# Manual for

# DMA-22(A)

**Universal On-Board and Stand-Alone Digital Amplifier and Controller**

| | Model Code | SW Version | Remarks |
|---|---|---|---|
| **Applicable for SW Versions:** | DMA-22(A)-01-x-PN-x<br>Operation Mode: 01 | V 13.xx | xx = 01 or higher<br>* = a, b, ….<br>x = additional options |
| | DMA-22(A)-02-x-PN-x<br>Operation Mode: 02 | V 27.xx<br>V 28.xx | |
| | DMA-22(A)-x-PN-x<br>Operation Mode: 03, 04, 06, 08 | V 48.xx | |
| | DMA-22(A)-x-PN-x-SANAout<br>Operation Mode: 03, 04, 06, 08, 10, 11 | V 49.xx | |
| | DMA-22(A)-05-x-PN-x-S10Vref<br>Operation Mode: 04, 05 | V 53.xx | |
| | DMA-22(A)-03-020-PN-x-SServo<br>Operation Mode: 03, 04 Servo | V 72.xx | |




**HCS Hydraulic Control Systems GmbH**
**Neuffener Str. 29**
**D-72636 Frickenhausen**
**Germany**
**Telefon: (+49) 7025 - 911 007**
**Telefax: (+49) 7025 - 911 008**
**Email: info@h-c-s-gmbh.de**
**Web: www.h-c-s-gmbh.de**

## Revision History

| Revision | Date | Description |
|---|---|---|
| R1.0 | 14.07.2022 | First edition |
| R1.0 | 15.07.2022 | Mode 1,2 added |
| R1.0 | 24.08.2022 | Connection diagram changed |
| R1.0 | 12.09.2022 | Wiring diagrams, technical data, supply voltage range updated |
| R1.0 | 30.09.2022 | LED display, while Profibus timeout defined |
| R1.0 | 07.06.2023 | SP1..SP4 *)The designation and availability of the input depends on the module variant. |
| R1.1 | 17.01.2024 | SW Versions added. Modification in TIA Portal example |
| R1.2 | 30.04.2024 | SW version 28.xx Versions added |
| R1.3 | 20.05.2024 | The parameters E2.03 to E2.14, C1.41 added in parameter list |
| R1.4 | 08.07.2024 | Function block definition for Special SW in Mode2 |
| R1.4 | 18.07.2024 | Blinking Error LED |
| R1.5 | 10.08.2026 | E18 range for SW Version V4x.02* expanded, selection 9 added.<br>Error/Comp Output Bit for CMD15 added |
| R1.6 | 16.10.2026 | E18 = 9: description of CNTRL_4 / STAT_5 completed, TST_CMP restricted to E18 = 1…8. Load capability of the Error/Comp output added (7.7.1). Function block PN_DMA_2a V14: automatic recovery after an error response (9.4). |

## Table of Content


**1 ABBREVIATIONS ..... 5**

**2 TERMS AND DEFINITIONS ..... 5**

**3 GENERAL INFORMATION ..... 5**

3.1 ABOUT THIS MANUAL ..... 5
3.2 SCOPE ..... 6
*3.2.1 Copyright ..... 6*
*3.2.2 Documents place of storage ..... 6*
*3.2.3 Warning signs, symbols and notes ..... 6*
*3.2.4 Safety and liability ..... 7*
*3.2.5 General information ..... 8*
3.3 TERMS AND CONDITIONS ..... 9
3.4 DELIVERY STATE (DEFAULT SETTING) ..... 9
3.5 CORRECT PRODUCT USE ..... 9
3.6 NON-INTENDED PRODUCT USE ..... 9
3.7 SELECTION AND QUALIFICATION OF PERSONNEL ..... 10
*3.7.1 Inquiries and ordering ..... 10*
*3.7.2 Service and repair ..... 10*
*3.7.3 Cleaning, storage, transport ..... 10*
*3.7.4 Delivery state (default setting) ..... 11*
3.8 INTRODUCTION ..... 11
3.9 GENERAL APPLICATIONS ..... 11

**4 TECHNICAL DATA ..... 12**

4.1 TECHNICAL DATA OVERVIEW ..... 12
4.2 MODEL CODE ..... 13
4.3 FEATURES OF THE DEVICE ..... 14
4.4 USED GSDML FILE ..... 14
4.5 PROGRAMMING PC INTERFACES ..... 14
4.6 CONNECTOR DEFINITION ..... 14
*4.6.1 Connector position layout ..... 14*
4.7 MODE 1,2 POSSIBLE CONNECTOR OPTIONS (OTHER ON CUSTOMER REQUEST) ..... 14
*4.7.1 Mode 3,4,5,6,8,10,11 possible connector options (other on customer request) ..... 15*
4.8 CONNECTION DIAGRAM ..... 15
*4.8.1 Mode 1 ..... 15*
*4.8.2 Mode 2 ..... 16*

*4.8.3 Mode 3, 4 – SW V 72.xx Servo valve ... 16*
*4.8.4 Mode 3, 4 ... 17*
*4.8.5 Mode 5 ... 18*
*4.8.6 Mode 6 ... 18*
*4.8.7 Mode 8 ... 19*
*4.8.8 Mode 10 ... 19*
*4.8.9 Mode 11 ... 20*
4.9 DEVICE CONTROLLER STRUCTURE ... 20
*4.9.1 Block diagram ... 20*
4.9.1.1 Option "analog output" -SANAout ... 20
4.9.1.2 Mode 1, Open loop, 1 valve ... 21
4.9.1.3 Mode 2, Open loop, 2 valves ... 21
4.9.1.4 Mode 3 Closed loop, 1 valve ... 22
4.9.1.5 Mode 3 -SServo Closed loop, 1 valve ... 23
4.9.1.6 Mode 4, Closed loop, 1 process ... 24
4.9.1.7 Mode 4 -SServo, Closed loop, 1 process ... 25
4.9.1.8 Mode 5, Closed loop, 1 process ... 26
4.9.1.9 Mode 6, Closed loop, 1 valve, 1 process ... 27
4.9.1.10 Mode 8, Closed loop, 2 processes ... 28
4.9.1.11 Mode 10 -SANAout, single closed loop – no valve ... 29
4.9.1.12 Mode 11 -SANAout, double closed loop – no valve ... 30
4.10 INPUTS ... 31
4.11 OUTPUTS: ... 31
4.12 FRONT VIEW: ADDRESS SELECTION AND CONNECTORS ... 31
*4.12.1 Single module Version (node with one module connected) ... 31*
*4.12.2 Multiple module Version (node with multiple modules connected) ... 32*

**5 GENERAL COMMUNICATION INFORMATION ... 32**

5.1 SPECIAL INFORMATION FOR MULTI-SLAVE NODES ... 32
5.2 GENERAL INFORMATION ABOUT TADR (TELEGRAM ADDRESS) ... 33
5.3 GENERAL INFORMATION ABOUT CMD (COMMAND) AND SADR (SLAVE ADDRESS) ... 33

**6 SIMPLE COMMANDS ... 34**

6.1 CMD 3 MASTER READ ... 34
*6.1.1 CMD = 3, Master Read parameters (7 bytes) ... 34*
*6.1.2 CMD = 3, Response from Slave (4 + CNT bytes) ... 34*
*6.1.3 CMD = 3, Error from Slave (4 bytes) ... 35*
6.2 TELEGRAM EXAMPLE CMD = 3 ... 35
6.3 CMD = 6, MASTER WRITE ... 36
*6.3.1 CMD = 6, Master Write single parameter (7 bytes) ... 36*
*6.3.2 CMD = 6, Response from Slave (7 bytes) ... 36*
*6.3.3 CMD = 6, Error from Slave (4 bytes) ... 37*
6.4 TELEGRAM EXAMPLE CMD = 6 ... 37
*6.4.1 Example, write parameter ... 37*
*6.4.2 Example, write set value ... 38*

**7 COMPLEX COMMANDS ... 39**

7.1 CMD = 15, MASTER WRITES MULTIPLE PARAMETERS (3+5·MODULE BYTES) ... 39
7.2 CMD = 15, RESPONSE FROM SLAVE (9 BYTES) ... 40
7.3 CMD = 15, ERROR FROM SLAVE (4 BYTES) ... 40
7.4 EXAMPLES, CMD = 15 ... 41
*7.4.1 Single slave ... 41*
*7.4.2 Multiple-slave (3 DMA modules) ... 41*
7.5 CMD = 15, STRUCTURE OF PROFINET TELEGRAM DEFINITION, DEPENDING ON SW VERSIONS ... 43
*7.5.1 General ... 43*
7.5.1.1 Error Message (indication by „Error occurred") ... 43
7.5.1.2 Explanation for „Hardware Enable" and „Software Disable": ... 43
7.5.1.3 Explanation of special functions and module state bits ... 44
*7.5.2 Version for Mode 1 (open loop, one valve with two solenoids) ... 45*

*7.5.3 Version for Mode 2 (open loop, two valves with one solenoid each) ... 46*
*7.5.4 Version for Mode 2 with separate enable (open loop, two valves with one solenoid each) ... 47*
*7.5.5 Version for Mode 3, 4, 10 (closed loop) ... 48*
*7.5.6 Version for Mode 6, 11 (two closed loops) ... 49*
*7.5.7 Version for Mode 8 (closed loop) ... 50*
7.6 DIGITAL INPUTS ... 51
*7.6.1 Digital input 0 (enable signal) input 1,2 (set Value selection) ... 51*
7.7 DIGITAL OUTPUTS ... 52
*7.7.1 Digital output: \Error or Comparator output ... 52*
*7.7.2 Internal supply output ... 53*
*7.7.3 Analog voltage output ... 53*
7.8 STATUS DISPLAY LEDS ... 54
*7.8.1 LED definition ... 54*
*7.8.2 LED status definition ... 54*

**8 PLC CONFIGURATOR ... 55**

**9 CONFIGURATION FOR SIEMENS TIA PORTAL ... 56**

9.1 SETUP THE NETWORK CONFIGURATION FOR THE DMA MODULE ... 56
9.2 ASSIGN THE IO-MODULE FROM THE GSD TO THE DEVICE ... 58
9.3 INSTALL THE FUNCTION BLOCK ... 58
9.4 EXPLANATION OF THE FUNCTION BLOCK ... 59
*9.4.1 General ... 59*
*9.4.2 Classic CPU 300,400: Definition of the data buffer address ... 60*
*9.4.3 CPU 1200,1500: Specification of the data buffer address based on the hardware identifier ... 60*
*9.4.4 CMD 3, Reading values from one DMA-2 node ... 61*
*9.4.5 CMD 6, Writing Parameters to one DMA-2 node ... 62*
*9.4.6 CMD 15, Writing and Reading once to all connected DMA nodes ... 63*

**10 LIST OF PARAMETERS ... 65**

**11 TROUBLESHOOTING ... 70**

11.1 GENERAL ... 70
11.2 NO PROFINET COMMUNICATION IS POSSIBLE ... 70
11.3 COMMUNICATION OK, BUT CURRENT OUTPUTS ARE INACTIVE ... 71
11.4 PNO CERTIFICATE ... 72

***12 DECLARATION OF CONFORMITY ... 73***

***13 NOTES: ... 74***

***14 HCS DISTRIBUTORS AND PARTNERS ... 74***

# 1 Abbreviations

| Abbreviation | Description |
|---|---|
| DMA | **D**igital **M**odule **A**mplifier |
| EMC | **E**lectro **M**agnetic **C**ompatibility; properties of the unit in order to comply with the EMC directive |
| USB | **U**niversal **S**erial **B**us; interface for communication between a PC and the DMA-22(A) |
| PC | **P**ersonal **C**omputer |
| µC | **M**icro **C**ontroller; CPU used inside the unit to control all functions |
| mamsl | **m**eters **a**bove **m**edian **s**ea **l**evel |
| IP65 | Protection Class; solid particle protection and protection level against water ingress |
| LED | **L**ight **E**mitting **D**iode |
| PWM | **P**ulse **W**idth **M**odulation; method to control the output current at low power losses |
| Ub | Supply voltage |
| FS | **F**ull **S**cale |

Table 1: Abbreviations

# 2 Terms and Definitions

| Term | Description |
|---|---|
| HCSTool | Software tool provided by HCS in order to adjust and parametrize DMA-22(A) units |
| CANopen | Bus interface for digital communication, defined by CIA |
| PROFIBUS | Process Field Bus interface for digital communication, defined by BMBF (German department of education and research) and used by Siemens |
| PROFINET | Process Field Net interface for digital communication, the standard is maintained and supported by PROFIBUS & PROFINET International |
| ETHERNET/IP | (IP = Industrial Protocol) is an industrial network protocol that adapts the Common Industrial Protocol to standard Ethernet |
| USB-C connector | Connector type used on the USB interface |

Table 2: Terms and definitions

# 3 General information

## 3.1 About this manual

This document describes the Profinet interface of the HCS digital amplifier.
It describes and explains the general structure of the Profinet interface and in a second part the device specific

This manual was prepared with great care and the contents reflect the author's best knowledge. However, the possibility of error remains and improvements are possible. Please feel free to submit any comments regarding errors or possibly incomplete information to HCS GmbH.

**DANGER**

**Read this manual carefully before working with the digital amplifier cards!**

**The general and especially the safety instructions of this manual have to be observed in any case!**

**Electronic equipment of all kind can have component failures or software may lead to unpredictable reactions. Secondary safety measures must be taken in order to ensure safety under all circumstances. This is especially the case for safety critical applications.**

**Furthermore, it is the responsibility of the user to always comply with applicable safety standards (e.g. EN 13849) and to implement a system architecture capable to cover all safety requirements. The unit itself does not comply with any performance level given in EN 13849 or any SIL level as per EN 61508.**

**Applicable laws and safety standards have to be observed at any time. Before commissioning or using the equipment, the necessary risk analysis must be conducted and suitable protection measures must be taken.**

**HCS GmbH refuses any liability in case of omission to comply with these requirements.**

## 3.2 Scope

With the help of this document a professional, competent and trained user should be able to install, wire, connect, commission, set parameters, perform fault analysis and fix issues. Also the document will provide the user any technical and operational details necessary to safely operate the product. It remains the users responsibility to set-up a safe working environment while operating the product.

The information contained in this manual is valid at the time of this version's release. See footer for version number and release date of this manual.

### 3.2.1 Copyright

© All rights reserved. Except for the usual review purposes, no part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information retrieval system, without the written permission of **HCS HYDRAULIC CONTROL SYSTEMS GmbH** (following referred to as HCS).

### 3.2.2 Documents place of storage

This manual and all other associated documentation for hardware and software must always be kept in a location where they will be readily accessible and close to the digital amplifier and controller card DMA-22(A) or the equipment in which it is installed.

