'Number of objects' is set to 2. Object 1 is '6040.0 (Device control word channel 1)', Object 2 is '6300.1 (vproc setpoint channel 1)', and Objects 3 and 4 are 'Dummy mapping (unsigned32)'.
 - Kommunikation:** COB-ID is '000001FF', Transmission type is '255', and inhibit time is '0.0 ms'.
- TPDO (Transmit PDO):**
 - Mapping:** 'Number of objects' is set to 1. Object 1 is '6041.0 (Device status word channel 1)', and Objects 2, 3, and 4 are 'Dummy mapping (unsigned32)'.
 - Kommunikation:** COB-ID is '0000017F', Transmission type is '255', and inhibit time is '0.0 ms'.
- Global Kommunikation (Communication):**
 - Guard time: 0 ms
 - Life time factor: 0
 - Heartbeat time: 0 ms
 - COB-ID sync: 00000080
 - COB-ID emcy: 00000080

Buttons for 'OK', 'Cancel', and 'Help' are located at the bottom right of the window.

The settings of the PDO mapping is done individually per PDO. In the selection "PDO-No." can be selected, which PDO-setting should be changed.

Only the mapping and the communication parameters can be set per PDO. In the bottom of the window are global communications parameters which are not PDO-dependent.

2.6 Connection Example

As a connection example, reference is made to the operating instructions of the MD2 Electronics.

3 Representation of a CAN Message

In the operating instructions, CAN-messages are described in detail in tables, such as are illustrated below. The split-up corresponds to the usual standard CAN-driver of the software interface (2 byte CAN-Header, 8 byte user data and 3 byte transmission failure detection).

Serial data stream:

	Identifier	RTR	DLC	useful dates	CRC	ACK	EOF
No. of bits	11	1	4	0...64	15	2	7

CAN Header

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
M-Byte 0	ID 10	ID 9	ID 8	ID 7	ID 6	ID 5	ID 4	ID 3
M-Byte 1	ID 2	ID 1	ID 0	RTR	DLC 3	DLC 2	DLC 1	DLC 0

CAN Data

M-Byte 2	Data	Data	Data	Data	Data	Data	Data	Data
M-Byte 3	Data	Data	Data	Data	Data	Data	Data	Data
M-Byte 4	Data	Data	Data	Data	Data	Data	Data	Data
M-Byte 5	Data	Data	Data	Data	Data	Data	Data	Data
M-Byte 6	Data	Data	Data	Data	Data	Data	Data	Data
M-Byte 7	Data	Data	Data	Data	Data	Data	Data	Data
M-Byte 8	Data	Data	Data	Data	Data	Data	Data	Data
M-Byte 9	Data	Data	Data	Data	Data	Data	Data	Data

CAN Trailer

M-Byte 10	CRC 14	CRC 13	CRC 12	CRC 11	CRC 10	CRC 9	CRC 8	CRC 7
M-Byte 11	CRC 6	CRC 5	CRC 4	CRC 3	CRC 2	CRC 1	CRC 0	ACK 1
M-Byte 12	ACK 0	EOF 6	EOF 5	EOF 4	EOF 3	EOF 2	EOF 1	EOF 0

M-Byte x :Message Byte x
 ID 10 ... ID 0 :CAN-Identifier (COB-ID)
 RTR :Remote transmission request-bit
 DLC 3 ... DLC 0 :Data length code, length of the user data (value range 0...8, binary coded)
 Data :User data
 CRC 14 ... CRC 0 :Cyclic redundancy check
 ACK 1 ... ACK 0 :Acknowledge
 EOF 6 ... EOF 0 :End of frame

For the following documentation the CAN messages are represented in the following format. Only relevant dates like COB-ID and data bytes are shown.

M-Byte 0...1 (CAN Header)	M-Byte 2...9 (CAN data)	M-Byte 10...12 (CAN Trailer)
COB-ID	Byte 0...7 (user data)	Not used in further descriptions
e.g. 384 + node address	e.g. Status word	

3.1 Data Coding with CAN / CANopen

Fundamental rule of the data coding for CAN / CANopen:

- **First** transmitted is: the highest value bit of the lowest value byte.
- **Last** transmitted is: the lowest value bit of the highest value byte.

→ This corresponds to the description known as "Intel" format.

4 Communication Profile

The CANopen-communication profile (CiA DS-301) regulates the "how" of the communication. It specifies elements for the exchanging of real-time data and parameter data as well as a simplified network management. In this, particular attention has been paid to the resource-saving implementability and with this to the good performance of the corresponding software layer.

CANopen uses the following services for the different types of data. The communication profile contains:

- PDO (Process Data Object)
for real-time data, max. 8 bytes (1 telegram), high priority telegram.
- SDO (Service Data Object)
for system parameters, data split-up over several telegrams, low priority telegram.
- NMT (Boot_Up)
- Node-guard (life-/node guarding)
- Emergency (status)
- SYNC (synchronization)

4.1 Default- Identifier-Distribution

After switching-on the WANDFLUH MD2 Electronics control system, it has a default-identifier-distribution in accordance with the CANopen standard CiA DS 301. The following table provides an overview of this distribution from the point of view of the control system:

Object	Identifier (binary)	resulting COB - ID (decimal / hex)	Function	Objects for Comm. parameter / Mapping parameter
Broadcast objects				
NMT	000000000000	0	Boot-Up	-
SYNC	000100000000	128 (80h)	Synch	1005h, 1006h, 1007h
TIME Stamp	001000000000	256 (FFh)	Time stamp object	1012h, 1013h
Point to point objects (referred to node address)				
Emergency	0001xxxxxxx	128 (80h) + node address	Emergency telegram	1014h, 1015h
Rx_PDO1	0100xxxxxxx	512 (200h) + node address	Receive PDO1	1400h / 1600h
Rx_PDO2	0110xxxxxxx	768 (300h) + node address	Receive PDO2	1401h / 1601h
Rx_PDO3	1000xxxxxxx	1024 (400h) + node address	Receive PDO3	1402h / 1602h
Rx_PDO4	1010xxxxxxx	1280 (500h) + node address	Receive PDO4	1403h / 1603h
Tx_PDO1	0011xxxxxxx	384 (180h) + node address	Transmit PDO1	1800h / 1A00h
Tx_PDO2	0101xxxxxxx	640 (280h) + node address	Transmit PDO2	1801h / 1A01h
Tx_PDO3	0111xxxxxxx	896 (380h) + node address	Transmit PDO3	1802h / 1A02h
Tx_PDO4	1001xxxxxxx	1152 (480h) + node address	Transmit PDO3	1803h / 1A03h
Tx_SDO	1011xxxxxxx	1408 (580h) + node address	Transmit SDO (Parameter)	1200h
Rx_SDO	1100xxxxxxx	1536 (600h) + node address	Receive SDO (Parameter)	1200h
NMT Error Control	1110xxxxxxx	1792 (700h) + node address	Life - /node guarding	1016h, 1017h

xxxxxxx = Control - ID = node address settable via parameterization software PASO

4.2 Process Data Communication (PDO)

The real-time data transfer is performed by means of "Process Data Objects (PDO)". The transfer of PDOs is performed with no protocol overhead. CANopen also defines the default settings for the process data exchange, such as, e.g. identifiers used, data assignment and communication behavior. The default setting for the data assignment (default-mapping) can be modified through so-called mapping parameters.

PDOs can be transmitted either event-controlled (asynchronous) or synchronized. Also the requirements via the CAN-Feature "Remote-Transmit-Request" are supported. With this, it is possible to flexibly meet the requirements of the application.

There are two kinds of use for PDOs. The first is data transmission and the second is data reception. It is distinguished in Transmit-PDOs (TPDOs) and Receive-PDOs (RPDOs).

4.2.1 PDO Communication Parameters

The PDO communication parameters (index 1400...1403 and 1800...1805) describe the transmission behavior of the PDOs. There the PDO identifiers, the type of transmission, the transmission inhibit time and the CMS priority group are listed.

Sub-index	Field in the PDO structure	Data type
0	Number of entries	Unsigned8
1	COB-ID	Unsigned32
2	Type of transmission	Unsigned8
3	Transmission inhibit time	Unsigned16

The type of transmission (sub-index 2) defines the transmission-/receiving characteristics of a PDO. A differentiation between a synchronous and an asynchronous transmission type is made. The asynchronous PDOs are transmitted event-controlled or in case of a remote request, the synchronous ones are triggered through a SYNC signal or transmitted by means of a time control.

Type of transmission:

Type of transmission (decimal)	PDO transmission					Remarks
	Cyclic	Acyclic	Synchro- nous	Asynchro- nous	RTR only	
0		X	X			Transmission related to SYNC.
1-240	X		X			Transmission related to SYNC, 1...240 x SYNC for one PDO.
241-251	Reserved					
252			X		X	Transmission only on remote transmission request
253				X	X	Transmission only on remote transmission request.
254				X		Only for TPDOs. Manufacturer specific.
255				X		Defined in the Device-Profile (TPDO immediate after RPDO).

COB-ID Code:

Bit number	Value	Description
31 (MSB)	0	PDO available
	1	PDO not available
30	0	RTR permitted
	1	RTR not permitted
29	0	11-bit ID
	1	29-bit ID
28 – 11	0	If bit 29 = 0
	X	If bit 29 = 1, COB-ID
10 – 0 (LSB)	X	COB-ID

A remote transmission request (RTR) must always be transmitted with the number of requested data bytes (DLC), otherwise the MD2-Electronics answers without any data bytes.

Note: The CiA does not recommend using RTR.

Important: The designation of the process data objects takes place from the point of view of the MD2 Electronics control system.

The representation of a CANopen message for a PDO can be seen from the following table.

COB-ID	Byte 0 ... 7
384 + node address	Mapped Tx_PDO1 bytes (refer to object 1A00H)

COB-ID	Byte 0 ... 7
512 + node address	Mapped Rx_PDO1 bytes (refer to object 1600H)

The corresponding COB-ID is calculated as follows:

COB-ID_TX-PDO1 = 384 + node address

COB-ID_RX-PDO1 = 512 + node address

4.2.2 Example for MD2 Electronics PDO:

RPDO1 Output data (controlword and command value, refer to object 1600h):

With Control Mode 1, 3 and 4:

COB-ID	Byte 0 + 1	Byte 2 + 3	Byte 4 + 5	Byte 6 + 7
512 + node address	controlword Channel 1	Command value VPOC Channel 1	Not used	Not used

RPDO1 Output data (controlword and command value, refer to object 1601h):

With Control Mode 1, 3 and 4:

COB-ID	Byte 0 + 1	Byte 2 + 3	Byte 4 + 5	Byte 6 + 7
768 + node address	controlword Channel 2	Command value VPOC Channel 2	Not used	Not used

RPDO1 Output data (controlword and command value, refer to object 1602h):

With Control Mode 1, 3 and 4:

COB-ID	Byte 0 + 1	Byte 2 + 3	Byte 4 + 5	Byte 6 + 7
1024 + node address	controlword Channel 3	Command value VPOC Channel 3	Not used	Not used

RPDO1 Output data (controlword and command value, refer to object 1603h):

With Control Mode 1, 3 and 4:

COB-ID	Byte 0 + 1	Byte 2 + 3	Byte 4 + 5	Byte 6 + 7
1280 + node address	controlword Channel 4	Command value VPOC Channel 4	Not used	Not used

TPDO1 Input data (statusword, refer to object 1A00h):

With Control Mode 1 and 3:

COB-ID	Byte 0 + 1	Byte 2 + 3	Byte 4 + 5	Byte 6 + 7
384 + node address	statusword Channel 1	Not used	Not used	Not used

TPDO2 Input data (statusword, refer to object 1A01h):

With Control Mode 1 and 3:

COB-ID	Byte 0 + 1	Byte 2 + 3	Byte 4 + 5	Byte 6 + 7
640 + node address	statusword Channel 2	Not used	Not used	Not used

TPDO3 Input data (statusword, refer to object 1A02h):

With Control Mode 1 and 3:

COB-ID	Byte 0 + 1	Byte 2 + 3	Byte 4 + 5	Byte 6 + 7
896 + node address	statusword Channel 3	Not used	Not used	Not used

TPDO4 Input data (statusword, refer to object 1A03h):

With Control Mode 1 and 3:

COB-ID	Byte 0 + 1	Byte 2 + 3	Byte 4 + 5	Byte 6 + 7
1152 + node address	statusword Channel 4	Not used	Not used	Not used

4.3 Service Data Communication (SDO)

The device parameters in the object directory are read and written via service data objects. Service data objects (SDO) are data structures of any size. In the case of CANopen, they are addressed via a 16-bit - index and an 8-bit sub-index.

The WANDFLUH MD2 electronic control systems operate as servers, at the request of the client (e.g., SPS), they make data available (upload), or receive data from the client (download). A transmission with the number of data bytes ≤ 4 byte is called *Expedited Transfer* and a transmission with the number of data bytes > 4 byte is called *Segmented Transfer*.

Upload:

- The client requests data together with index and sub-index of the required device parameter.
- The server responds with device parameters (including index and sub-index).

Download:

- The client transmits data together with the index and sub-index.
- The server confirms the correct receipt.

The representation of a CANopen message for an SDO can be seen from the following illustration.

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4...7
1408 + node address	Control word value xxH	Index low-byte	Index high-byte	Sub-index	Data bytes

The corresponding COB-ID is calculated as follows:

$$\text{COB-ID_SDOTX} = 1408 + \text{node address}$$

$$\text{COB-ID_SDORX} = 1536 + \text{node address}$$

On principle a handshake takes place between the client and the server. If the parameter to be transmitted comprises up to 4 bytes, then a single handshake is sufficient (a telegram pair).

During the download, the client transmits the data together with the index, sub-index and the server confirms the receipt. During the upload, the client requests the data, in that it transmits the index and sub-index of the required parameter and the server transmits the parameter (incl. index and sub-index) in its response telegram. For upload and download the same identifier pair is used.

4.3.1 Upload SDO Protocol

Client → Server, Initiate Upload Request

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4...7
1536 + node address	Control word value 40H	Index low-byte	Index high-byte	Sub-index	reserved

Server → Client, Upload Response (Expedited Transfer)

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4...7
1408 + node address	Control word value 40H + ((4 - noB) * 4 + 3)	Index low-byte	Index high-byte	Sub-index	Data bytes

Server → Client, Upload Response (Segmented Transfer)

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4...7
1408 + node address	Control word value 41H	Index low-byte	Index high-byte	Sub-index	Number of bytes to be uploaded

Client → Server, Upload Request (only for segmented Transfer)

COB-ID	Byte 0	Byte 1...7
--------	--------	------------

1536 + node address	Control word value 60H or 70H (toggle bit)	reserved
---------------------	---	----------

Client → Server, Upload Response (only for segmented Transfer)

COB-ID	Byte 0	Byte 1...7
1408 + node address	Control word value 00H or 10H (toggle bit) if no more segments + ((7 – noB) * 2 + 1)	Segmented data bytes

noB: number of valid data bytes (min. 1)

4.3.2 Download SDO Protocol
Client → Server, initiate Download Request (Expedited Transfer)

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4...7
1536 + node address	Control word value 20H + ((4 – noB) * 4 + 3)	Index low-byte	Index high-byte	Sub-index	Data bytes

Server → Client, Download Response

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4...7
1408 + node address	Control word value 60H	Index low-byte	Index high-byte	Sub-index	reserved

4.3.3 Abortion of a Parameter Communication

In the case of a faulty parameter communication, it is aborted. To do this, the client, resp. the server transmits an SDO telegram with the following structure:

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 + 5	Byte 6	Byte 7
1408 + node address	Control word value 80H	Index low-byte	Index high-byte	Sub-index	Additional code	Error code	Error class

The following error descriptions from DS 301 are supported by the WANDFLUH control system:

Error class	Error code	Additional code	
0x05	0x03	0x0000	Toggle-bit error
0x06	0x01	0x0000	Not supported access to an object
0x06	0x02	0x0000	Object does not exist in the object dictionary
0x06	0x04	0x0041	Object cannot be mapped to the PDO
0x06	0x04	0x0042	PDO length exceeded
0x06	0x04	0x0043	Value invalid
0x06	0x04	0x0047	Initialization error
0x06	0x06	0x0000	Access failed due to a hardware error
0x06	0x07	0x0010	Data type, length of service parameter does not match
0x06	0x07	0x0012	Data type, length of service parameter too high
0x06	0x07	0x0013	Data type, length of service parameter too low
0x06	0x09	0x0011	Sub-index does not exist
0x06	0x09	0x0031	Value of parameter written too high
0x06	0x09	0x0032	Value of the parameter written too low
0x08	0x00	0x0020	Data cannot be transferred or stored to the application
0x08	0x00	0x0021	No data transfer because of local control
0x08	0x00	0x0022	No data transfer because of present device state
0x08	0x00	0x0000	General error

4.4 Emergency Objects (EMCY)

If an internal error occurs, then the MD2 transmits an 8 byte long emergency telegram. This telegram is transmitted with the highest priority. An emergency object is transmitted only once per "error event". As long as no new errors on the device occur, no further emergency objects are transmitted.

4.4.1 Emergency Object Data

The emergency telegram consists of the following 8 bytes:

COB_ID	Byte 0 + 1	Byte 2	Byte 3...7
128 + node address	Error code (refer to the table below)	Error register (object 1001h)	Manufacturer-specific errors

The part "Manufacturer-specific errors" is not used.

Error Code

In the case of an error on the MD2, here a value corresponding to the error is indicated. In the object 1003H the last occurring errors are filed. In doing so, the sub-index 0 indicates the number of the current errors. The following table lists all possible errors with the corresponding error code:

Error code (Hex)	Name	Description	Reaction
0000	No error	No error is present	
1000	General error	A general error is present	FAULT
3412	Power supply voltage too low	The MD2 supply voltage is too low	FAULT
2311	Valve	Solenoid driver 1 cable break or short-circuit	FAULT
2312		Solenoid driver 2 cable break or short-circuit	FAULT
3422	Control voltage too low	The control (analog command signal) voltage is too low or there occurred a cable break	FAULT
8100	Communication	Communication Reset or stop node transition (see communication state machine)	FAULT

Error Register

As long as an error is present, it can be read out through the object 1001H (description under "Error register (Flag)" on page 34).

4.5 Network Management Objects (NMT)

The Network Management (NMT) is node oriented and follows a master-slave structure. NMT objects are used for executing NMT services. Through NMT services, nodes are initialised, started, monitored, resetted or stopped. All nodes are regarded as NMT slaves.

NMT requires that one device in the network fulfils the function of the NMT Master.

4.5.1 Device Control Services

CANopen makes it possible to start the control system with a single telegram. When switching-on (Power-On), the control system carries out an initialization and switches into the status PRE-OPERATIONAL.

With a single telegram (Start_Remote_Node), the control system now can be switched into the status OPERATIONAL.

COB-ID	Byte 0	Byte 1
0 (NMT)	Command specifier	Node address

The following states and state commands are possible:

Designation	Command-specifier	Function
Start_Remote_Node	1(dec) = 01(hex)	Starts the control system, enables the outputs, starts the transmission of PDOs
Stop_Remote_Node	2(dec) = 02(hex)	Stops the communication. Only NMT objects can still be transmitted.
Enter_Pre-Operational_State	128(dec) = 80(hex)	Stops the PDO transmission, SDO continues to be active
Reset_Node	129(dec) = 81(hex)	Carries out a control system reset
Reset_Communication	130(dec) = 82(hex)	Carries out a reset of the communication functions

Status	Description
Initialisation	<p>This state is divided into three sub-states in order to enable a complete or partial reset of the node.</p> <ul style="list-style-type: none"> • Reset-Application: In this state the manufacturer specific parameters and the profile area parameters are set to their power-on values. After this, the state Reset Communication is entered autonomously. • Reset-Communication: In this state the parameters of the communication profile area are set to their power-on values. After this the state Initialising is entered autonomously. • Initialising: This is the first sub-state the device enters after power-on. After finishing the basic node initialisation the device executes the write boot-up object service and enters the state Pre-Operational autonomously.
Pre-Operational	<ul style="list-style-type: none"> • All communication objects are allowed, with the exception of the PDO objects • PDO communication is not allowed • Device parameters and allocation of applications objects (PDO-mapping) are allowed
Operational	<ul style="list-style-type: none"> • All communication objects are allowed, resp. active • Access via SDO is possible, the application, resp. device state machine can, however, disable certain objects
Stopped	<ul style="list-style-type: none"> • In this status the whole communication is stopped, only the node guarding and heartbeat objects are still active.

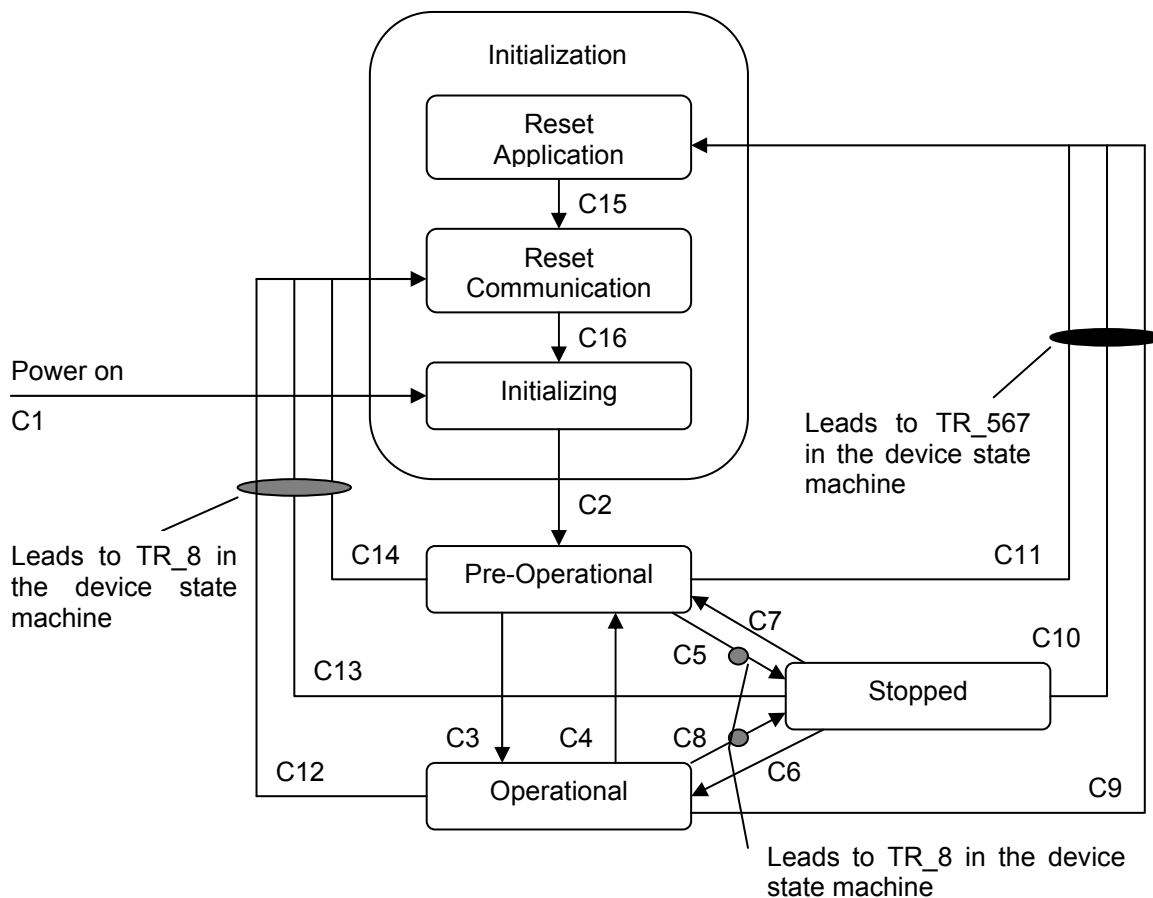
States and Communication Object Relation

The table below shows the relation between communication states and communication objects. Services on the listed communication objects may only be executed if the device is in the appropriate communication state.