### 3.2.3 Warning signs, symbols and notes

**⚠ DANGER**

**Identifies safety instructions that are intended to warn of an immediate and impending danger to life and limb. Failure to observe these safety instructions will – with a very high probability - lead to death, serious personal injury (disablement)!**

**⚠ WARNING**

**Identifies safety instructions that are intended to warn of potential danger to life and limb. Failure to observe these safety instructions might lead to death, serious personal injury (disablement)!**

**⚠ CAUTION**

**Identifies safety instructions that are intended to warn of slight personal injury. Failure to observe these safety instructions might lead to slight personal injury.**

**NOTICE**

**Failure to observe this safety notice can result in property damage!**

Identifiers important information

Other identifiers (when applicable):

| Identifier | Description |
|---|---|
| ●/ - | Listings |
| □ | References to another chapter, page, table or figure in this manual |
| blue text | Hyperlink within the document |
| 1., 2., … | Steps in a procedure that should be performed in consecutive order |
| 'STATE' | States of a state machine |
| «ES» | LEDs of the amplifier (for example, «ES») |
| < > | parameter name |
| "…" | Used for references |

Table 3: Identifiers

### 3.2.4 Safety and liability

**NOTICE**

**The information in this document is subject to change without further notice.**
**HCS assumes no responsibility for any faults that may appear in this manual.**

**This operating manual represents the knowledge of HCS and during the drafting of this operating manual the greatest possible care was taken. Nevertheless, HCS disclaims any responsibility and liability claims for individual applications of the user. This is particularly true in cases of non-compliance, omissions, faults, misinterpretations and misunderstandings.**

**Applicable laws and safety standards have to be observed at any time. Before commissioning or using the equipment, the necessary risk analysis must be conducted and suitable protection measures must be taken.**

**This manual only describes the functionality and influence of the parameters. The described software functionality can be used in various amplifier models which can be implemented in a vast range of applications. Hence it is not possible to assume liability for the influence of the parameters.**

**HCS GmbH refuses any liability in case of omission to comply with these requirements.**

**⚠ WARNING**

**In the case of applications with critical safety requirements or where accident prevention regulations must be observed, it may be necessary to isolate the components from the solenoids with relays in e-stop circuits.**

**To switch off the enable signal (0 V at terminal 8d) is insufficient. In these cases, hydraulic and/or mechanical safety measures to stop the drive must be provided (e.g. through switching valves with position monitoring).**

**⚠ WARNING**

**All types of proportional directional valves, which may be used in all kinds of different environments and applications, can eventually fail and hence are able to cause damage.**

**It is the customers obligation to analyze all safety related aspects of the application. It is within the full responsibility of the machine builder and/or system integrator to make the final selection of products and reassure and confirm that all requirements regarding safety and performance are met.**

**The procedure of selecting the proper control system and related safety levels is covered by the machine directive EN 13849.**

**NOTICE**

**Liability for any content of this document is rejected!**

**Any claims against HCS GmbH – based on whatever legal reason – resulting from the use of the information, programs, engineering and performance data etc., described in this document shall be excluded. Exclusion in such cases shall not apply for mandatory liability, e.g. under the "Product Liability Act", in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition. Damages for breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.**

**We reserve the right to make changes to this document at any time without prior notice.**

### 3.2.5 General information

**⚠ WARNING**

**ESD (electrostatic discharge) may damage components on the card. In order to prevent any damage please always follow the recommendations below!**

**Discharge static voltage from your body by using best practice.**
**Work in safe environment and do not use any devices in the working surrounding which could generate or hold static charge. Avoid using the product in areas where floors or surfaces are composed of materials that could generate static charges.**

**Handle all equipment very carefully and do not touch exposed pins or components.**

**Store and transport the units only in its original packaging.**

**⚠ WARNING**

**During commissioning, particular attention must be paid to the correct design and realization of the wiring. This must be checked before applying the supply voltage.**

**To avoid collisions, the safety devices and limit switches must be activated. All safety regulations are to be observed.**

**Monitoring the fault signal (terminal X3/10) is recommended.**

### Malfunctions can occur with the following:

- Changes to the settings made by the supplier.
- Neglecting of operating parameters (e.g. supply voltage, application of inadmissible signals on inputs or outputs, ambient conditions, wiring, unsafe loads such as motors, contactors, relays, ohm loads, etc.)
- Faults in series-connected controls components and set points or actual values
- Faults in the subsequent hydraulic components
- Removing of solenoids connections.

## 3.3 Terms and conditions

Please pay attention to our general terms and conditions (available on request).

## 3.4 Delivery state (default setting)

The product is shipped in a ready-to-use state (default settings). After correct installation and setting of all parameters relevant for the application, the amplifier card is ready for use.

## 3.5 Correct product use

The digital module amplifier “DMA-22(A)-x-PN” are used for the following applications:

- Controlling of proportional valves (namely directional control valves); with or without feedback (depending of operation mode).
- Controls the delivery volume of a pump with a proportional or servo valve in terms of pressure and performance, without the need for hydraulic-mechanical control parts.
- For industrial applications only.

The operation of other inductive or resistive loads is not recommended (motors, heaters, etc.). If you are considering any applications like this please contact HCS GmbH.

If used correctly, the safety of the user as well as the safe and proper function of the amplifier DMA-22(A) is given.

## 3.6 Non-intended product use

The digital amplifier and controller cards “DMA-22(A)-x-PN” should not be used in the following cases:

- if noticeably damaged
- when the electrical connections are damaged
- if they do not function correctly
- after incorrect handling or storage
- in non-appropriate applications or environments.

In these cases, the digital amplifier and controller card must be disabled and secured against accidental restarting.

In the case of applications with critical safety requirements or where accident prevention regulations must be observed, it may be necessary to isolate the components from the solenoids with relays in e-stop circuits. In these cases, hydraulic and/or mechanical safety measures to stop the drive must be provided (e.g. through switching valves with position monitoring).

During commissioning, particular attention must be paid to the correct design and realization of the wiring. This must be checked before applying the supply voltage.

To avoid collisions, the safety devices and limit switches must be activated. All safety regulations are to be observed.

Malfunctions can occur when:

- Changes to the settings made by the supplier.
- Neglecting of operating parameters (e.g. supply voltage, application of inadmissible signals on inputs or outputs, ambient conditions, wiring, unsafe loads such as motors, contactors, relays, ohm loads, etc.)
- Faults in series-connected controls components and set points or actual values
- Faults in the subsequent hydraulic components
- Removing of valve connection.

## 3.7 Selection and qualification of personnel

Only qualified users may work with the amplifier. Qualified users are properly trained experts with the required knowledge and experience. In particular, these experts must have the authorization to bring into operation systems and power circuits in accordance with safety engineering standards. They must be familiar with safety concepts common in automation.

### 3.7.1 Inquiries and ordering

To order the product, the complete order code is requested.
see □ Chapter „4.2 Model code", page 13
Please pay attention to our general terms and conditions (available on request) and forward orders and enquiries to the companies as listed on our web page:

# Sales Partners (h-c-s-gmbh.de)

https://www.h-c-s-gmbh.de/en/sales-partners

They are our official distributors/partners. We reserve the right to forward any direct inquiries or orders to our distributors/partners. Nevertheless, we will be available for any necessary technical support.

Germany and other countries without distributor/partner:

Neuffener Strasse 29
D-72636 Frickenhausen
Tel: (+49) 7025 - 911 007
Fax: (+49) 7025 - 911 008
Email: info@h-c-s-gmbh.de
www.h-c-s-gmbh.de

### 3.7.2 Service and repair

**⚠ WARNING**

**Do not attempt - under any circumstances - to repair the product yourself**

After repair, certain adjustments and test procedures must be performed; this can only be made by qualified and authorized personnel. Products that need repair can be sent to the addresses:
See □ Chapter "3.7.1 Inquiries and ordering", page 10
Please enclose a detailed description of the error, malfunction or failure with the sent item and state the serial number and the purchase date. This will speed up the process and guarantees a fast and reliable repair.
See □ Chapter "3.7.1 Inquiries and ordering", page 10
In the case of a fault or a malfunction, your distributor can give you instructions on the phone or in writing before accepting a repair order. For service and repair, we offer experienced and qualified personnel. In case you need our assistance, please contact the address:
See □ Chapter "3.7.1 Inquiries and ordering", page 10

### 3.7.3 Cleaning, storage, transport

The product should only be transported and stored in the original packaging to ensure suitable protection against mechanical damage as well as electrostatic discharge.

- Keep the amplifier card away from moisture and dust
- Observe the allowed storage and transportation temperature range
- If it should be necessary to clean the amplifier, we recommend sending it back to the manufacturer or any distributor and partner:
  See □ Chapter "3.7.1 Inquiries and ordering", page 10

| ⚠ CAUTION |
| --- |
| **Unpacking and handling should be left to suitably trained personnel.<br>Beware of damaging the amplifier by electrostatic discharge** |

### 3.7.4 Delivery state (default setting)

The product is shipped in a ready-to-use state (default settings). After correct installation and setting of all parameters relevant for the application, the digital amplifier and controller card is ready for use.

## 3.8 Introduction

The digital module amplifier and controller “DMA-22(A)-x” features leading edge technology. This electronic device meets the industrial standards for EMC. This ensures a high interference security and low interference emission.

The performance characteristics are possible through the use of the most current microprocessor technology (32-bit floating-point CPU) combined with other state of the art components. In addition to all control functions, the microprocessor also handles closed-loop control. The system features are essentially determined by the software and provide reserve capacity for further developments and adaptations.

The following features distinguish the “DMA-22(A)-x” series:

- Fully digitized amplifier card with the advantage of
  - no on-board potentiometer
  - no jumper settings required
  - digital setting and display of all parameters by means of PC with HCSTool
  - user safety during parameterization
- flexible and reliable system:
  - use of a modern 32-Bit floating point µC
  - high performance and yet power reserve
  - high reliability and safety through the use of a integrated watch-dog and reset function
  - variable settings for magnetic systems making high flexibility possible
- functional use of the interface:
  - change of selected parameters “on-the-fly” without interference or interruptions of the working cycle
  - analyzing of system performance through selection of display parameters with the PC and by using the 4-channel oscilloscope function included in HCSTool
  - bus interfaces available (PROFIBUS/PROFIDRIVE, PROFINET, ETHERNET/IP, CAN-OPEN)

## 3.9 General Applications

The amplifier card “DMA-22(A)-x” is generally used for

- Controlling of hydraulic motors, - installations and – systems, e.g.:
  - position
  - speed
  - pressure
  - revolutions per minute
  - torque
  - power etc.

- Controlling of different process values:
  - pump controlling
  - controlling of pressures
  - controlling of pilot- and main stage
  - cascade controlling of components etc.

# 4 Technical data

## 4.1 Technical data overview

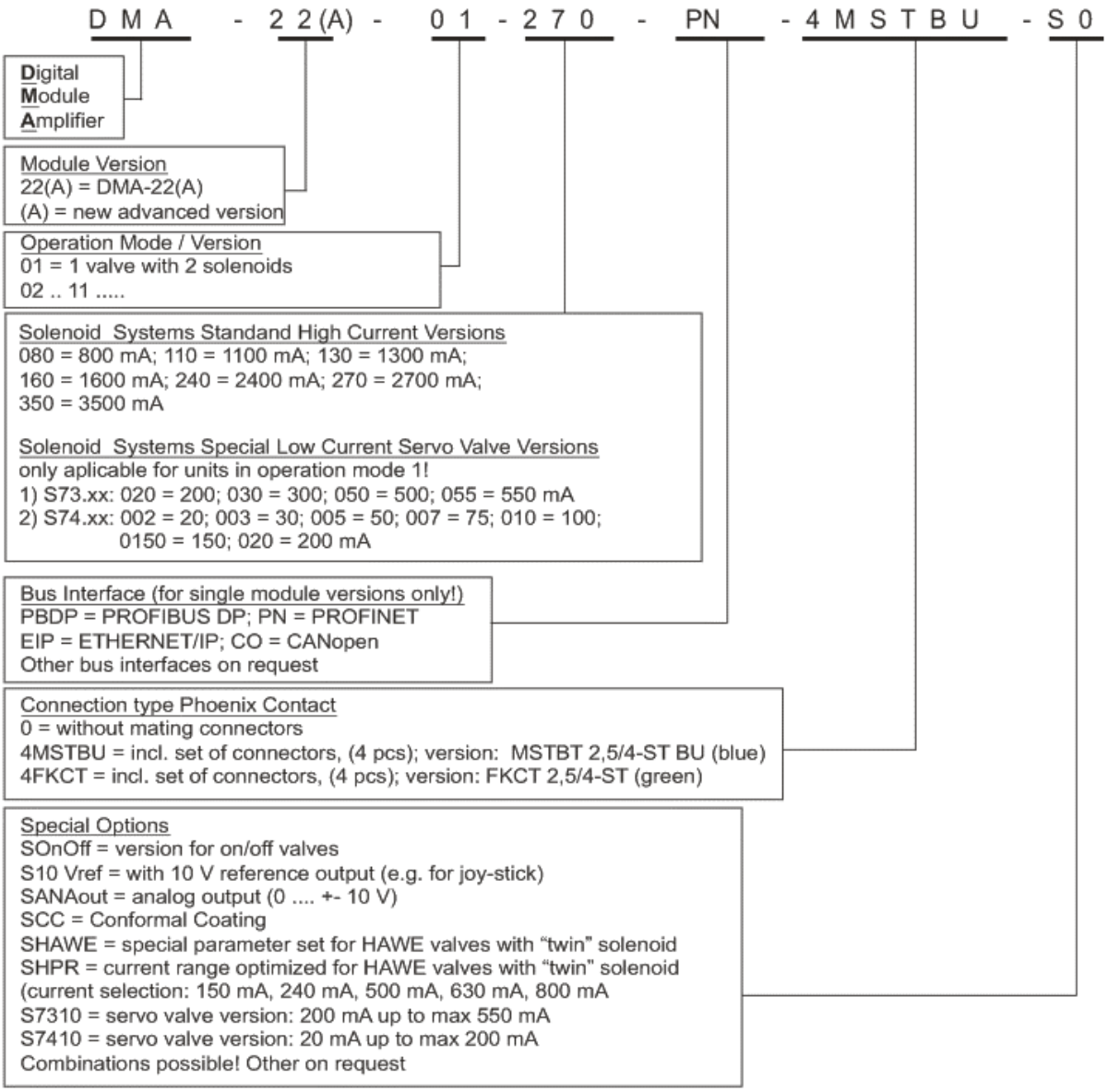
| Feature | Range, characteristics |
|---|---|
| Model | Modular digital amplifier in snap-on mounting technology |
| Supply voltage | 10 V DC ... 32 V DC; residual ripple < 10 % (max. 50 VA power draw) |
| Duty cycle | 100 % |
| Preliminary fuse | 3.15 A; quick blowing |
| Ambiant operating temperature | - 40 °C ... + 70 °C |
| Storage temperature | - 45 °C … 85 ° C |
| Humidity (relative air humidity) | max. 95 % non condensing |
| Max. elevation | 2,000 m (mamsl) |
| Class of protection | IP20 (EN60529) |
| EMC | In accordance with the applicable industrial standards (CE - conformity) *1 |
| Connection, type of connector | 16 pole (4 x 4); screw terminals for 0.2 - 2.5 mm² (AWG 24 -12)<br>Phoenix Combicon connector with screw terminals, type: MSTBT 2,5/ 4-ST<br>for detailed technical data refer to Phoenix Contact Combicon Product Catalog |
| Cable specification | 1.5 mm² (AWG16) for supply and solenoids, shielding recommended, max length: 50 m<br>0.5 mm² (AWG20) for analog and digital signals, shielding mandatory, max length: 50 m |
| Mounting/housing | Mounting: top-hat rail (mounting rail) in accord. with EN50022 with inetgrated PE contact<br>Housing configuration: ventilated (IP20)<br>Material: PA 66 - FR (blue); flammability in accordance with UL94V0<br>Dimensions approx.: (w x h x d) 22,5 x 100 x 114 mm<br>Weight: approx. 0.13 kg (inlcuding mating connectors) |
| Mounting postion | Any; preferably vertical for better heat dissipation through convection |
| Analog set value input | 0 … +-10 V or 4 ... 20 mA (parameter selectable) with 16 Bit resolution (volt. = differential)<br>Input resistance: voltage input = 200 kOhm; current input = 255 Ohm |
| Analog feedback input | 12 V ... +- 2.5 V<br>Special input for PRL2 LVDT spool position transducer with filter function |
| Digital inputs | 1 input (ENABLE); |
| Signal range for digital inputs | 0 ... 2.6 V = low; > 9 V = high; current draw input < 20 mA; input resistance: 20 kOhm |
| Solenoid current (output) | 2 PWM output stages, each for up to 3,5 A (with over-energ. and quick de-energization) |
| Solenoid system selection | 0.8 A; 1.1 A; 1.3 A; 1.6 A; 2.4 A; 2.7 A and 3.5 A (intermediate values adjustable)<br>Servo valve current ranges available! |
| Digital output | 1 output, Uout @ 10 mA load = USupply – 6 V; output = 0 V in case of error!<br>(ERROR / COMPARATOR funcion combined) |
| Status signals | 1 multi color „STATUS“ LED at front. Run/OK = green; Enable = yellow; Error = red |
| Interface 1 | USB-C at front |
| Interface 2 (for versions with option CANopen only) | CANopen (CiA 408 or Device Profile Fluid Power); connection to DMA using standard CAN frames according to ISO 11898-1…3 and ISO 11898-5 |
| PWM frequency | 22.2 kHz - independant from dither frequency (customer specific frequencies available) |
| Dither frequency range | 1 ... 300 Hz - independant from PWM frequency (other frequency range on request) |
| Cycle times | General and closed loop 1: 0.1 msec |