	INITIALISING	PRE-OPERATIONAL	OPERATIONAL	STOPPED
PDO			X	
SDO		X	X	
Synchronisation Object		X	X	
Emergency Object		X	X	
Boot-up Object	X			
Network Management Object		X	X	X

Communication state machine:

Transition	Description
C1	When switching-on the power supply, the control system automatically passes into the initialization status
C2	Initialization completed – automatic change into the status PRE-OPERATIONAL
C3, C6	Start_Remote_Node indication
C4, C7	Enter_Pre-Operational_State indication
C5, C8	Stop_Remote_Node indication
C9, C10, C11	Reset_Node indication
C12, C13, C14	Reset_Communication indication
C15	Application reset carried out
C16	Communication reset carried out



4.5.2 Error Control Services

Through Error control services the NMT detects failures in a CAN-based Network. This is principally achieved through periodically transmitting of messages by a device. There exist two possibilities to perform the error control.

Heartbeat Protocol

The MD2 can produce a cyclic heartbeat; this can be read by the master, in order to see whether the valve is still "alive" and in what condition it is. The heartbeat protocol defines an error service, which does not require a remote frame. The cycle time for the heartbeat is set through the object 1017H, with the time 0 the heartbeat is switched off.

COB-ID	DLC	Byte 0
1792 + node address	1	0 = Boot-up 4 = Stopped 5 = Operational 127 = Pre-operational

Node Guarding Protocol

The guarding is achieved through transmitting guarding requests (Node guarding protocol) by the NMT Master. If a NMT Slave has not responded within a defined span of time (Node life time) or if the NMT Slave's communication status has changed, the NMT Master informs its Application about that event.

COB-ID	RTR	DLC
1792 + node address of the MD2 to be monitored	1	1

The MD2 addressed responds with the following telegram:

COB-ID	DLC	Byte 0
1792 + node address of the MD2 addressed	1	Status of the MD2 4 / 132(toggled) = stopped 5 / 133(toggled) = operational 127 / 255(toggled) = pre-operational The bit 7 is toggled after every telegram. If the bit is not toggled, then the NMT-Master assumes an error of this node.

The Guard Time is filed in the object 100Ch and the Life Time Factor in the object 100Dh. These entries can be read and changed by the master NMT-Master by means of an SDO - access. The time, which may pass between the node guarding telegrams until the MD2 issues an error, is called Life Time.

Calculation of the Life Time:

Life Time = Guard Time x Life Time factor

If the Life Time is exceeded, the NMT-Master does not transmit a node guarding anymore, then the MD2 transmits a corresponding emergency telegram.

4.5.3 Bootup Service

Through this service, the NMT slave indicates that a local state transition occurred from the state INITIALISING to the state PRE-OPERATIONAL.

COB-ID	Byte 0
1792 + node address	0

4.6 Synchronous Transmission (SYNC)

Synchronous transmission of message means that the transmission of the message is fixed in time with respect to the transmission of the SYNC telegram. The synchronous message is transmitted within a given time window with respect to the SYNC transmission, and at most once for every period of the SYNC.

The synchronous mechanism is intended for transferring commanded values and actual values on a fixed timely base.

In general the fixing of the transmission time at synchronous PDO messages coupled with the periodicity of transmission of the SYNC message guarantees that devices may arrange to sample process variables from a process environment and apply their actuation in a co-ordinated fashion.

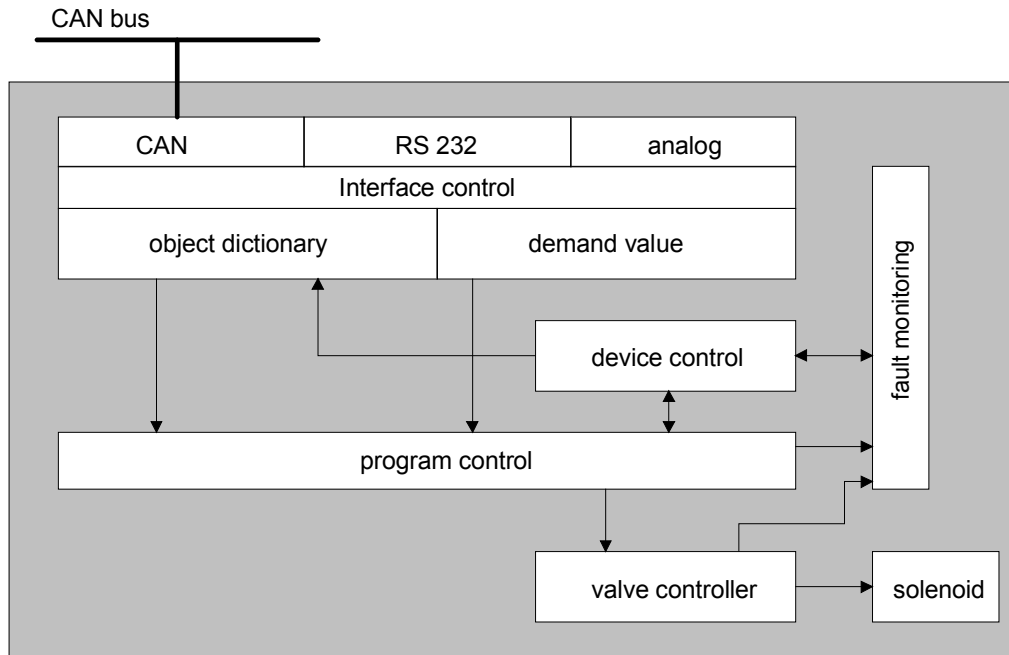
The SYNC telegram is a CAN-Message with high priority and without process dates. One device in the network generates the SYNC telegram (SYNC producer), all other devices with a synchronous PDO behavior (refer to the PDO transmission type) react to it.

To transmit a synchronous command value, the value has to be transmitted with a synchronous receive-PDO (transmit type 0...244) first. The command value will be processed until the next SYNC telegram is received. Likewise a synchronous transmit-PDO will only be sent after a SYNC telegram received.

5 The Device Profile DSP-408 (in accordance with CiA)

The device profile explains the data and their format, which are exchanged between the CANopen master and the MD2 Electronics (slave). The device profile is based on the specification of the profile "Fluid Power Technology" as defined by the VDMA (the German Engineering Federation). The device profile has been defined for hydraulic devices, such as: proportional valves, hydrostatic pumps and hydrostatic drives.

5.1 Device architecture



The DP-Slave controller card contains the complete Hardware of the MD2 Electronics. This Hardware includes the interface for the Fieldbus and the interface for the parameterisation software PASO. Also included are 1 or 2 solenoid outputs for the cylinder.

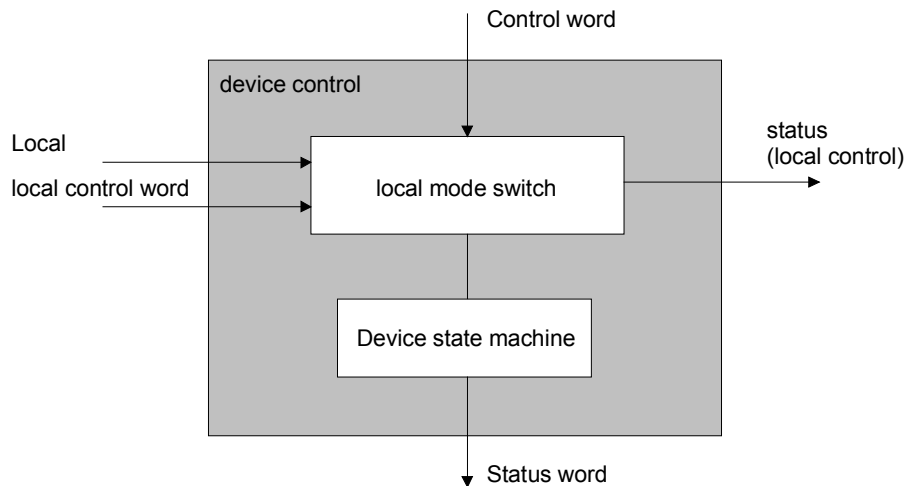
The Fieldbus control is made through a higher level Fieldbus Master.

The local control can be made via the digital- and analog in- and outputs or via parameterisation software PASO.

5.2 Device Control

The following picture shows the principle function of the CAN-Slave controller card.

5.2.1 Local control



Local mode ("local")

In the local mode, the control commands and the current states will be set resp. displayed through the parameterisation software PASO. The local mode has 2 states: "Disabled" and "Enabled", switch over through the digital input.

To activate the Local mode from the PASO, the PASO parameter "Operating mode = local" (window "Enable Channel") must be sent (condition: MD2-state "Init" or "Disabled").

To activate the Local mode from the bus, the bus parameter "Operating mode = 1" must be sent via the CAN-bus (condition: MD2-state "Init" or "Disabled").

PASO mode ("Remote PASO")

In the PASO mode, the control commands and the current states will be set resp. displayed through the parameterisation software PASO. The local mode has 2 states: "Disabled" and "Enabled", switch over through the PASO command "Enable" resp. "Disable".

To activate the PASO mode from the bus resp. local mode, the PASO command "Command - Valve operation" must be activated (condition: DSV-state "Init" or "Disabled").