*1: details on request

Remark:
all values nominal if not stated otherwise!

Table 4: Properties

## 4.2 Model code




**Ordering code examples:**

Version for one valve with 2.7 A solenoid; operation in mode 1, including connectors

***DMA-22(A)-01-270-4MSTBU-S0***

Version with PROFINET for two valves and 0.8 A solenoids; operation in mode 2, including connectors

***DMA-22(A)-02-080-PN-4MSTBU-S0***

## 4.3 Features of the device

- Meets the standard IEC 61158 and IEC 61784.
- Meets PNO standard Version V2.2
- LAN 10/100Base-T(X)
- 2 x RJ-45 LAN (Daisy Chain)
- Cycling data exchange RT and IRT with an PROFINET IO-Controller
- Sending and receiving of diagnostic- and process alarms.
- I&M0...4-data available
- Supporting of PROFINET Naming (device name) and TCP/IP addressing
- Shared Device supported
- Media Redundancy Protocol (MRP) supported

## 4.4 Used GSDML File

„GSDML-V2.3-HCS-DMA-yyyymmdd.xml “
“GSDML-01F0-0002-DMA1.bmp”

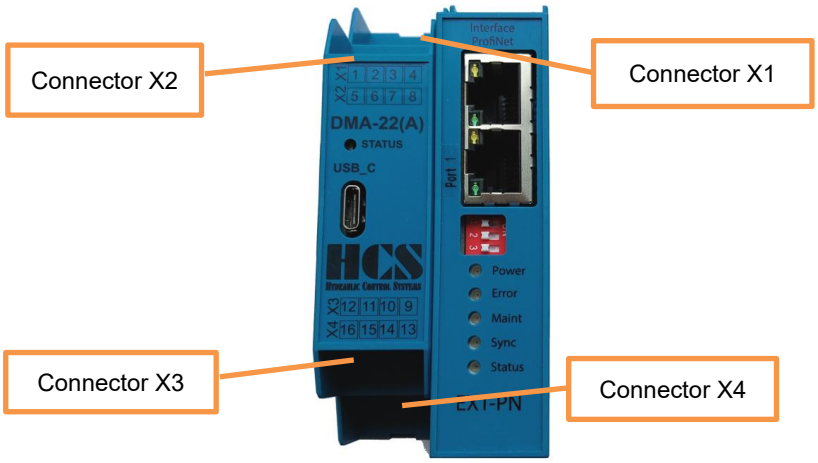
## 4.5 Programming PC Interfaces

Onboard USB interface in cooperation with HCSTool or HyperTerminal.

## 4.6 Connector definition

Options are only possible by hardware changes!
The possibility of some options is depending on the combination of all needed options.

### 4.6.1 Connector position layout




## 4.7 Mode 1,2 possible connector options (other on customer request)

| Usage | Mode 1 | Mode 2 | Option | Connector | PIN |
|---|---|---|---|---|---|
| Solenoid | Output A- | Output A- | - | X1 | 1/1 |
| | Output A+ | Output A+ | - | | 2/2 |
| | Output B- | Output B- | - | | 3/3 |
| | Output B+ | Output B+ | - | | 4/4 |
| set value | Set value S1.01 | Set value S1.01 | - | X2 | 1/5 |
| | Set value S1.02 | Set value S2.01 | - | | 2/6 |
| | Set value S1.03 | Analog set value S1.05- | - | | 3/7 |
| | Set value S1.04 | Analog set value S1.05+ | - | | 4/8 |
| Analog set value, Error, Enable | Enable Input | Enable Input | - | X3 | 1/9 |
| | \Error/Comp. Output | \Error/Comp. Output | - | | 2/10 |
| | Analog set value S1.06+ | Analog set value S1.06+ | - | | 3/11 |
| | Analog set value S1.06- | Analog set value S1.06- | - | | 4/12 |
| Power | - | - | Analog Output- | X4 | 1/13 |
| | - | - | Analog GND | | 2/14 |
| | Power 0V | Power 0V | - | | 3/15 |
| | Power 24V | Power 24V | - | | 4/16 |

Table 5: Connection definition, mode 1,2

### 4.7.1 Mode 3,4,5,6,8,10,11 possible connector options (other on customer request)

| Usage | Standard | Option mode 10,11 | Option mode 5 +10V Ref | Connector | PIN |
|---|---|---|---|---|---|
| Solenoid | Output A- | - | - | X1 | 1/1 |
| | Output A+ | - | - | | 2/2 |
| | Output B- | - | - | | 3/3 |
| | Output B+ | - | - | | 4/4 |
| Set value, Analog feedback 1 | S1.01 | - | Analog GND | X2 | 1/5 |
| | S1.02 / S2.01 | - | +10V ref | | 2/6 |
| | FB1 - | - | - | | 3/7 |
| | FB1 + | - | - | | 4/8 |
| Analog set value, Analog feedback 2, Error, Enable | Enable Input | - | - | X3 | 1/9 |
| | \Error/Comp. Output | - | - | | 2/10 |
| | set value S1.06+ External FB2+ | - | - | | 3/11 |
| | set value S1.06-External FB2 - | - | - | | 4/12 |
| Power, Digital-Input | - | Analogue GND | - | X4 | 1/13 |
| | - | Analog Output- | S1.02(Din_1) | | 2/14 |
| | Power 0V | - | - | | 3/15 |
| | Power 24V | - | - | | 4/16 |

Table 6: Connection definition, mode 3,4,5,6,8,10,11

## 4.8 Connection diagram

**CAUTION**

The connection diagram may vary due to specific hardware or software specifications. The corresponding connection diagram for the hardware you are using is always printed on the side of the DMA module.

Please always follow the connection plan printed on the module.

### 4.8.1 Mode 1

Example for the standard connection:

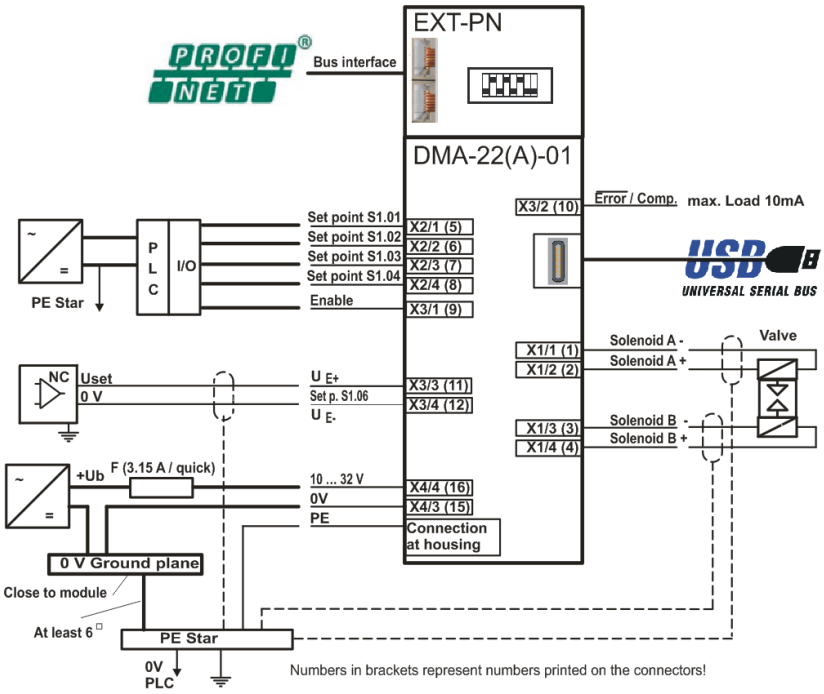



Figure 1 : Connection diagram mode 1

### 4.8.2 Mode 2

Example for the standard connection:

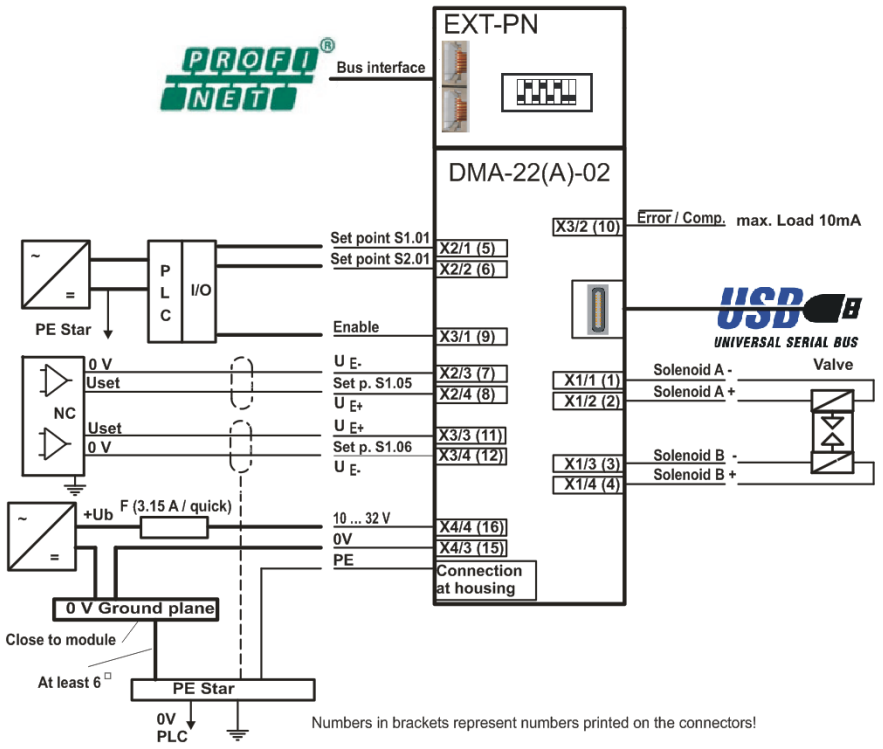



Figure 2 : Connection diagram mode 2

### 4.8.3 Mode 3, 4 – SW V 72.xx Servo valve

Example for the standard connection:

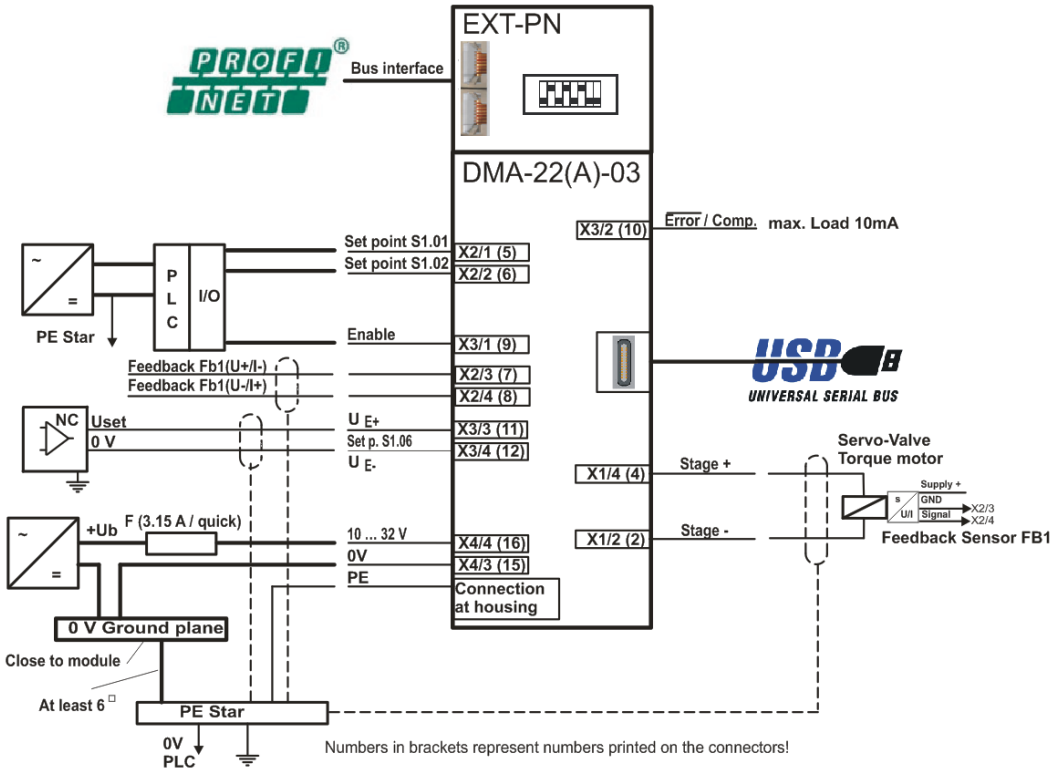



Figure 3 : Connection diagram mode 3,4 servo valve

### 4.8.4 Mode 3, 4

Example for the standard connection:

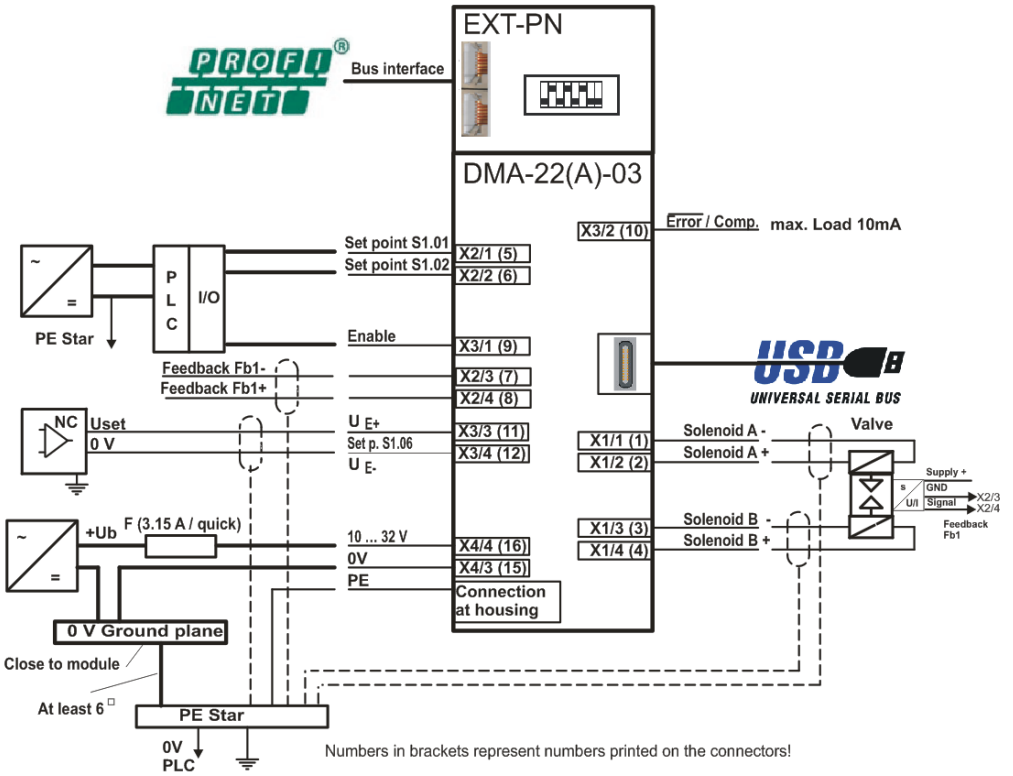



Figure 4 : Connection diagram mode 3,4 w.o. analog output

Example for the connection with analog output:

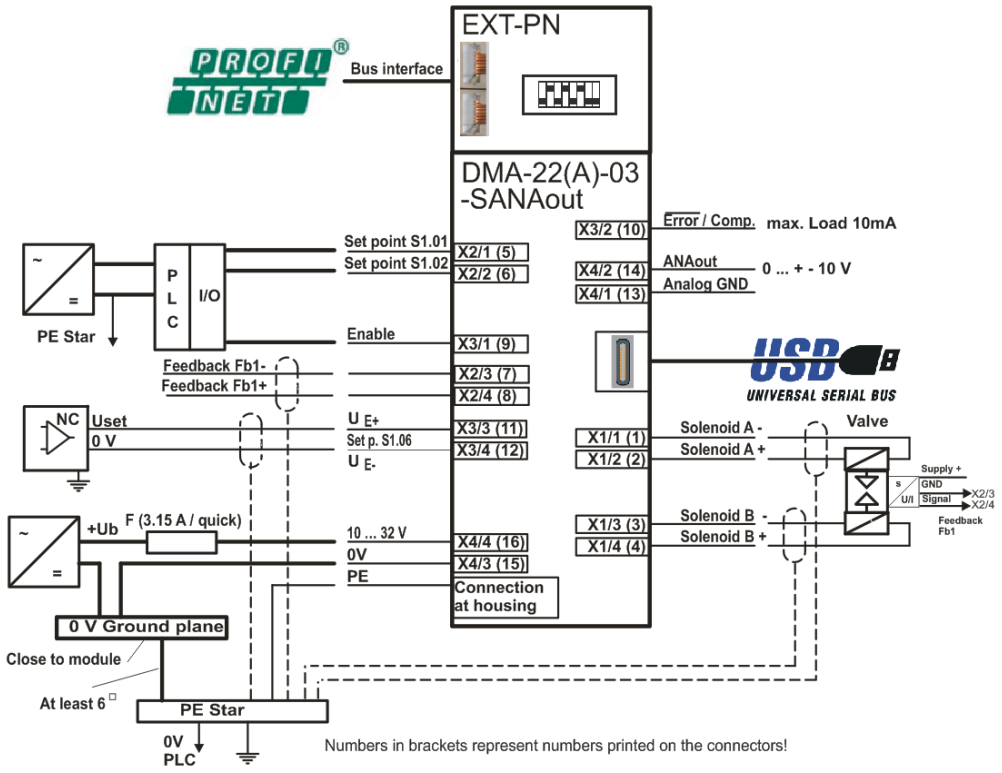



Figure 5 : Connection diagram mode 3,4 with analog output

### 4.8.5 Mode 5

Example for the standard connection with 10V reference output:

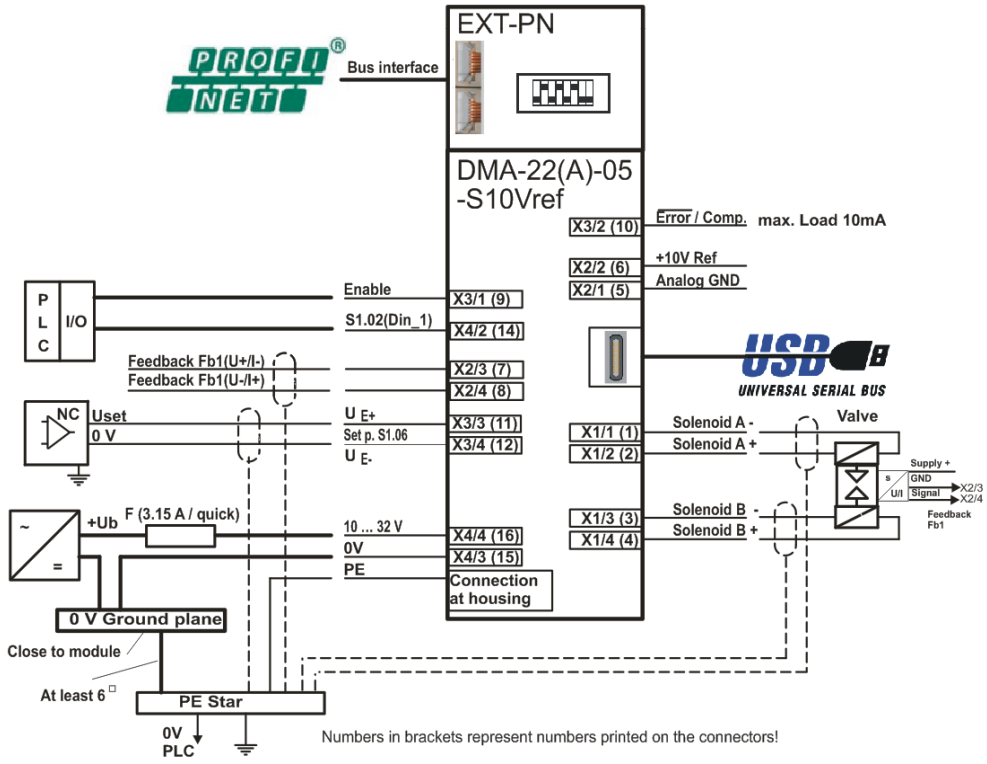



Figure 6 : Connection diagram mode 5 with 10V ref.

### 4.8.6 Mode 6

Example for the standard connection:

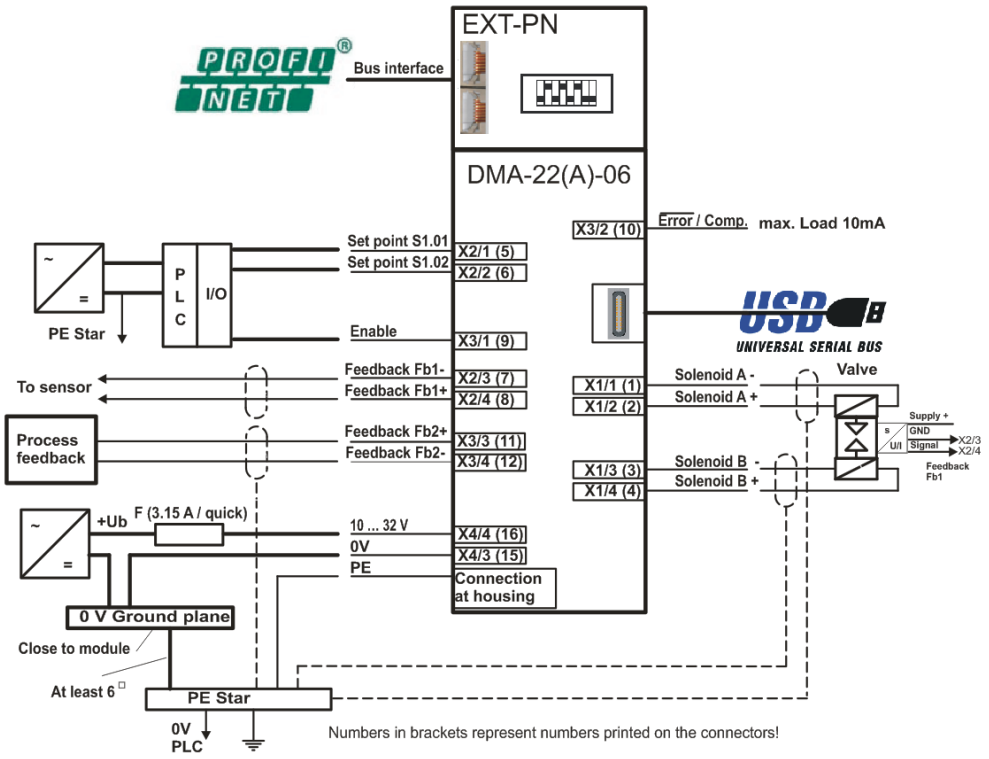



Figure 7 : Connection diagram mode 6

### 4.8.7 Mode 8

Example for the standard connection:

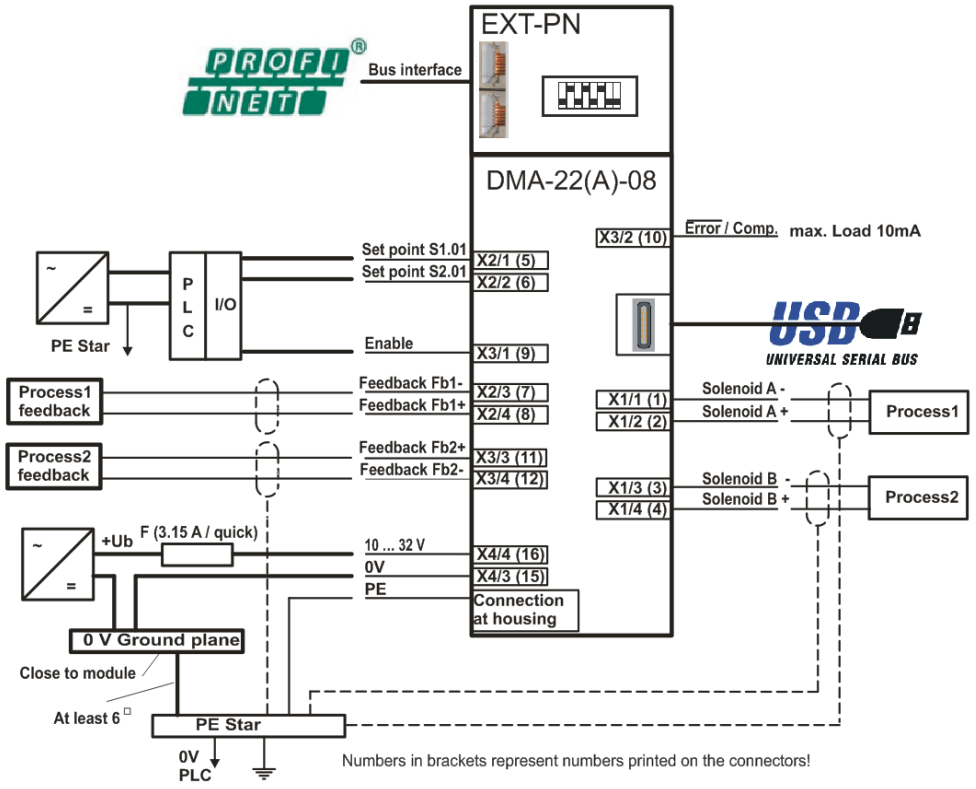



Figure 8 : Connection diagram mode 8

### 4.8.8 Mode 10

Example for the connection with Analog output:

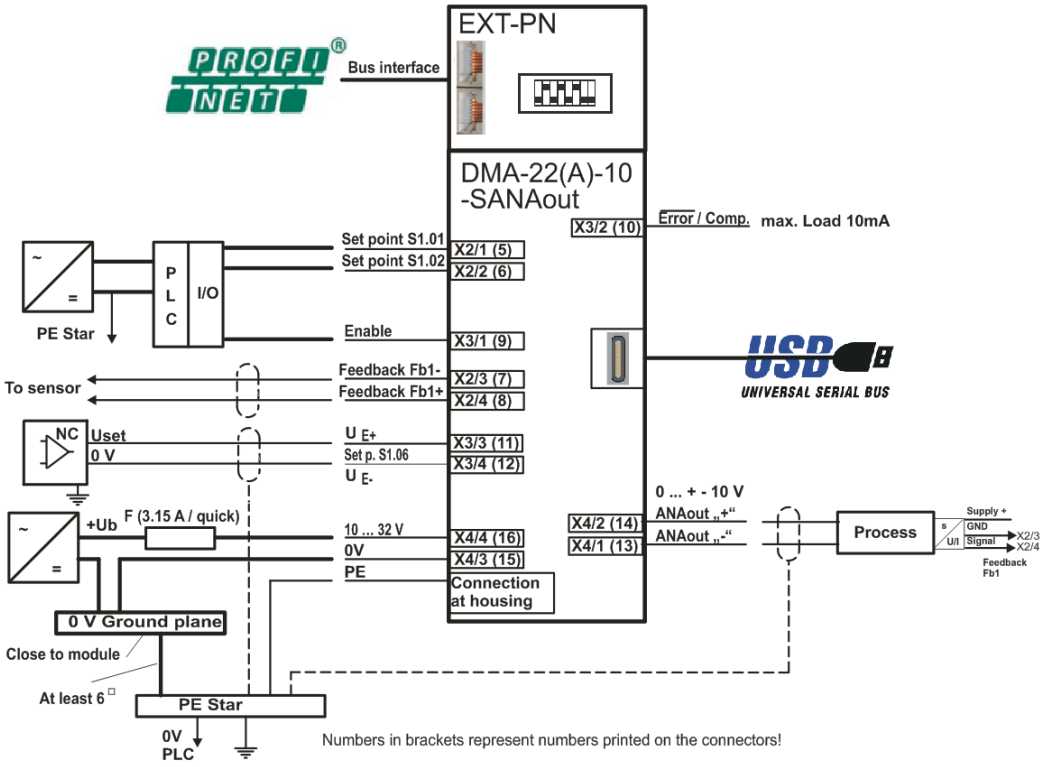



Figure 9 : Connection diagram mode 10

### 4.8.9 Mode 11

Example for the connection with Analog output:

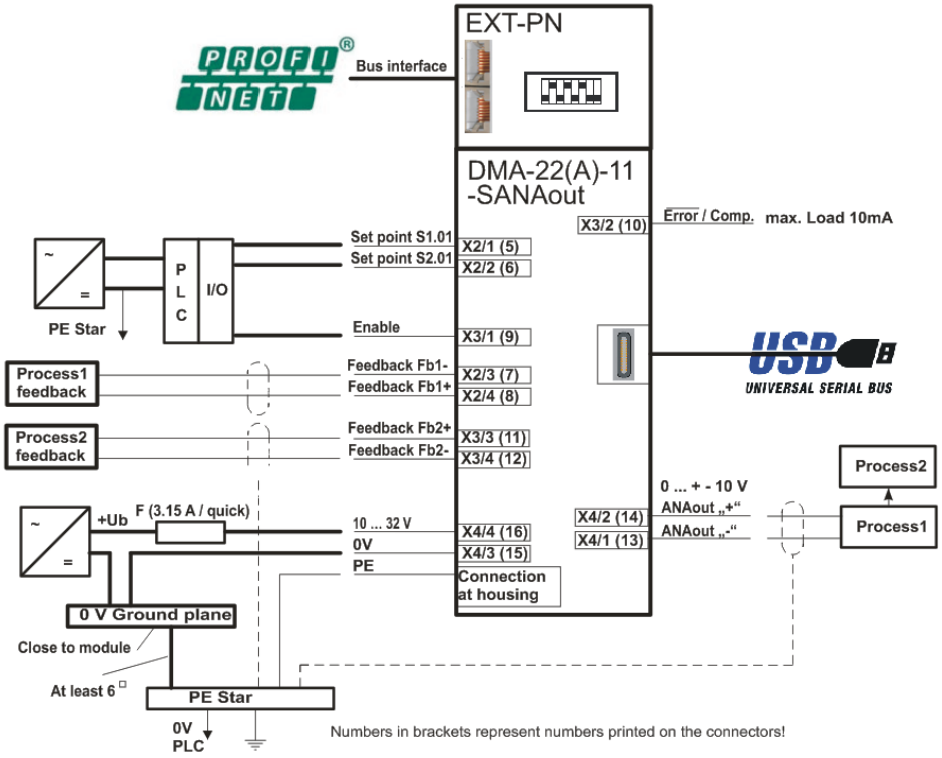



Figure 10 : Connection diagram mode 11

## 4.9 Device controller structure

### 4.9.1 Block diagram

Numbers in brackets do indicate the numbers printed on the connectors X1..X4

**Not all predefined parameters are available in all control modes. Please make sure in the parameter overview which parameters are valid for which mode.**