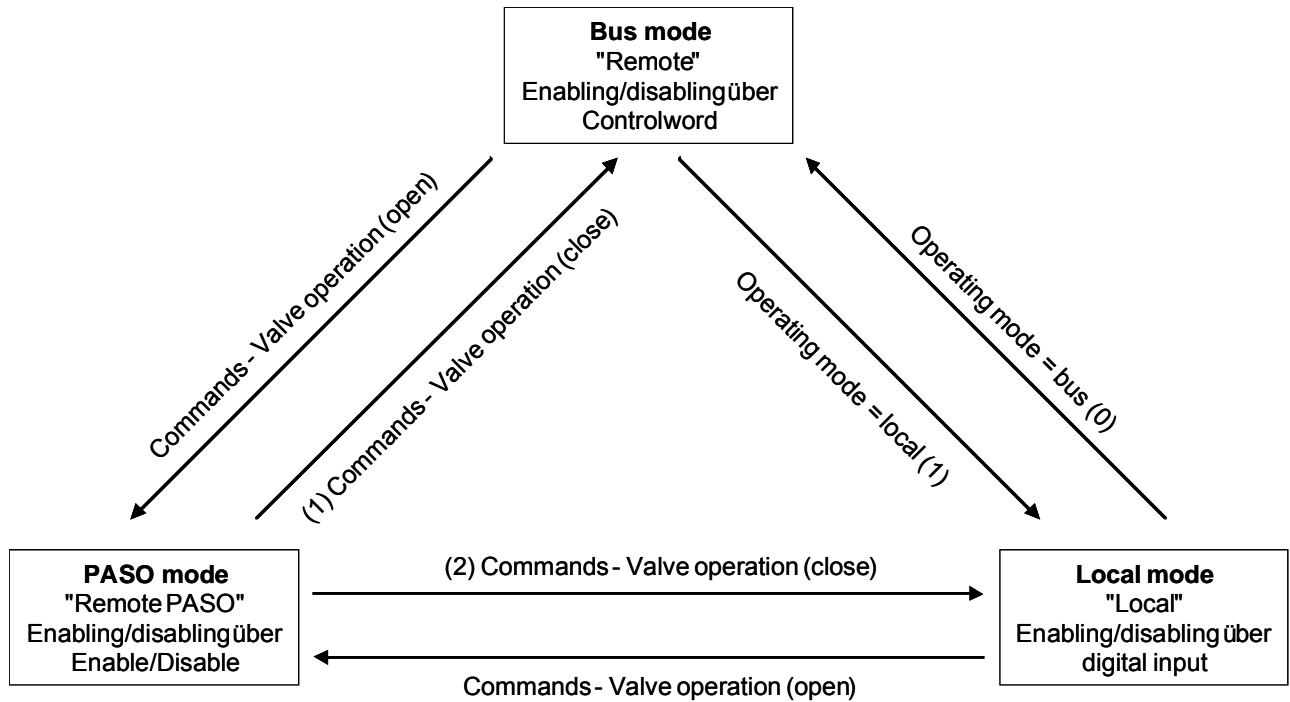
Bus mode ("Remote")

In the Bus mode, the control commands and the current states will be set resp. displayed through the CAN-bus. The Bus mode has several states (refer to chapter "Device State Machine" page 26), switch over through the Bus parameter "control word".

To activate the Bus mode from the PASO mode, the PASO command "Local Control" must be activated (condition: MD2-state "Init" or "Disabled").

To activate the Bus mode from the local mode, the bus parameter "Device local=0" must be sent via the CAN-bus (condition: MD2-state "Init" or "Disabled").

The picture on the next page shows the different possibilities of switch over the different states.



To change a operating state is only possible,
if MD2-state is "Init" or "Disabled"

- (1) if operating mode = bus (0)
- (2) if operating mode = local (1)

In the operating state "PASO mode", to send the bus
parameter "Operating mode = local" is possible

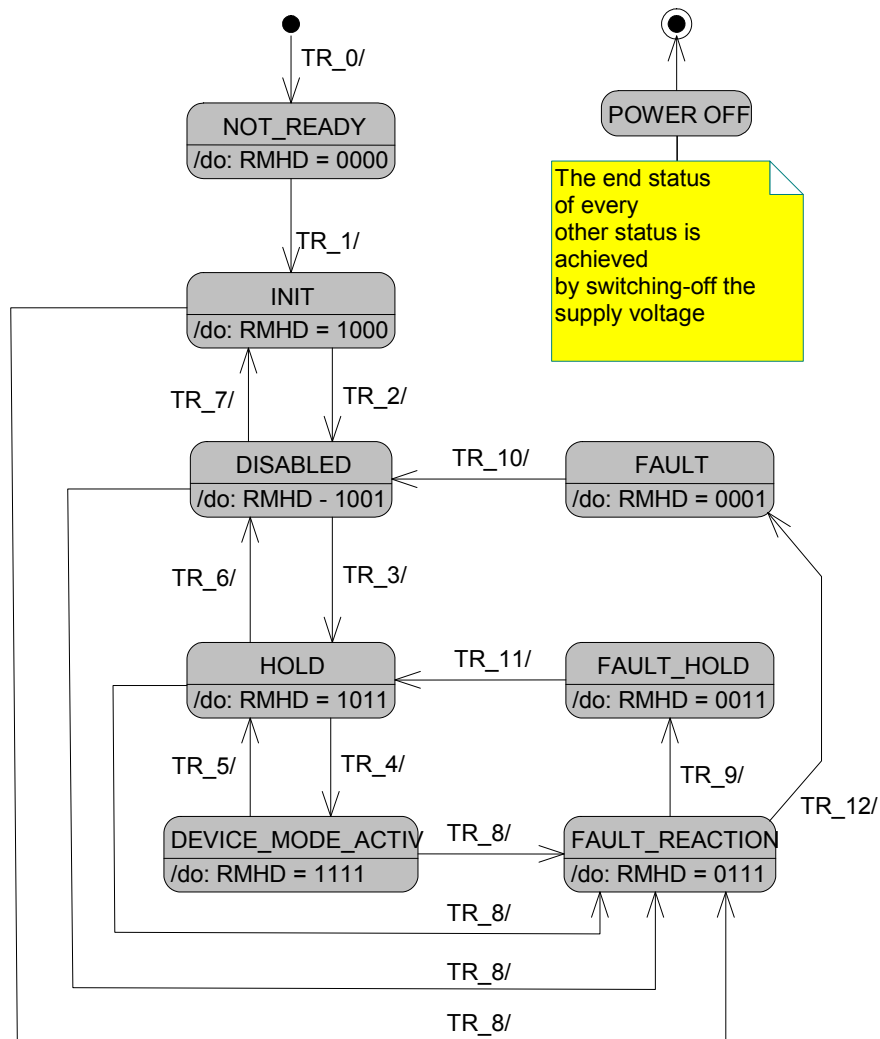
Switch over the different states

5.2.2 Device State Machine

In the following, with the help of a status diagram it is described, how the start-up of the CANopen-slave MD2 takes place and which statuses are reached when and how. Certain transitions are automatically forced by the communication state machine (refer to the chapter "Boot-up").

The following table describes the possible states and what is done in these states:

Status	Description
NOT_READY	<ul style="list-style-type: none"> The supply voltage is present on the MD2 Self test is running The device functions are disabled
INIT	<ul style="list-style-type: none"> Device parameters can be set Initialisation of device parameters with stored values The device functions are disabled
DISABLED	<ul style="list-style-type: none"> Device parameters can be set The device functions are disabled In this state, the different device modes like "Device Mode", "operating mode" and "Device Local mode" can be set.
HOLD	<ul style="list-style-type: none"> Device parameters can be set The last set-point value present is maintained active The set-point value of the status DEVICE_MODE_ACTIVE is not active Device modes setting is disabled
DEVICE_MODE_ACTIVE	<ul style="list-style-type: none"> Device parameters can be set The operating mode selected with the parameter "Control Mode" and the device mode selected with the parameter "Device Mode" are active Changing the operating mode is not possible (the writing of the parameter "Device Mode" is responded to negatively)
FAULT_HOLD	<ul style="list-style-type: none"> Device parameters can be set The actual value present is read or the set-point value of the HOLD status is active To leave this state, the corresponding transitions in the table below have to be executed.
FAULT	<ul style="list-style-type: none"> Device parameters can be set The device functions are disabled To leave this state, the corresponding transitions in the table below have to be executed.
FAULT_REACTION	This status is reached, if the device is not anymore ready for operation <ul style="list-style-type: none"> Device parameters can be set The device function can be disabled or enabled



RMHD = R: Status word "Ready" (bit 3)
 M: Status word "Device mode active enable" (bit 2)
 H: Status word "Hold enable" (bit 1)
 D: Status word "Disable" (bit 0)

The following table describes the transitions from one status to the next one:

Transition	Description	Control word bit							
		7	6	5	4	3 R	2 M	1 H	0 D
TR_0	Switching-on the supply voltage	Internal transition							
TR_1	Device initialization successfully completed	Internal transition							
TR_2	Bit "Disable" active	x	x	x	x	x	x	x	1
TR_3	Bit "Hold enable" active	x	x	x	x	x	x	1	1
TR_4	Bit "Device mode active enable" active	x	x	x	x	x	1	1	1
TR_5	Bit "Device mode active enable " not active	x	x	x	x	x	0	x	x
TR_6	Bit "Hold enable" not active	x	x	x	x	x	0	0	x
TR_7	Bit "Disable" not active	x	x	x	x	x	0	0	0
TR_8	Error present. This transition can also be forced by the communication state machine (transition C5, C8, C12, C13, C14)	Internal transition							
TR_9	Error reaction successful (HOLD active)	Internal transition							
TR_10	Error reset (return to the status DISABLED). The "reset fault" bit in the controlword imperatively has to change from 0 to 1	x x x x 0 x 0 x							
		→ x x x x 1 x 0 x							
TR_11	Error reset (return to status HOLD). The "reset fault" bit in the controlword imperatively has to change from 0 to 1	x x x x 0 x 1 x							
		→ x x x x 1 x 1 x							
TR_12	Error reaction successful (DISABLED active)	Internal transition							
TR_567	This transition can be forced by the communication state machine (transition C9, C10, C11). Or else by the input Enable from 1 → 0	x x x x x 0 0 0							

RMHD = R: Control word "Reset fault" (bit 3)
 M: Control word "Device mode active enable" (bit 2)
 H: Control word "Hold enable" (bit 1)
 D: Control word "Disable" (bit 0)

5.3 Program Control

The MD2 through the fieldbus can be set to the following operating modes; in doing so, one differentiates between the Control mode and the Device mode:

Controle mode	Description
Local operating mode	The MD2 is operated through the local possibilities such as e.g. the digital inputs and outputs or PASO.
Spool position control open loop vpc (1)	A proportional spool valve is driven with a set-point value, the set-point value is proportional to the valve opening. The spool position is not recorded and controlled (open loop). This control mode is only selectable with MD2 amplifier.
Pressure control valve open loop vprc (3)	A proportional pressure control valve is driven with a set-point value; the set-point value is proportional to the valve pressure. The pressure is not measured and controlled with a pressure sensor (open loop). This control mode is selectable with MD2 amplifier a. MD2 controller.
Pressure control valve closed loop vprc (4)	A proportional pressure control valve with 1 solenoid is driven with a set-point value; the set-point value is proportional to the valve pressure. The pressure is measured and controlled with a pressure sensor (closed loop). This control mode is only selectable with MD2 controller.
Open loop movement dcol (6)	A proportional spool valve is driven with a set-point value; the set-point value is proportional to the position of the axis. The Position is not measured and controlled with a position sensor (open loop). This control mode is only selectable with MD2 controller.
Velocity control axis dsc (7)	A proportional flow valve is driven with a set-point value; the set-point value is proportional to the valve flow. The flow is measured and controlled with a flow sensor (closed loop). This control mode is only selectable with MD2 controller.
Position control axis dpc (9)	A proportional spool valve is driven with a set-point value; the set-point value is proportional to the position of the axis. The position is measured and controlled with a position sensor (closed loop). This control mode is only selectable with MD2 controller.
Pressure control valve closed loop 2-solenoids (-5)	Wandfluh - specific A proportional pressure control valve with 2 solenoids is driven with a set-point value; the set-point value is proportional to the valve pressure. The pressure is measured and controlled with a pressure sensor (closed loop). This control mode is only selectable with MD2 controller.
2-Point controller with 1 solenoid. (-6)	Wandfluh – specific Control of a 1-solenoid valve with a switching solenoid in closed loop (with feedback signal). It can only be activated one solenoid (correspond to solenoid driver 1). This control mode is only selectable with MD2 controller.
2-Point controller with 2 solenoids. (-7)	Wandfluh – specific Control of a 2-solenoid valve with switching solenoids resp. of two 1-solenoid valves with one switching solenoid each in closed loop (with feedback signal). Two solenoids can be activated. This control mode is only selectable with MD2 controller.
3-Point controller with 2 solenoids. (-8)	Wandfluh – specific Control of a 1-solenoid valve with a switching solenoid in closed loop (with feedback signal). It can only be activated one solenoid (correspond to solenoid driver 1). This control mode is only selectable with MD2 controller.