#### 4.9.1.1 Option “analog output” -SANAout

If the option -SANAout is defined, the analog Output is active

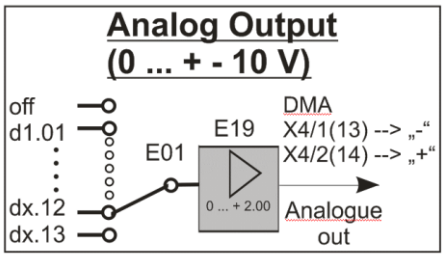



Figure 11: Option: analog output

### 4.9.1.2 Mode 1, Open loop, 1 valve

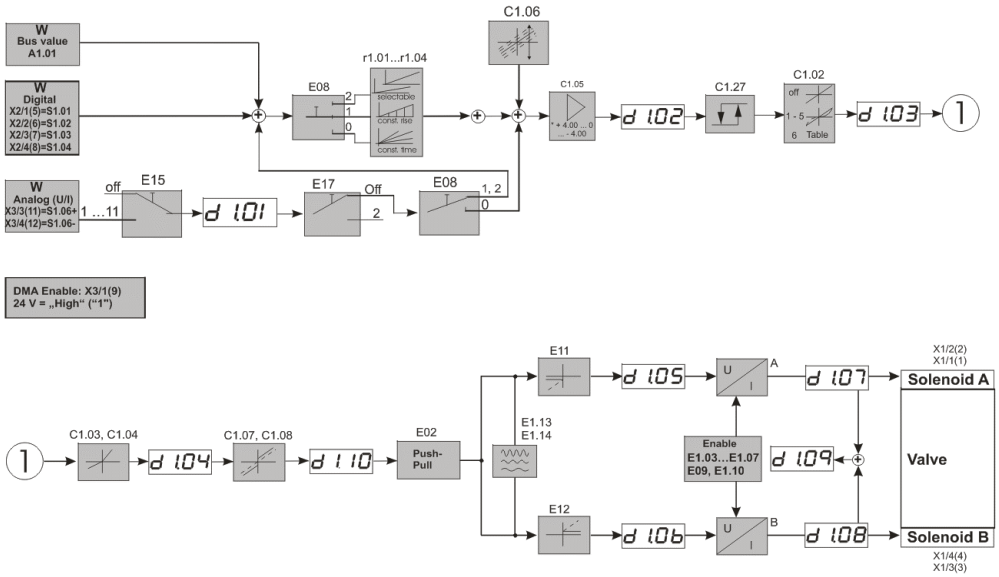



Figure 12: Block diagram mode 1

### 4.9.1.3 Mode 2, Open loop, 2 valves

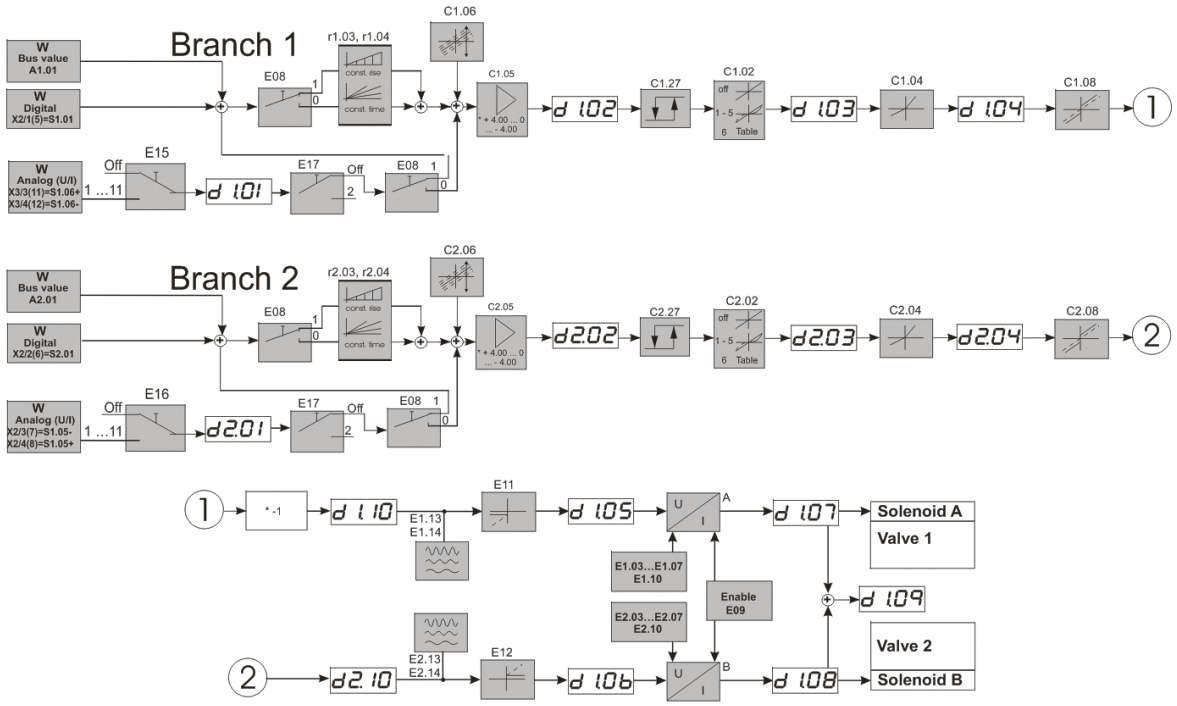



Figure 13: Block diagram mode 2

### 4.9.1.4 Mode 3 Closed loop, 1 valve

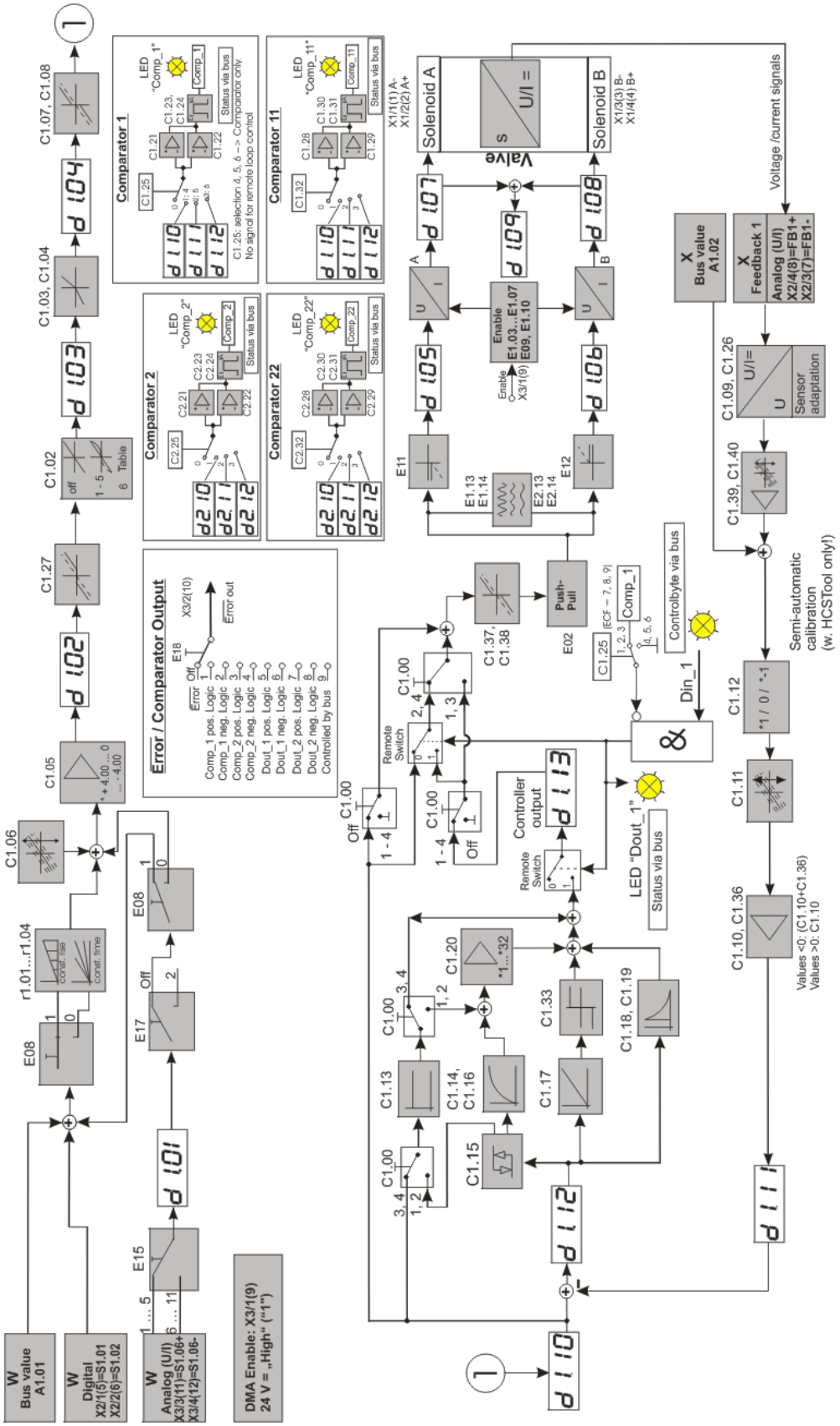



Figure 14: Block diagram mode 3

### 4.9.1.5 Mode 3 -SServo Closed loop, 1 valve

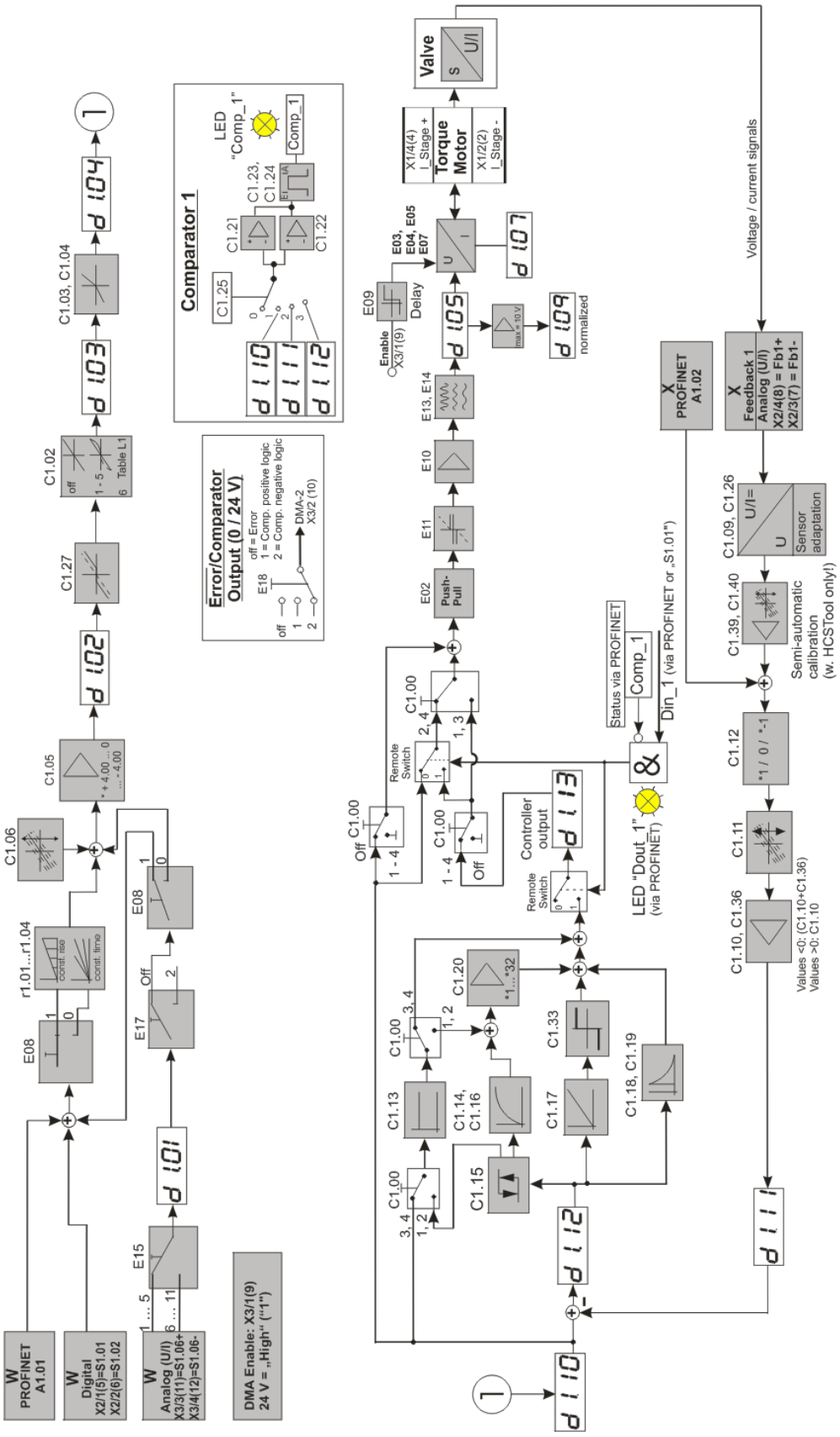



Figure 15: Block diagram mode 3 with servo valve

### 4.9.1.6 Mode 4, Closed loop, 1 process

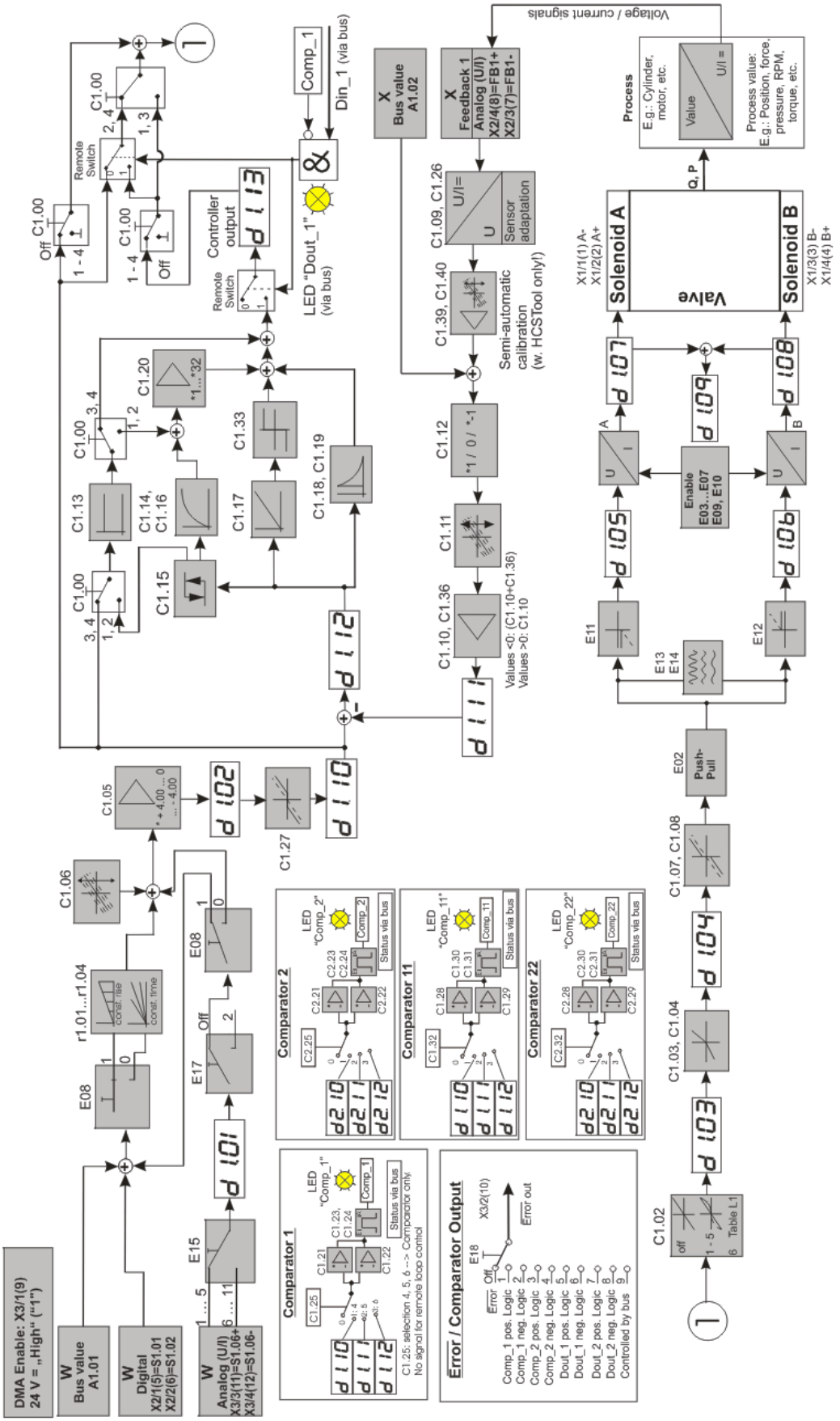



Figure 16: Block diagram mode 4

### 4.9.1.7 Mode 4 -SServo, Closed loop, 1 process

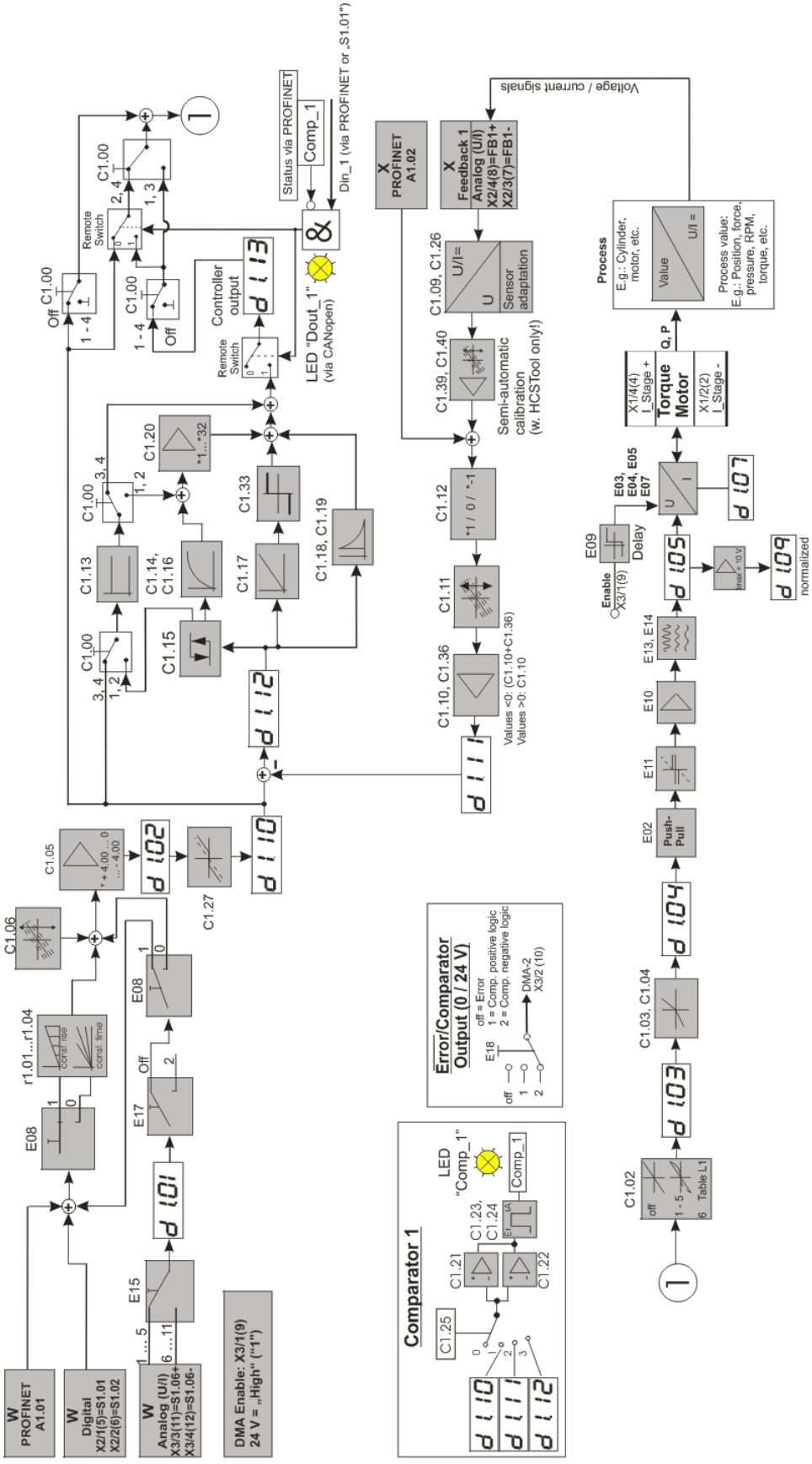