Device mode	Description
Set-point value setting through the bus	The set-point-value setting for the CANopen-Slave MD2 takes place through the fieldbus. This corresponds to the standard device mode.

Set-point value setting locally	The set-point value setting for the CANopen-Slave MD2 takes place locally.
---------------------------------	--

5.4 MD2 Object Dictionary

TODO: Seitenumbrüche hinzufügen, nach dem Übersetzen

(In accordance with CiA DS-301 and DSP-408 "Device Profile Fluid Power Technology")

The most important part of a device profile is the Object Dictionary description. The Object Dictionary is essentially a grouping of objects accessible via the network in ordered pre-defined fashion. Each object is addressed using a 16-bit index.

The overall layout of the standard Object Dictionary is shown below.

Index (Hex)	Object
0000	Not used
0001 – 001F	Static Data Types
0020 – 003F	Complex Data Types
0040 – 005F	Manufacturer Specific Complex Data Types
0060 – 007F	Device Profile Specific Static Data Types
0080 – 009F	Device Profile Specific Complex Data Types
00A0 – 0FFF	Reserved for further use
1000 – 1FFF	Communication Profile Area
2000 – 5FFF	Manufacturer Specific Profile Area
6000 – 9FFF	Standardised Device Profile Area
A000 – FFFF	Reserved for further use

5.4.1 Scaled parameter

By parameter with an unit (e.g. mm, psi, l/min, etc.), the adjusting range and the resolution depends on the control mode and the selected unit. The following table shows the connection::

Control mode pressure control valve open loop vprc (4)

Unit:	bar	psi	kN	mPa
Range:	0...500000	0...8000000	0...1000000	0...50000
Resolution:	1/1000	1/1000	1/1000	1/1000

Control mode velocity control axis dsc (7)

Unit:	l/min	mm/s	inch/s	mPa/Min	Grad/s
Range:	0...500000	0...2000000	0...10000000	0...100000	0...360000
Resolution:	1/1000	1/1000	1/1000	1/1000	1/1000

Control mode position control axis dpc (9)

Unit:	mm	Grad	Zoll
Range:	0...2000000	0...360000	0...100000
Resolution:	1/1000	1/1000	1/1000

5.4.2 Internal bus resolution

In the Device Profile in accordance with CiA DSP 408, an internal resolution value is defined. This value is 0 ... 16384.

This scaling can with the help of PASO be adjusted to the MD2 to a given set point to be able to adapt. This scaling can be adapt through the PASO for adapting the MD2 to a given command value.

5.4.3 Overview MD2 Object Dictionary

Communication parameter:

Parameter	ID	Page
Device Type	1000	34
Error register (Flag)	1001	34
Predefined Error Field	1003	34
COB-ID SYNC	1005	35
Guard Time	100C	35
Life Time Factor	100D	36
Save Parameter	1010	36
Restore Defaults	1011	37
COB-ID Emergency	1014	38
Producer Heartbeat Time	1017	39
Identity Object	1018	39
Communication parameter RxPDO1 – RxPDO8	1400 1401	40
Mapping RxPDO1 – RxPDO8	1600 1601	41
Communication parameter TxPDO1 – TxPDO8	1800 1801	41
Mapping TxPDO1 – TxPDO8	1A00 1A01	44

Function parameter global:

Parameter	ID	Page
Device version	6050	51
Device serial number	6052	51
Device model description	6054	51
Device vendor name	6057	52
Device capability	605F	52
Device temperature	2002	53
Device node address	2000	53
Device node baudrate	2001	53

Function parameter channel depending:

Parameter	Channel number								Page
	1	2	3	4	5	6	7	8	
Device control word	6040	6840	7040	7840	8040	8840	9040	9840	46
Device status word	6041	6841	7041	7841	8041	8841	9041	9841	47
Device mode (Command value mode)	6042	6842	7042	7842	8042	8842	9042	9842	47
*Actual value input mode	2200	2900	3000	3700					49
*Speed positive	2400	2B00	3200	3900					49
*Speed negative	2401	2B01	3201	3901					50
Device control mode (Control mode)	6043	6843	7043	7843	8043	8843	9043	9843	50
Device local (Operating mode)	604F	684F	704F	784F	804F	884F	904F	984F	51
Device version	6050	6050	6050	6050	6050	6050	6050	6050	51
Device serial number	6052	6052	6052	6052	6052	6052	6052	6052	51
Device model description	6054	6054	6054	6054	6054	6054	6054	6054	51
Device vendor name	6057	6057	6057	6057	6057	6057	6057	6057	52

Signals:

Parameter		Channel number								Page
		1	2	3	4	5	6	7	8	
Command value	vpc	6300	6B00	7300	7B00	8300	8B00	9300	9B00	55
	vprc (openloop)	6380	6B80	7380	7B80	8380	8B80	9380	9B80	
	*vprc (closeloop)	6380	6B80	7380	7B80					
	*dcol	6480	6C80	7480	7C80					
	*dsc	6500	6D00	7500	7D00					
	*dpc	6600	6E00	7600	7E00					
	*n-point controller	2280	2300	2380	2400					55
Actual value input 16 bits	*all	2201	2901	3001	3701					56
Actual value input 32 bits	*all	2202	2902	3002	3702					56
*Actual value	vprc	6381	6B81	7381	7B81					56
	dsc	6501	6D01	7501	7D01					
	dpc	6601	6E01	7601	7E01					
*Control deviation	vprc	63D0	6BD0	73D0	7BD0					57
	dsc	6550	6D50	7550	7D50					
	dpc	6650	6E50	7650	7E50					

* MD2-Controller only

5.4.4 Device Type

Indicates the sub-assembly type code of the device specified in the device profile (DSP-408).

Object description

Object number	1000h
Variable name	Device Type
Object code	VAR (7h)
Data type index	Unsigned32 (7h)
Length	4

Values-Description

Sub-Index	0
Description	Code of the device type [Read Only] Bit 0-15 = Device profile number (408 dec) Bit 16-30 = reserved Bit 31 = 1 (modular device) Value: 10000198h
Prescribed range	Unsigned32

5.4.5 Error register (Flag)

This object is an error register for the device. The device can map internal errors to this byte. The object is a part of the emergency object. These bits are fixed assigned by the standard!

Object description

Object number	1001h
Variable name	Error register
Object code	VAR (7h)
Data type index	Unsigned8 (5h)

Values-Description

Sub-index	0
Description	Error register [Read Only] Bit 0 = General error Bit 1 = Current Bit 2 = Voltage Bit 3 = Temperature Bit 4 = Communication error Bit 5 = Device profile-specific Bit 6 = Reserved (value always 0) Bit 7 = Manufacturer-specific
Prescribed range	Unsigned8

5.4.6 Predefined Error Field

This object contains the occurred errors of the device, which have been indicated through the emergency object. It therefore contains a history of errors.

1. The entry in sub-index 0 indicates the number of the current errors, which have been recorded in the list. This list starts with the sub-index 1 of this object.
2. Every new error is saved in the sub-index 1, the older errors are moved down by one place on the list.
3. If a "0" is written to the sub-index 0, then all history entries are deleted.
4. The error numbers are of the type UNSIGNED32 and they are composed of a 16-bit error code and a 16-bit field with additional error information. The additional error information is manufacturer-specific error information and it is located in the higher 2 bytes (MSB). The error code is located in the lower 2

bytes (LSB). If the object is supported, then it has to consist of at least two entries: the length entry in the sub-index 0 as well as at least one error entry in the sub-index 1.

Object description

Object number	1003h
Variable name	Predefined Error Field
Object code	ARRAY (8h)
Data type index	Unsigned32 (7h)

Values-Description

Sub – Index	00h
Description	Number of errors [Read/Write]
Default Value	0
Prescribed range	0 - 254

Sub – Index	01h
Description	Standard error field [Read Only]
Default Value	none
Prescribed range	Unsigned32

Sub – Index	02h - FEh
Description	Standard error field [Read Only]
Default Value	none
Prescribed range	Unsigned32

5.4.7 COB-ID SYNC

This object defines the COB-ID of the "SYNC" object (SYNC).

Object description

Object number	1005h
Variable name	COB-ID SYNC message
Object code	VAR (7h)
Data type index	Unsigned32 (7h)
Length	4

Values-Description

Description	COB-ID used by the SYNC [Read / Write]
Prescribed range	Unsigned32
Default Value	80h

5.4.8 Guard Time

The guard time together with the life time factor defines the cycle time for the life guarding protocol, in case of the value 0, the life guarding is switched-off. The time is indicated in ms.

This object can only be changed if the producer heartbeat time is set to 0. If not, the device send a abort SDO transfer (abort code: 060A 0023h).

Object description

Object number	100Ch
Variable name	Guard Time
Object code	VAR (7h)
Data type index	Unsigned16 (6h)
Length	2

Values-Description

Sub-Index	0
Description	Value in ms
Prescribed range	Unsigned16

5.4.9 Life Time Factor

The life time factor multiplied with the guard time results in the cycle time for the life guarding protocol, in the case of the 0, the life guarding is switched-off.

This object can only be changed if the producer heartbeat time is set to 0. If not, the device send a abort SDO transfer (abort code: 060A 0023h).

Object description

Object number	100Dh
Variable name	Life Time Factor
Object code	VAR (7h)
Data type index	Unsigned8 (5h)

Values-Description

Sub-Index	0
Description	Value as multiplier for the guard time
Prescribed range	Unsigned8

5.4.10 Save Parameter

Through this object, the changed parameters can be saved in the EEPROM of the MD2. Through sub-index 1, all parameters can be saved. In order to save more selectively, the parameters in addition have been classified in groups:

1. Communication parameters (sub-index 2): This group comprises the communication-specific parameters in the object directory range 1000h...1FFFh. Saved here are the following parameters:

Parameter	Index	Sub-Index
Communication parameter RxPDO1	1400h	01h, 02h
Communication parameter RxPDO2	1401h	01h, 02h
Communication parameter TxPDO1	1800h	01h, 02h
COB-ID SYNC object	1005h	00h
Guard time	100Ch	00h
Life time factor	100Dh	00h
COB-ID Emergency object	1014h	00h
Producer heartbeat time	1017h	00h

2. Application parameters (sub-index 3): this group comprises the parameters in the object directory range 6000h...9FFFh as well as the manufacturer-specific parameters 2000h...5FFFh.

In order to prevent an inadvertent saving, a certain signature ("save") has to be written to the respective sub-index. In the hex-code, the 32-bit signature has the following appearance:

Signature	MSB			LSB	
ASCII	e	v	a	s	
hex	65h	76h	61h	73h	

If a correct signature is received in the corresponding sub-index, the device saves the parameters and confirms this with an SDO response (initiate download response). If the saving operation has failed, the device responds with an Abort SDO Transfer (abort code: 0606 0000h).

In the case of a wrong signature, the device refuses to save and responds with an Abort SDO Transfer (abort code: 0800 002xh)

In the case of a read access to a sub-index, the subassembly with the value 1h indicates, that the respective parameter group can be selectively saved.

The saved values remain valid after the Reset or Power-On of the device. Through the object 1011h they can be reset to the default values.

Object description

Object number	1010h
Variable name	Save Parameter
Object code	ARRAY (88h)
Data type index	Unsigned8 (5h)

Values-Description

Sub-Index	00h
Description	Number of elements (sub-indexes) [Read Only]
Prescribed range	Unsigned8
Prescribed range	none

Sub-Index	01h
Description	Saving of all parameters (through signature "save") [Read/Write]
Prescribed range	Unsigned32
Prescribed range	none

Sub-Index	02h
Description	Save communication parameters (through signature "save") [Read/Write]
Prescribed range	Unsigned32
Prescribed range	none

Sub-Index	03h
Description	Save application parameters (through signature "save") [Read/Write]
Prescribed range	Unsigned32
Prescribed range	none

5.4.11 Restore Defaults

With this object, the device parameters can be reset to default values in accordance with DS301, resp. DSP 408 and WAG-specific values. Through sub-index 1, all parameters can be set to default values. In order to be able to load more selectively, the parameters additionally have been classified in groups:

1. Communication parameters (sub-index 2): this group comprises the communication-specific parameters in the object directory range 1000h...1FFFh. The list of the parameters can be found at object 1010h.
2. Application parameters (sub-index 3): this group comprises the parameters in the object directory range 6000h...9FFFh as well as the manufacturer-specific parameters 2000h...5FFFh.

In order to prevent an inadvertent resetting, a certain signature ("load") has to be written to the respective sub-index. In the Hex-Code, the 32-bit signature has the following appearance:

Signature	MSB			LSB
ASCII	d	a	o	l
hex	64h	61h	6Fh	6Ch

If a correct signature is received in the corresponding sub-index, the device restores the appropriate parameters and confirms this with an SDO response (initiate download response). If the restoring has failed, then the device responds with an Abort SDO Transfer (abort code: 0606 0000h).

In case of a wrong signature, the device refuses to restore and responds with an Abort SDO Transfer (abort code: 0800 002xh)

The default values are set valid after the device is reset by a "communication reset". After that the parameters have to be stored to keep them after a power-on.

In case of a read access, the sub-assembly with the value 1h indicates, that the respective parameter group can be selectively reset.

In order to make the default values valid, a reset has to be carried out. If the default values are to be saved, then after the reset a "Save" command has to be transmitted via object 1010h.

Object description

Object number	1011h
Variable name	Restore Defaults
Object code	ARRAY (8h)
Data type index	Unsigned8 (5h)

Values-Description

Sub-index	00h
Description	Number of elements (sub-indexes) [Read Only]
Prescribed range	Unsigned8
Prescribed range	none

Sub-index	01h
Description	Restore all default parameters (through signature "load") [Read/Write]
Prescribed range	Unsigned32
Prescribed range	none

Sub-index	02h
Description	Restore communication default parameters (through signature "load") [Read/Write]
Prescribed range	Unsigned32
Prescribed range	none

Sub-index	03h
Description	Restore application default parameters (through signature "load") [Read/Write]
Prescribed range	Unsigned32
Prescribed range	none

5.4.12 COB-ID Emergency

This object defines the COB-ID of the "Emergency" object (EMCY).

Description COB-ID:

	MSB				LSB	
Bits	31	30	29	28-11	10-0	
11-bit-ID	0 / 1	0 / 1	0	0 0	11-Bit Identifier	
29-bit-ID	0 / 1	0 / 1	1	29-Bit Identifier		

Object description

Object number	1014h
Variable name	COB-ID Emergency Object
Object code	VAR (7h)
Data type index	Unsigned32 (7h)
Length	4

Values-Description

Description	COB-ID by the EMCY [Read / Write]
Prescribed range	Unsigned32
Default Value	80h + Node-ID

5.4.13 Producer Heartbeat Time

The producer heartbeat time defines the cycle time for the heartbeat, in case of the value 0, the heartbeat is switched-off.

Object description

Object number	1017h
Variable name	Producer Heartbeat Time
Object code	VAR (7h)
Data type index	Unsigned16 (6h)
Length	2

Values-Description

Description	Value in ms
Prescribed range	Unsigned16

5.4.14 Identity Object

The object contains general information about the device. The sub-index 01 contains the Vendor ID. This is a unique value allocated to each manufacturer.

Object description

Object number	1018h
Variable name	Identity Object
Object code	RECORD (9h)
Data type index	IDENTITY (23h)
Length	4

Values-Description

Sub-Index	00h
Description	Number of entries [Read Only]
Prescribed range	Unsigned8 (1 to 4)
Default Value	1

Sub-Index	01h
Description	Manufacturer identification (CiA Vendor-ID) [Read only]
Prescribed range	Unsigned32
Default Value	0x0000014D

Sub-Index	02h
Description	Product code [Read Only]
Prescribed range	Unsigned32
Default Value	none (this value is not used)

Sub-Index	03h
Description	Revision number [Read Only]
Prescribed range	Unsigned32
Default Value	none

Sub-Index	04h
Description	Serial number [Read Only]
Prescribed range	Unsigned32
Default Value	none

5.4.15 Communication parameter RxPDO1 – RxPDO8

Contains the communication parameters from Receive PDO1 (object 1400) to PDO8 (object 1407). The communication parameters can be changed in any way required by the user. The setting can also be made through the PASO.

Description COB-ID:

	MSB				LSB
Bits	31	30	29	28-11	10-0
11-bit-ID	0 / 1	0 / 1	0	0 0	11-Bit Identifier
29-bit-ID	0 / 1	0 / 1	1	29-Bit Identifier	

Bit Number	Value	Description
31 (MSB)	0	PDO active
	1	PDO not active
30	0	RTR to this PDO permitted
	1	No RTR to this PDO permitted
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID (CAN 2.0B)
28-11	0	If bit 29=0
	X	If bit 29=1: bits 28-11 of the 29-bit-COB_IDs
10-0 (LSB)	X	Bits 10-0 of the COB-ID

The transmission type (refer to sub-index 02) defines the transmission characteristics of the corresponding process data object. The significance is explained in chapter "PDO – Communication Parameters".

Object description

Object number	1400h – 1407h
Variable name	RxPDO1-8 communications parameter
Object code	RECORD (9h)
Data type index	PDO CommPar (20h)

Values-Description

Sub-index	00h
Description	Number of the communication parameter [Read only]
Prescribed range	Unsigned8
Default Value	2

Sub-index	01h
Description	COB-ID used by the PDO [Read only]
Prescribed range	Unsigned32
Default Value	384 + Node-ID (RxPDO1), 640 + Node-ID (RxPDO2) 896 + Node-ID (RxPDO3), 1152 + Node-ID (RxPDO4)

Sub-index	02h
Description	Transmission type [Read / Write] For details refer to "PDO Communication Parameters" on page 13.
Prescribed range	Unsigned 8
Default Value	255

5.4.16 Mapping RxPDO1 – RxPDO8

Receive PDO Mapping. The MD2 electronics supports dynamic PDO mapping. The mapping of the receive data can be set with the object 1600 to 1607 and through the PASO.

To set the mapping objects, the following information about the mapped object are required:

- object index (4 Byte)
- object sub-index (2 Byte)
- object size in Bit (2Byte)

These values are put together to a hex-value.

Example for object 6040 (Device control word channel 1):

- object index: 0x6040
- object sub-index : 0x00
- object size: 16 [Dez] → 0x10 [Hex]

As a result the value 0x60400010 is the value for the mapping entry.

For changing the mapping, first the PDO must be blocked (refer to "Communication parameter RxPDO1 – RxPDO" on page 40) and the number of mapping (sub-index 0) entries must be set to 0.

The following tabel shows the standard mapping.
The RxPDO 5-8 are deactivated as standard.

PDO	Object 1	Object 2	Object 3	Object 4	Transmission type	Device control mode
RxPDO1	Controllword Channel 1	Command Value VPOC Channel 1	-	-	255	1, 3
RxPDO2	Controllword Channel 2	Command Value VPOC Channel 2	-	-	255	1, 3
RxPDO3	Controllword Channel 3	Command Value VPOC Channel 3	-	-	255	1, 3
RxPDO4	Controllword Channel 4	Command Value VPOC Channel 4	-	-	255	1, 3

Object description

Object number	1600-1607h
---------------	------------

Variable name	RxPDO1 – RxPDO8 Mapping
Object code	RECORD (9h)
Data type index	PDO mapping parameter (21h)

Values-Description

Sub-index	00h
Description	Number of mapped objects [Read/Write]
Prescribed range	Unsigned8 (0 to 64)
Default Value	Refer to above table

Sub-index	01h
Description	1. mapped object [Read/Write]
Prescribed range	Unsigned32
Default Value	Refer to above table

Sub-index	02h
Description	2. mapped object [Read/Write]
Prescribed range	Unsigned32
Default Value	Refer to above table

Sub-index	03h
Description	3. mapped object [Read/Write]
Prescribed range	Unsigned32
Default Value	Refer to above table

Sub-index	04h
Description	4. mapped object [Read/Write]
Prescribed range	Unsigned32
Default Value	Refer to above table

5.4.17 Communication parameter TxPDO1 – TxPDO8

Contains the communication parameters from Transmit PDO1 (object 1800) to PDO8 (object 1807). The communication parameters can be changed in any way required by the user. The setting can also be made through the PASO.

Description COB-ID:

	MSB				LSB
Bits	31	30	29	28-11	10-0
11-bit-ID	0 / 1	0 / 1	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11-Bit Identifier
29-bit-ID	0 / 1	0 / 1	1	29-Bit Identifier	

Bit Number	Value	Description
31 (MSB)	0	PDO active
	1	PDO not active
30	0	RTR to this PDO permitted
	1	No RTR to this PDO permitted
29	0	11-bit ID (CAN 2.0A)
	1	29-bit ID(CAN 2.0B)
28-11	0	If bit 29=0
	X	If bit 29=1: bits 28-11 of the 29-bit-COB_IDs
10-0 (LSB)	X	Bits 10-0 of the COB-ID

The transmission type (refer to sub-index 02) defines the transmission characteristics of the corresponding process data object. The significance is explained in chapter "PDO – Communication Parameters".

Object description

Object number	1800h – 1807h
Variable name	TxPDO1-8 communications parameter
Object code	RECORD (9h)
Data type index	PDO CommPar (20h)

Values-Description

Sub-index	00h
Description	Number of the communication parameter [Read only]
Prescribed range	Unsigned8
Default Value	2

Sub-index	01h
Description	COB-ID used by the PDO [Read / Write]
Prescribed range	Unsigned32
Default Value	384 + Node-ID (TxPDO1), 640 + Node-ID (TxPDO2) 896 + Node-ID (TxPDO3), 1152 + Node-ID (TxPDO4)

Sub-index	02h
Description	Transmission type [Read / Write] For details refer to "PDO Communication Parameters" on page 13.
Prescribed range	Unsigned 8
Default Value	255

5.4.18 Mapping TxPDO1 – TxPDO8

Transmit PDO Mapping. The MD2 electronics supports dynamic PDO mapping. The mapping of the receive data can be set with the object 1600 to 1607 and through the PASO.