Figure 17: Block diagram mode 4 with servo valve

### 4.9.1.8 Mode 5, Closed loop, 1 process

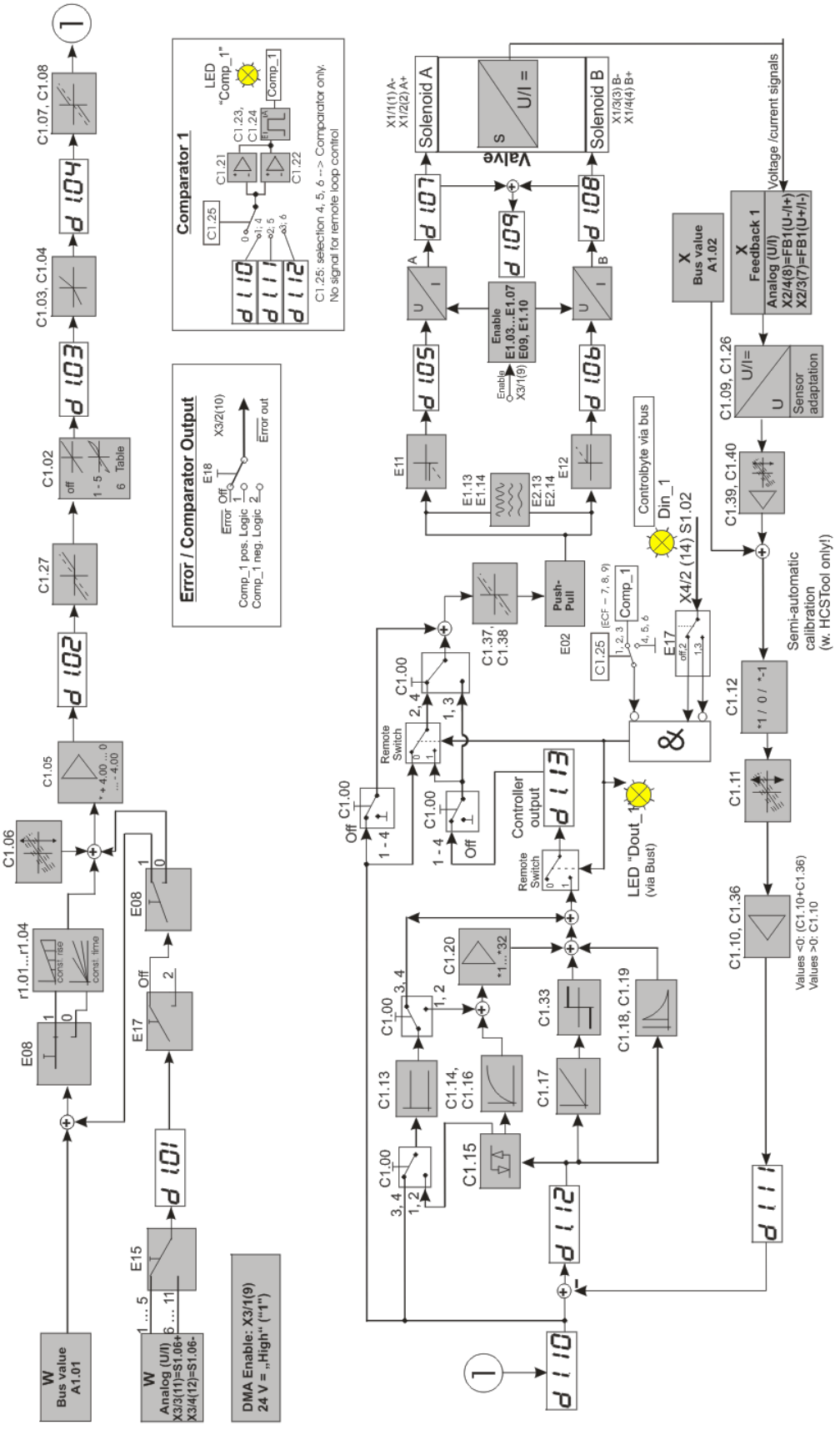



Figure 18: Block diagram mode 5

### 4.9.1.9 Mode 6, Closed loop, 1 valve, 1 process

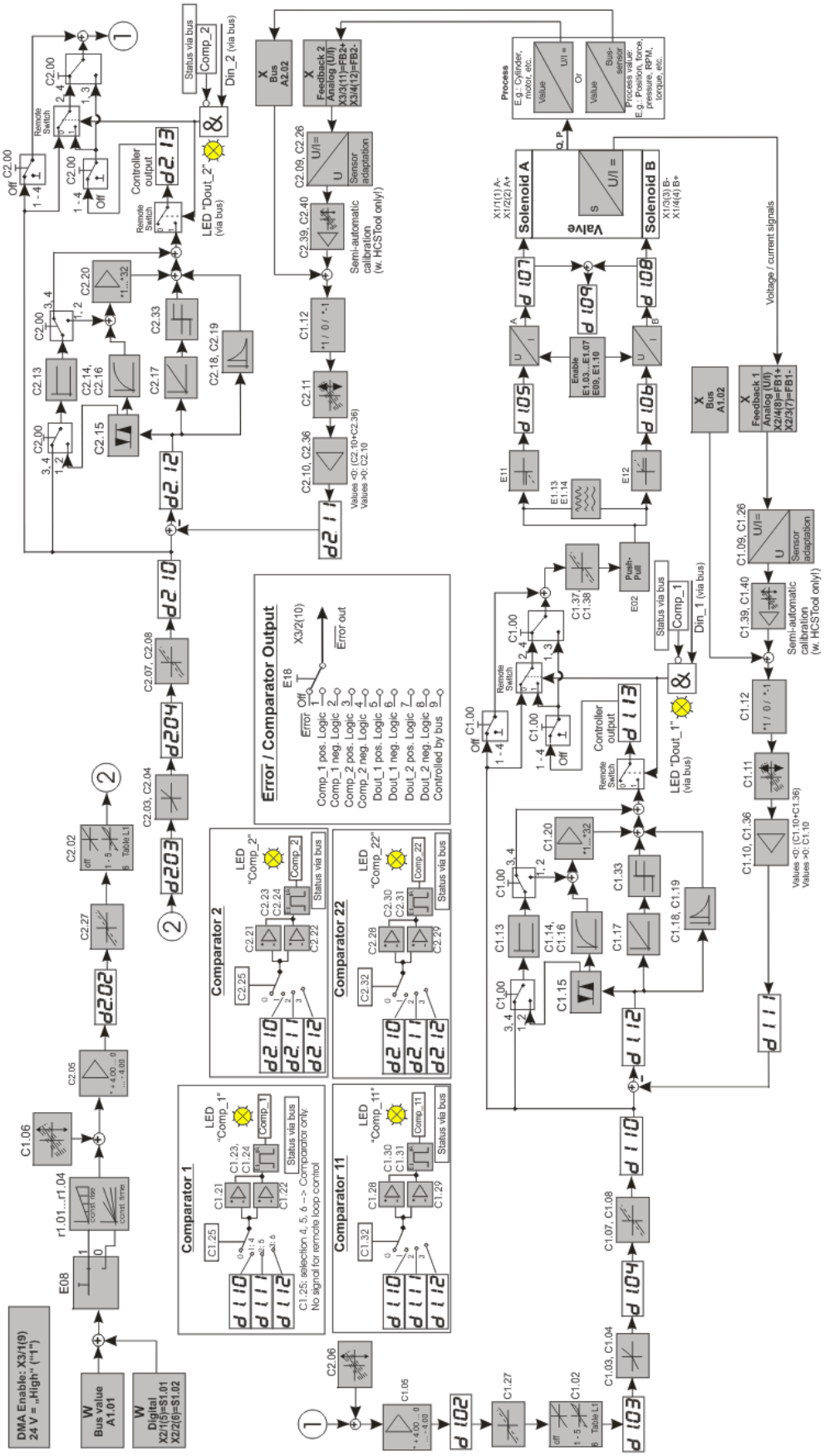



Figure 19: Block diagram mode 6

### 4.9.1.10 Mode 8, Closed loop, 2 processes

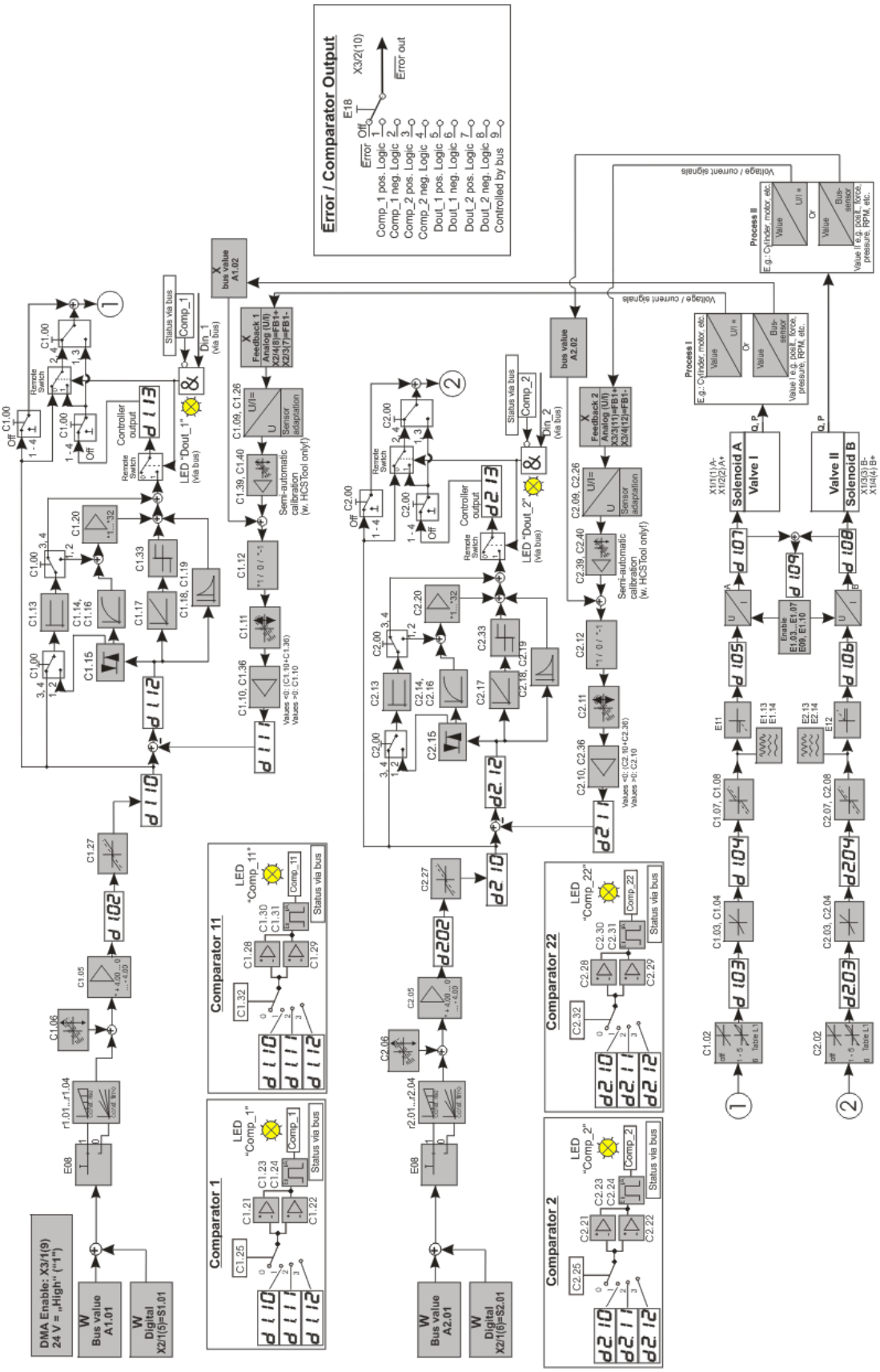



Figure 20: Block diagram mode 8

### 4.9.1.11 Mode 10 -SANAout, single closed loop – no valve

Hardware option “Analog out” is needed!

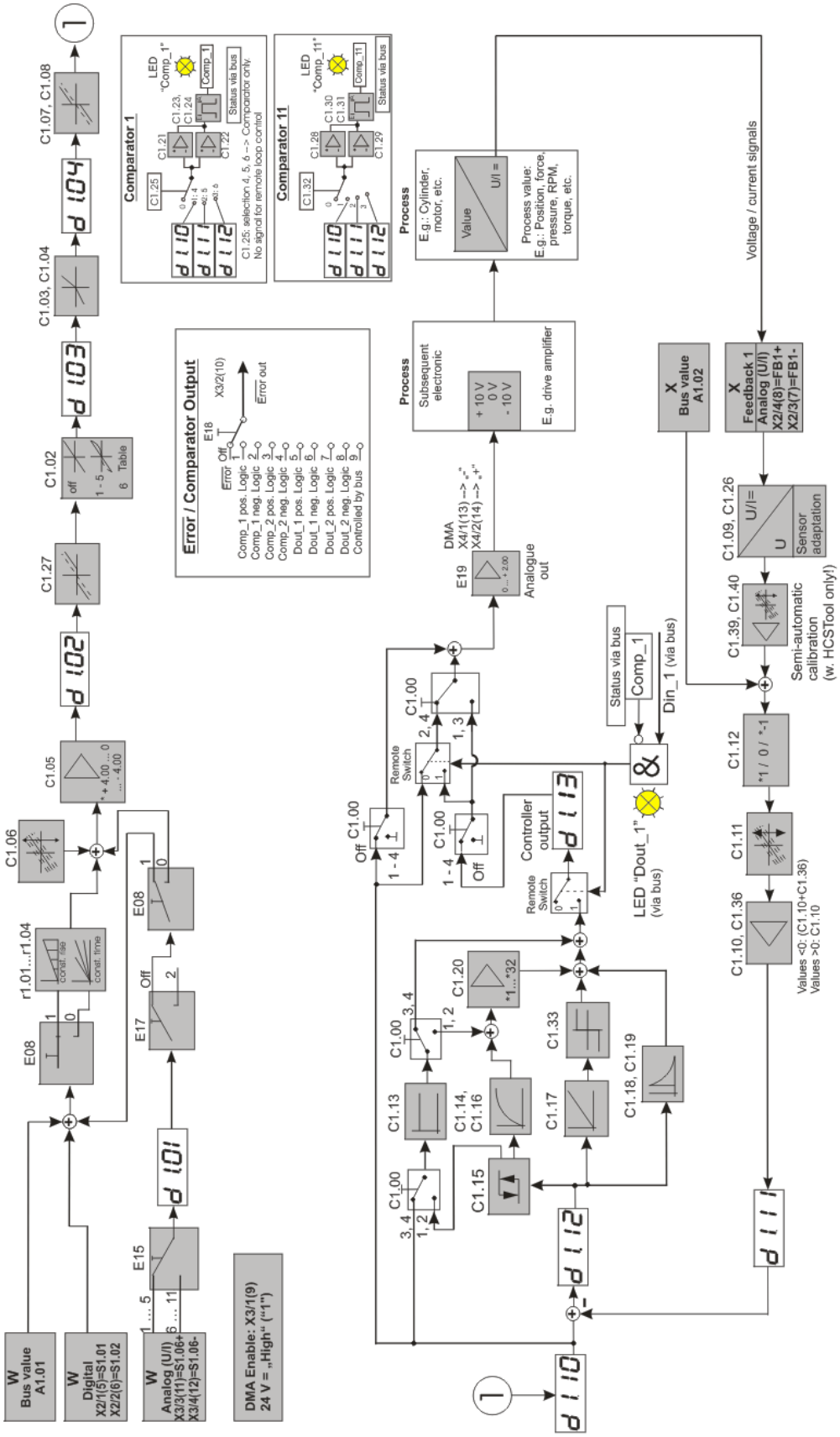



Figure 21: Block diagram mode 10

### 4.9.1.12 Mode 11 -SANAout, double closed loop – no valve

Hardware option “Analog out” is needed!

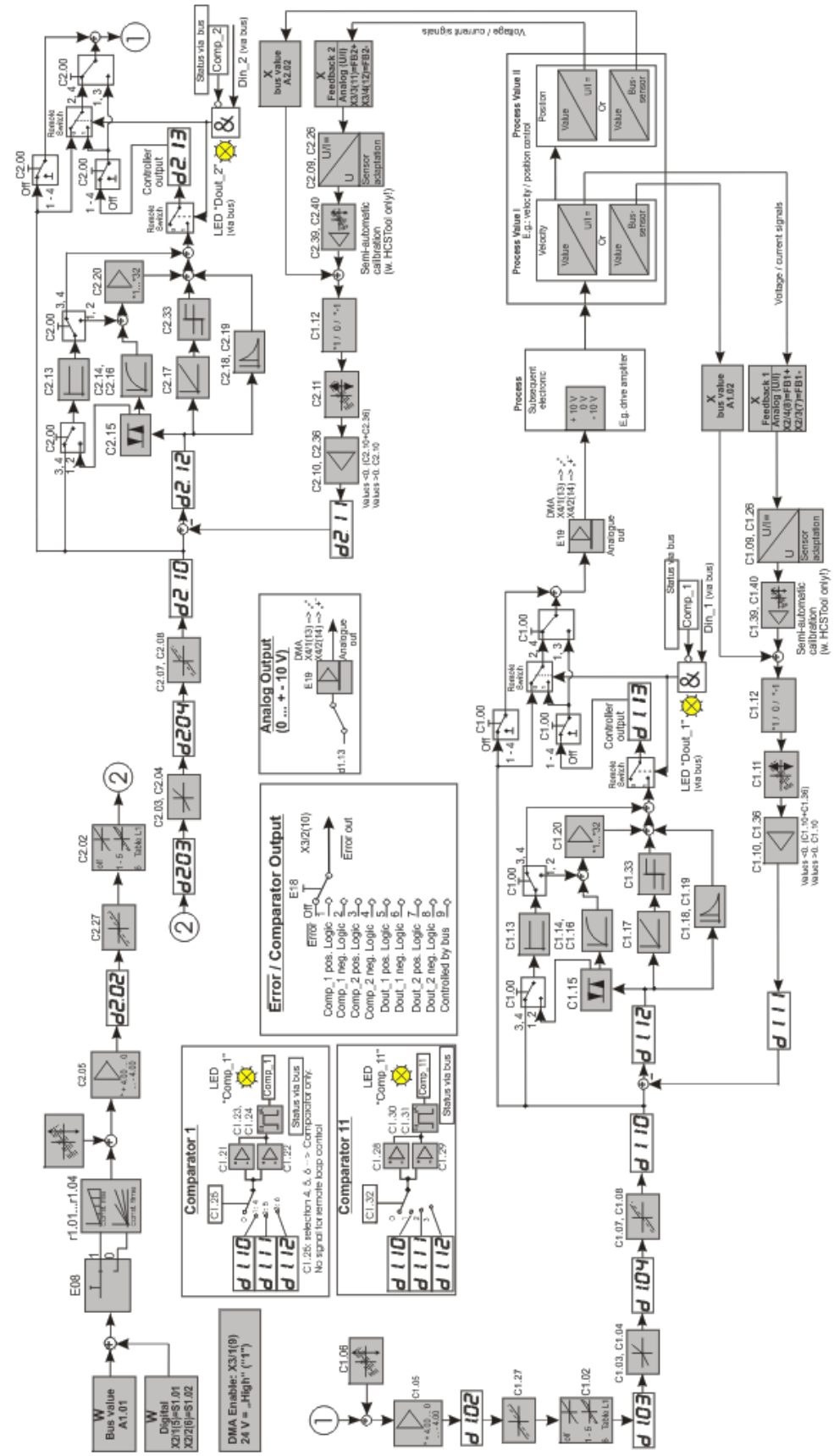



Figure 22: Block diagram mode 11

## 4.10 Inputs

Digital Input, normally used as enable signal.
For more information see □ Chapter „7.6 Digital inputs“, page 51