To set the mapping objects, the following information about the mapped object are required:

- object index (4 Byte)
- object sub-index (2 Byte)
- object size in Bit (2Byte)

These values are put together to a hex-value.

Example for object 6040 (Device control word channel 1):

- object index: 0x6040
- object sub-index : 0x00
- object size: 16 [Dez] → 0x10 [Hex]

As a result the value 0x60400010 is the value for the mapping entry.

For changing the mapping, first the PDO must be blocked (refer to "Communication parameter TxPDO1 – TxPDO8" on page 42) and the number of mapping (sub-index 0) entries must be set to 0.

The following table shows the standard mapping.
 The TxPDO 5-8 are deactivated as standard.

PDO	Object 1	Object 2	Object 3	Transmission type	Device control mode
TxPDO1	Statusword Channel 1	-	-	255	1, 3
TxPDO2	Statusword Channel 2	-	-	255	1, 3
TxPDO3	Statusword Channel 3	-	-	255	1, 3
TxPDO4	Statusword Channel 4	-	-	255	1, 3

Object description

Object number	1A00h - 1A07h
Variable name	TxPDO1 – TxPDO8 Mapping
Object code	RECORD (9h)
Data type index	PDO Mapping parameter (21h)

Werte-Beschreibung

Sub-index	00h
Description	Number of mapped objects [Read/Write]
Prescribed range	Unsigned8 (0 to 64)
Default Value	Refer to above table

Sub-index	01h
Description	1. mapped object [Read/Write]
Prescribed range	Unsigned32
Default Value	Refer to above table

Sub-index	02h
Description	2. mapped object [Read/Write]
Prescribed range	Unsigned32
Default Value	Refer to above table

Sub-index	03h
Description	3. mapped object [Read/Write]
Prescribed range	Unsigned32
Default Value	Refer to above table

Sub-index	04h
Description	4. mapped object [Read/Write]
Prescribed range	Unsigned32
Default Value	Refer to above table

5.4.19 Device control word

The control word is bit coded, i.e., each individual bit has a certain control function. The table below lists the individual functions with the bit belonging to it.

MSB								LSB							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High - Byte								Low - Byte							

Bit	Name	Description
0	Disable (D)	These bits form the device control commands. Refer to the description of the device state machine.
1	Hold enable (H)	
2	Device mode active (M)	
3	Reset fault (R)	Resets an error/fault
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	
8	Reserved	
9	Reserved	
10	Reserved	
11	Reserved	
12	Reserved	
13	Reserved	
14	Reserved	
15	Manufacturer-specific	

Object description

Object number	6040h, 6840h, 7040h, 7840h, 8040h, 8840h, 9040h, 9840h
Variable name	Device control word
Object code	VAR (7h)
Data type index	Unsigned16 (6h)
Length	2

Values-Description

Description	Default value: 00'00h [read / write]
Prescribed range	0...65535

5.4.20 Device status word

The control word is bit coded, i.e., each individual bit has a status display function. The table below lists the individual functions with the bit belonging to it.

MSB								LSB							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
High - Byte								Low - Byte							

Bit	Name	Description
0	Disable (D)	These bits determine the device condition. Refer to the description of the device state machine.
1	Hold enable (H)	
2	Device mode active (M)	
3	Ready (R)	
4	Local control	Is active, if the MD2 is operated locally
5	Warning	
6	Reserved	
7	Reserved	
8	Reserved	
9	Ramp running	The command value ramp is active (only in device mode 1 and 3)
10	Reserved	
11	Reserved	
12	Window reached	The target window is reached (only in device mode 4, 7 and 9)
13	Reserved	
14	Reserved	
15	Manufacturer-specific	

Object description

Object number	6041h, 6841h, 7041h, 7841h, 8041h, 8841h, 9041h, 9841h
Variable name	Device status word
Object code	VAR (7h)
Data type index	Unsigned16 (6h)

Values-Description

Description	Default value: 00'00h [Read Only]
Prescribed range	0...65535

5.4.21 Device mode (Command value mode)

The object specifies the source for the command value. It can come via CAN or from a local source (PASO, analog-/digital-/PWM-input).

This object corresponds to the PASO-parameter "Command value mode" in "Signal Scaling".

Value	Description
1	Command value via CAN
2	Command value comes from a local source

Object description

Object number	6042h, 6842h, 7042h, 7842h, 8042h, 8842h, 9042h, 9842h
Variable name	Device mode
Object code	VAR (7h)
Data type index	Integer8 (2h)

Values-Description

Description	[read / Write]
-------------	----------------

Prescribed range	1 or 2
------------------	--------

5.4.22 Device mode (actual value mode)

The object specifies the source for the command value. It can come via CAN or from a local source (PASO, analog-/digital-/PWM-input).

This object corresponds to the PASO-parameter "Command value mode" in "Command Scaling".

Value	Description
1	Command value via CAN
2	Command value comes from a local source

Object description

Object number	6042h, 6842h, 7042h, 7842h, 8042h, 8842h, 9042h, 9842h
Variable name	Device mode
Object code	VAR (7h)
Data type index	Integer8 (2h)

Values-Description

Description	[read / write]
Prescribed range	1 or 2

5.4.23 Actual value mode (Controller only)

The object specifies the source for the actual value. It can come via CAN or from a local source (PASO, analog-/digital-/PWM-input).

This object corresponds to the PASO-parameter "Feedback value mode" in "Feedback Scaling".

Value	Description
1	Actual value via CAN
2	Actual value comes from a local source

Objekt-Beschreibung

Object description	See object dictionary page 32
Object number	Actual value mode
Variable name	VAR (7h)
Object code	Integer8 (2h)

Values-Description

Description	[read / write]
Prescribed range	1 or 2

5.4.24 Speed positive

Corresponds to the "Speed +" parameter in PASO.

Object description

Object number	See object dictionary page 32
Variable name	Demand value positive velocity
Object code	VAR (7h)
Data type index	Value parameter record Unsigned32 (82h)

Values-Description

Sub-index	00h
Description	Value [Read/Write]
Default Value	0
Prescribed range	refer to section "Scaled parameter" page 31

5.4.25 Speed negative

Corresponds to the "Speed -" parameter in PASO.

Object description

Object number	See object dictionary page 32
Variable name	Demand value negative velocity
Object code	VAR (7h)
Data type index	Integer32 (4h)

Values-Description

Sub-index	00h
Description	Value [Read/Write]
Default Value	0
Prescribed range	refer to section "Scaled parameter" page 31

5.4.26 Device control mode (Control mode)

With this parameter the control mode of the device is indicated or switched. The object is read/write, if switching between different control modes is supported, otherwise read only. Supported control modes are indicated in object 605Fh (Device Capability) by associated bits enabled (1).

The value can only be changed in device state "Init" or "Disabled", otherwise the device responds with an Abort SDO transfer (abort code: 0800 0022h).

NOTE: If the control mode has changed, the PDO's must be mapped new. This will be done automatically after a reset resp. if the device will be switched off an on (Attention: save the parameter control mode before).

Value	Description	Kind
0	The MD2 device is waiting for a control mode	
1	Spool position control open loop vpoc	open loop
3	Pressure control valve open loop vprc	open loop
4	Pressure control valve closed loop vprc	closed loop
6	Open loop movement dcol	open loop
7	Velocity control axis dsc	closed loop
9	Position control axis dpc	closed loop
-5	Pressure control valve 2-solenoids closed loop vprc	closed loop
-6	2-point controller 1-solenoid	closed loop
-7	2-point controller 2-solenoids	closed loop
-8	3-point controller 2-solenoids	closed loop

Object description

Object number	6043h, 6843h, 7043h, 7843h, 8043h, 8843h, 9043h, 9843h
Variable name	Device control mode (Control mode)
Object code	VAR (7h)
Data type index	Integer8 (2h)

Values-Description

Description	Default value: 00h [Read Only]
Prescribed range	0, 1, 3, 4, 6, 7, 9, -5, -6, -7, -8

5.4.27 Device local (Operating mode)

The object specifies the source for the control word. It can come via CAN or from a local source. This object corresponds to the PASO-parameter "Operating mode" in "Enable Channel".

If a local operation is requested by a different commander, e.g. PASO, then this object becomes irrelevant; a write request to this object is "rejected".

Value	Description
0	Operation via CAN
1	Control word comes from a local operation

Object description

Object number	604Fh, 684Fh, 704Fh, 784Fh, 804Fh, 884Fh, 904Fh, 984Fh
Variable name	Device local (Operating mode)
Object code	VAR (7h)
Data type index	Integer8 (2h)

Values-Description

Description	[read / Write]
Prescribed range	0 or 1

5.4.28 Device version

Device version; corresponds to the PASO field Software Version.

Object description

Object number	6050h
Variable name	Device version
Object code	VAR (7h)
Data type index	Visible String (9h)

Values-Description

Description	"V1.0.0.0" [Read Only]
Prescribed range	ASCII values for numerals and numbers

5.4.29 Device serial number

Device serial number; corresponds to the PASO field Serial Number.

Object description

Object number	6052h
Variable name	Device serial number
Object code	VAR (7h)
Data type index	Visible String (9h)

Values-Description

Description	"123.4567" [Read Only]
Prescribed range	ASCII values for numerals and numbers

5.4.30 Device model description

Valve type description, corresponds to the PASO field Valve Type.

Object description

Object number	6054h
Variable name	Device model description

Object code	VAR (7h)
Data type index	Visible String (9h)

Values-Description

Description	"Valve type" [Read Only]
Prescribed range	ASCII values for numerals and numbers

5.4.31 Device vendor name

Names the manufacturer of the device.

Object description

Object number	6057h
Variable name	Device vendor name
Object code	VAR (7h)
Data type index	Visible String (9h)

Values-Description

Description	"Wandfluh AG" [Read Only]
Prescribed range	ASCII values for numerals and numbers

5.4.32 Device capability

This object contains information about the applications capability of the device.

Object description

Object number	605F, 685F, 705F, 785F, 805F, 885F, 905F, 985F
Variable name	Device capability
Object code	VAR (7h)
Data type index	Unsigned32 (7h)

Values-Description

Description	Default value: depending on the valve [Read only] Value 0 = disabled / not supported Value 1 = enabled / supported Bit 0-13 = Reserved Bit 14 = n-point control (Wandfluh specific) Bit 15 = Manufacturer specific Bit 16 = Hydraulic drive Bit 17 = Open loop movement Bit 18 = velocity-control Bit 19 = P/Q control Bit 20 = Position-control Bit 21-23 = Reserved Bit 24 = Hydraulic proportional valve Bit 25 = Spool position control open loop (without LVDT) Bit 26 = Spool position control closed loop (with LVDT) Bit 27 = Pressure control valve open loop (without feedback sensor) Bit 28 = Pressure control valve closed loop (with feedback sensor) Bit 29 = P/Q Valve Bit 30 = Reserved Bit 31 = Modular device (can have various functions)
Prescribed range	Unsigned32

5.4.33 Device temperature

The current temperature inside the device. Corresponds to the " MD2-Temperature" in the PASO Menu "Analysis - Values".

Object description

Object number	2002h
Variable name	device temperature
Object code	VAR (7h)
Data type index	Integer16 (3h)

Values-Description

Description	current Temperature in °C [Read Only]
Prescribed range	-55 ... +150

5.4.34 Device node address

The MD2 node address can be set via CAN. The address is set valid after the device is reset by a "communication reset" or a power cycle. The parameter has to be stored to keep after a power-on.

Value	Description
1...127	MD2 node address

Object description

Object number	2000h
Variable name	device node address
Object Code	VAR (7h)
Data type index	Unsigned8 (5h)

Values description

Description	Default value: 01h [Read/Write]
Prescribed range	1...127

5.4.35 Device node baudrate

The MD2 node baudrate can be set via CAN. The baudrate is set valid after the device is reset by a "communication reset" or a power cycle. The parameter has to be stored to keep after a power-on.

Value	Description
10	10 kBaud
20	20 kBaud
50	50 kBaud
100	100 kBaud
125	125 kBaud
250	250 kBaud
500	500 kBaud
1000	1000 kBaud

Baudrates outside the above list causes an Abort SDO Transfer.

Object description

Object number	2001h
Variable name	Device node baudrate
Object Code	VAR (7h)

Data type index	Unsigned16 (6h)
-----------------	-----------------

Values description

Description	Default value: 14h [Read/Write]
Prescribed range	0...65535

5.4.36 Command value

Corresponds to the command value for the device.

Object description

Object number	See object dictionary page 32
Variable name	Control Mode 1: vpoc_setpoint Control Mode 3: vprc_setpoint Control Mode 4: vprc_setpoint Control Mode 6: dcol_setpoint Control Mode 7: dsc_setpoint Control Mode 9: dpc_setpoint Control Mode -5: vprc_setpoint
Object code	RECORD (9h)
Data type index	Control Mode 1, 3, 4, -5: Value parameter record Integer16 (84h) Control Mode 6, 7, 9 Value parameter record Integer32 (85h)

Values-Description

Sub-index	00h
Description	Number of entries [Read Only]
Prescribed range	0...255

Sub-index	01h
Description	Value [Read/Write]
Default Value	0
Prescribed range	Control Mode 1, 3, 4, -5: -16384...+16384 (refer to section "Scaled parameter" page 31) Control mode 6, 7, 9: Min-Interface...Max-Interface Setpoint

5.4.37 Command value for n-point control (Controller only)

Corresponds to the n-point command value for the device.

Object description

Object number	See object dictionary page 33
Variable name	n-point controller set point
Object code	VAR (07h)
Data type index	INTEGER32 (04h)

Values-Description

Sub-index	00h
Description	Value [Read/Write]
Default Value	0
Prescribed range	Min-Interface...Max-Interface Setpoint

5.4.38 Actual value input 16 bits (MD2-controller only)

Entspricht dem Istwerteingang.

Objekt-Beschreibung

Object number	See object dictionary page 33
Variable name	Actual value input 16 bits
Object code	VAR (7h)
Data type index	INTEGER16 (03h)

Sub-index	00h
Description	Wert [Read/Write]
Default Value	0
Prescribed range	Min-Interface...Max-Interface command value

5.4.39 Actual value input 32 bits (MD2-controller only)

Entspricht dem Istwerteingang.

Objekt-Beschreibung

Object number	See object dictionary page 33
Variable name	Actual value input 32 bits
Object code	VAR (7h)
Data type index	INTEGER32 (04h)

Sub-index	00h
Description	Wert [Read/Write]
Default Value	0
Prescribed range	Min-Interface...Max-Interface command value

5.4.40 Actual value (Controller only)

Corresponds to the actual value from the device.

Object description

Object number	See object dictionary page 33
Variable name	Control Mode 1, 3, 6: - Control Mode 4: vprc actual value Control Mode 7: dsc actual value Control Mode 9: dpc actual value
Object code	RECORD (9h)
Data type index	Control Mode 4: Value parameter record Integer16 (84h) Control Mode 7: Value parameter record Integer32 (85h) Control Mode 9: Value parameter record Integer32 (85h)

Values-Description

Sub-index	00h
Description	Number of entries [Read Only]
Prescribed range	0...255

Sub-index	01h
Description	Value [Read Only]
Default Value	0
Prescribed range	Control Mode 4: 0...+16384 (refer to section "Internal bus resolution" page 31)

	Control Mode 7, 9: refer to section "Scaled parameter" page 31)
--	--

5.4.41 Control deviation

Corresponds to the control deviation value from the device.

Object description

Object number	See object dictionary page 33
Variable name	Control Mode 1, 3, 6: - Control Mode 4: vprc control deviation Control Mode 7: dsc control deviation Control Mode 9: dpc control deviation
Object code	RECORD (9h)
Data type index	Control Mode 4: Value parameter record Integer16 (84h) Control Mode 7: Value parameter record Integer32 (85h) Control Mode 9: Value parameter record Integer32 (85h)

Values-Description

Sub-index	00h
Description	Number of entries [Read Only]
Prescribed range	0...255

Sub-index	01h
Description	Value [Read Only]
Default Value	0
Prescribed range	Control Mode 4: 0...+16384 (refer to section "Internal bus resolution" page 31) Control Mode 7, 9: refer to section "Scaled parameter" page 31)

Commissioning

To support the MD2-CAN commissioning, the parameterisation software PASO-MD2 can be connected. PASO offers the possibility to show certain process data like command value, valve current, device state (state machine) and so on. The bus-node settings (node address and baudrate) can also be made with PASO, as well as certain CAN-bus diagnostics (refer to section "Fieldbus Diagnostics" page 8);

5.5 Step by step instructions for the first commissioning

For the first commissioning, the following steps should be observed:

5.5.1 Test the hydraulic system

1. Switch off the hydraulic system
2. Switch off the fieldbus master
3. Switch on the MD2
4. In the PASO window "Fieldbus_Fieldbus-Info" in the section "Bus State" the following statement will be displayed: Bus-Status = Pre-Operational (refer to section "Fieldbus Diagnostics" page 8)
5. In the PASO status line, the statements "Disabled" or "Init" will be displayed
6. Switch on the hydraulic system
7. With the PASO Menu "Commands_Valve operation", the solenoids can be operated directly.

IMPORTANT: The hydraulic moves in an open loop system! Be sure, that the hydraulic system can move free.

8. In the PASO window "Parameters_Soleniods", the parameters for the minimum (Imin) and maximum (Imax) current and the dither signal (frequency and level) can be set
9. Disable the device with the PASO Menu "Commands_Disable". In the PASO status line, the statements "Remote PASO" and "Disabled" will be displayed
10. Set the control of the device to Local with the PASO Menu "Commands_Local Control". In the PASO status line, the statements "Remote" and "Init" will be displayed

5.5.2 Adjust the mode of operation

1. In the PASO window "Configuration_Mode of operation", the adjustments for the desired mode of operation can be made

5.5.3 Test the fieldbus

1. Load the EDS-file in the fieldbus master and select the desired baudrate (refer to section "Presupposition and information for the Fieldbus master" page 60)
2. Adjust the node address and the baudrate on the MD2 (refer to section "Presupposition for the DP-Slave controller card" page 59)
3. Switch on the fieldbus master
4. The MD2 can be set via the fieldbus master with the NMT-services to the status "Pre-Operational" resp. "Operational" (refer to section "Device Control Services" page 19)
5. In the PASO window "Fieldbus_Fieldbus-Info" in the section "Bus State" the following statement will be displayed: Bus-Status = Pre-Operational resp. Operational (refer to section "Fieldbus Diagnostics" page 8)

5.5.4 Test the control via the fieldbus

Set the following parameters in the declared order with the SDO-services (refer to section "

1. Service Data Communication (SDO)" page 16) (only possible in the state "DISABLE"):
2. Set the parameter "Device local (Operating mode)" to "Control Operation via CAN (0)" (refer to section "Device local (Operating mode)" page 51).
3. Set the parameter "Device mode (Command value mode) to "CAN (1)" (refer to "Device mode (Command value mode)" page 47).
4. With the parameter "Device control mode (Control mode)" the desired device control mode can be selected (refer to section "Device control mode (Control mode)" page 50).
5. For the release of the MD2, the 3 bits "Disable (D)", "Hold enable (H)" and "Device mode active (M)" from the control word (refer to section "Device control word" page 46) must be set to logical 1. The CANopen-Slave controller is now in the state "ACTIVE".
6. With the SDO-services (refer to section "Service Data Communication (SDO)" page 16) resp. the PDO-services (refer to section "Process Data Communication (PDO)" page 13). a command value can now be set via the fieldbus.

5.6 Presupposition for the DP-Slave controller card

For the commissioning of a MD2, the following presupposition must be cleared:

- **What is the node address from the MD2?**

The node address can be set via the parameterisation software PASO in the menu item "Fieldbus_Fieldbus-Info" (refer to section "Fieldbus Settings" page 7)

- **What is the Device control mode (Control mode) for the DP-Slave controller card?**

The Device control mode (Control mode) can be set via the parameter "Device control mode (Control mode)". This selection is important for the for the function range of the DP-Slave controller card.

IMPORTANT: This parameter can only be changed if the MD2 is in the state "INIT" or "DISABLE" (refer to section "Device State Machine" page 26)

5.7 Presupposition and information for the Fieldbus master

For the commissioning of a Fieldbus master, the following presupposition must be cleared:

- **Node address**
What is the node address from the MD2?
- **EDS-file**
The EDS-file "WAGMD2C1.eds" must be present on the Master side. If not, this file must be copied into the project tool of the Master.

5.8 Delivery state

The MD2 is delivered with the following basic configuration:

- Node address 1
- Baudrate 20kBaude

5.9 Settings

The MD2 settings can be read or changed through CAN-bus or PASO.

After the MD2 power on, all parameter settings can be made through an SDO access. To keep the changed settings after a power cycle, they have to be stored in the nonvolatile memory. Settings can be stored with the "save parameter" object 1010h (refer to section "Save Parameter" page 36).

5.10 CAN Bus demand value

In the standard version the valve can only be controlled through the CAN-bus or for test reasons through PASO.

The following start up sequence need to be done after each power on:

- The MD2 is in "pre-operational" state, there is only a communication with SDO- and NMT services available. A PDO or SYNC telegram is not possible, the device does not respond to it.
- Through the NMT service the MD2 can be set to state "operational" (refer to section "Device Control Services" page 19). In this state PDO's and SYNC telegrams can be transmitted and the MD2 responds accordingly.
- To release the MD2 the 3 bits "D", "H" and "M" of the statusword have to be set to logical 1 (refer to section "Device control word" page 46). The MD2 is now in state "active", a command value can now be received from the setpoint object (e.g. 6300h) or an RxPDO transmission.

5.11 Starting after an error

- After an error recognition the device goes immediately into the state disabled by removing the internal release. Because of the error, an emergency object will be sent and the "ready" bit in the statusword (is a part of the transmit PDO) goes to 0.
- To release the device again, the bit "reset fault" (R) has to be changed from 0 → 1 once to reset the error (refer to section "Device State Machine" page 26).

6 Diagnostic and error detection

6.1 Diagnostic about the Fieldbus

A diagnosis about the Fieldbus is always possible via the parameterisation software PASO. This will be made via the menu item "Fieldbus_Fieldbus-Info". The following values will be displayed:

- Bus Node Address
- Baudrate
- Bustyp
- ID-Nummer
- Bus-Status
- Controlword
- Statusword

A detailed description of the diagnostic function you will find in the section "Fieldbus Diagnostics" page 8.

7 Version Index

In the following table, an index about the different version of the " Operating Instructions for MD2 Electronics CANopen - Protocol with Device Profile DSP 408" will be listed. The current version is always the version listed at last.

Version	Bezeichnung	Datum der Freigabe
0.1	Start Version	29.07.08
1.0	Description of the parameters added	28.08.08
1.1	Description of controller specific objects added	24.06.09
1.2	Description of controller specific objects added	04.11.09