## 4.11 Outputs:

- for error indication. In case of error this line is connected to ground with 1K input resistance. See □ Chapter „7.7 Digital outputs”, page 52

Options (only possible if the hardware offers the possibility)

- Internal supply output:
  Output for internal sensor or potentiometer
  10 V / 10 mA, current limited. (Output voltage defined by hardware revision)
- Analog voltage output:
  Output for checking the internal control loop values. Output for control an external controller.
  +-10 V / 6 mA

## 4.12 Front View: address selection and connectors

### 4.12.1 Single module Version (node with one module connected)

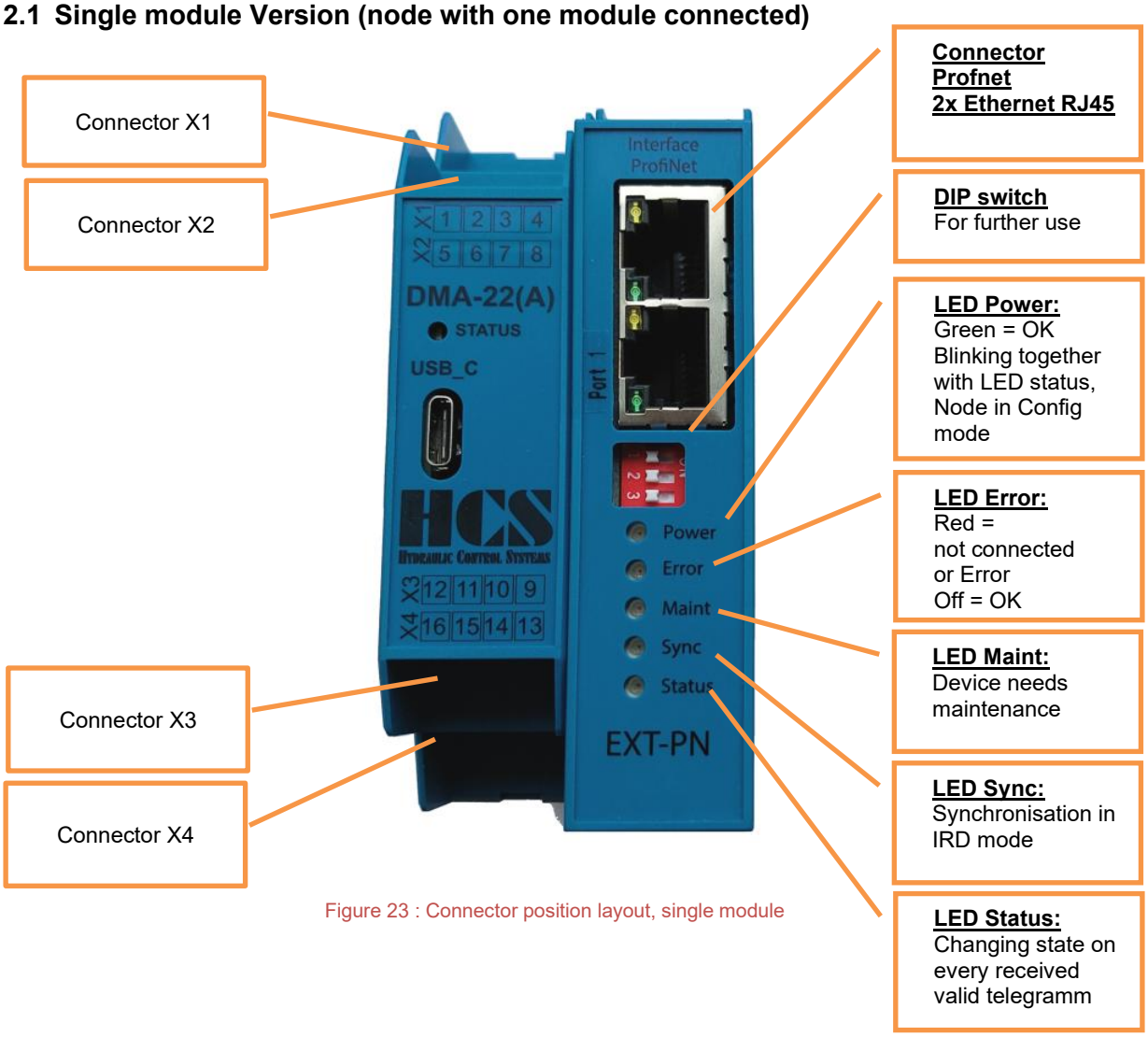



Figure 23 : Connector position layout, single module

### 4.12.2 Multiple module Version (node with multiple modules connected)

Example:
1 Profinet-Node and 3 DMA-22(A) Modules (Slaves) connected

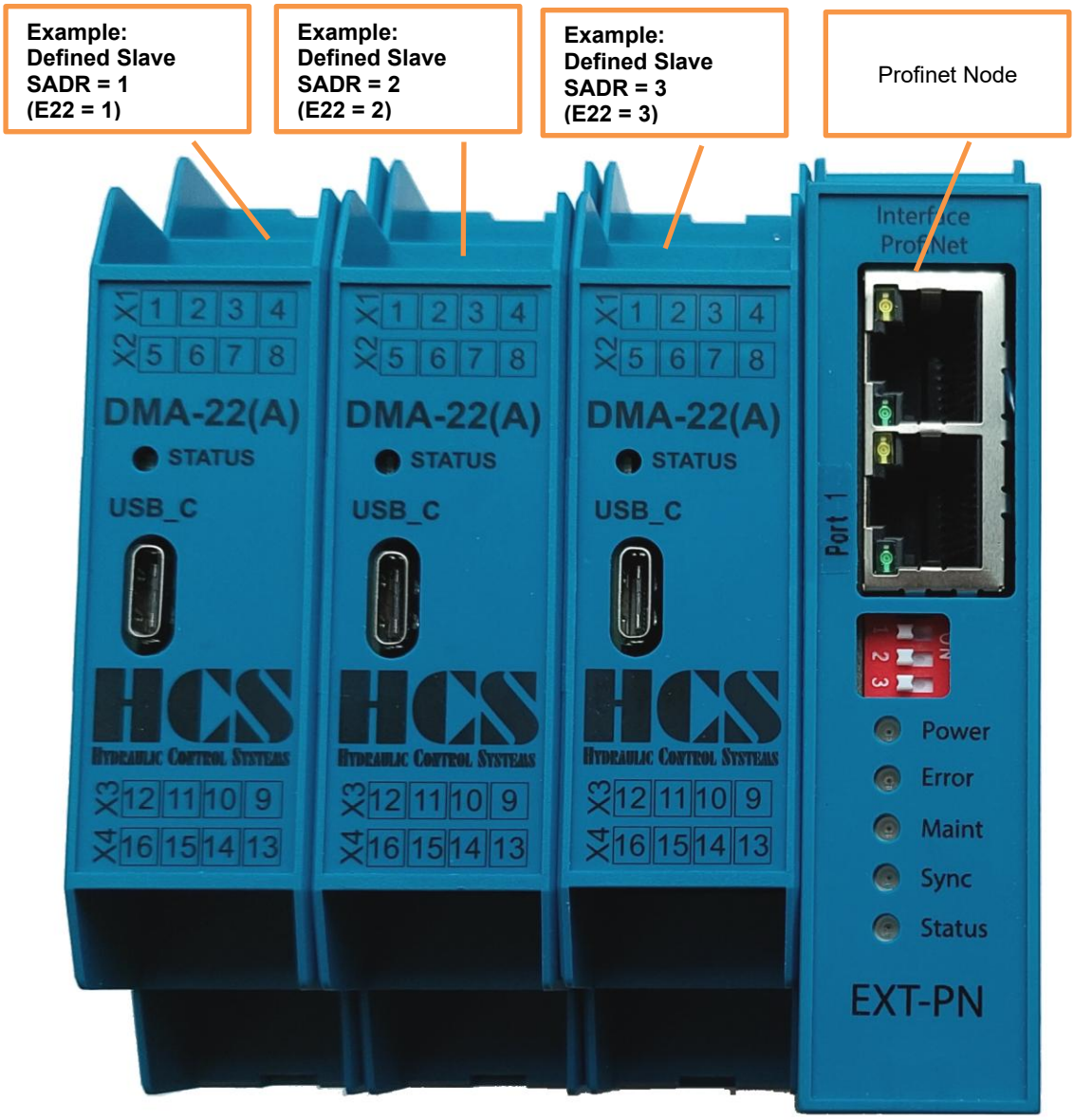



Figure 24 : Connector position layout, single module

# 5 General communication information

## 5.1 Special information for Multi-Slave Nodes

In cases where a Profinet node has more than one slave (modules) connected all slaves must be active (connected to the power supply). The hardware enable is connected to all of the DMA modules. Otherwise communication is not possible and the node-module will respond with a timeout error message.
The Profinet-Node connected to the PLC (Error LED is off).
If 3 modules are used then E22 must be set to „1“ for the first module on the left, to „2“ for the second module in the middle and to „3“ for the third module closest to the slave.

## 5.2 General information about TADR (Telegram Address)

TADR is a value which may be written (changed) with each cycle but changing of this value is optional and not mandatory. The value will be defined by the Profinet-Master.
The purpose is to either check the telegrams by the master or also in order to force sending of a new telegram from the node.
The design of the node is made in a way that the slave or slaves (module or modules in case of multi-slave version) are only responding if a received message from the master contains changed (different) data compared to the previous message.
So TADR can be used to force a response from a slave for example in order to get an update on the current status of the slave even if the other data in the message remain unchanged.
TADR value is valid in the range of 0 to 255 or in hexadecimal 0x00 to 0xFF.

## 5.3 General information about CMD (Command) and SADR (Slave Address)

Different ways of communication between master and slaves (modules) are possible. The difference is in the CMD instruction.

CMD = 6: Writing of a single parameter-ID of a slave (module) at a node. The selected SADR in the telegram will define which slave is selected for the communication. In the slave itself the address is set by parameter E22,
Special case: the Profinet node has only one slave (module) connected. In this case SADR is set to 1 (defined in parameter E22). SADR also could be set to any value between 1 and 32.
Theoretically up to 32 slaves (modules) can be connected to one of the Profinet nodes. Which of the slaves (modules) is selected for communication is again defined in the telegram by the SADR and in the slaves (modules) by setting of parameters E22. For more information please refer to □ Chapter „6.3.1 CMD = 6, Master Write single parameter (7 bytes) ”, page 36

CMD = 3: Reading of one or more parameters with parameter-ID in rising order. All other definitions from CMD = 6 are also applicable. For more information please refer to □ Chapter „6.1.1 CMD = 3, Master Read parameters (7 bytes)”, page 34

CMD = 15: This instruction allows writing or reading of a (pre-defined) set of parameters. The set of parameters itself is depending on the software version of the slaves (modules); refer also to □ Chapter „7.5 CMD = 15, structure of Profinet telegram definition, depending on SW Versions”, page 43
With this instruction all slaves (modules) at a given node with the SADR from 1 to 5 can be reached. This instruction cannot be used for slaves (modules) with a SADR > 5. In this case CMD = 3 and CMD = 6 must be used!
The best is to set SADR at the slaves in rising and consecutive order beginning with SADR =1 (E22 = 1).
In the telegram itself the number of slaves (modules) at the node is defined by SNUM. This at the same time, will also define the length of the telegram itself (number of bytes in the telegram).
For more information please to □ Chapter „7 Complex commands”, page 39

It is possible to „mix“ the instructions CMD = 3, CMD = 6 and CM = 15 for one node.
So, for example the slaves (modules) with E22 =1, E22 = 2 and E22 = 3 are actuated by CMD =15 and/or CMD = 3 and/or CMD =6,
But another slave (module) with address E22 greater than 5 is only actuated either by CMD =3 or CMD = 6.

# 6 Simple commands

## 6.1 CMD 3 Master Read

### 6.1.1 CMD = 3, Master Read parameters (7 bytes)

Write first…

| Byte | Abbreviation | Description |
|---|---|---|
| 0 | TADR | Telegram address, defined by the user, Range 0..255, 0..0xFF |
| 1 | SADR | Slave address |
| 2 | CMD | Command: 3 = Read multiple parameters by the master |
| 3 | IDH | High byte parameter ID of first parameter |
| 4 | IDL | Low byte parameter ID of first parameter |
| 5 | N-high | High-Byte Number parameters |
| 6 | N-low | Low-Byte Number parameters |

…Write last

**TADR** Telegram address, defined by the Profinet-Master.( Set by the User )

**SADR** Slave address, if more than one Slave is connected to the Profinet port
1 = standard (single slave)
2 to 32 = also possible (E22 defines the slave address)

**CMD** Command: 3 = Read parameter by the master

**IDH,IDL** H-Byte and Low-Byte of parameter-ID in HEX
0x0000 .. 0x0109 = Valid ID-Range. Please take into account the gaps in the parameter list

**N-high** High-Byte Number parameters (words), normally zero

**N-low** Low-Byte Number parameters (words), 1 to 8 (0x08)
(maximal 8 parameters readable at once)

### 6.1.2 CMD = 3, Response from Slave (4 + CNT bytes)

Write first…

| Byte | Abbreviation | Description |
|---|---|---|
| 0 | TADR | Telegram address, defined by the read command, Range 0..255, 0..0xFF |
| 1 | SADR | Slave address |
| 2 | CMD | Command: 3 = Read multiple parameters by the master |
| 3 | CNT | N*2, Number of data bytes |
| 4 | DAT1H | High byte value of first parameter |
| 5 | DAT1L | Low Byte value of first parameter |

If CNT > 2: ……

| | | |
|---|---|---|
| CNT+2 = N | DATnH | High byte value of last (n)parameter |
| CNT+3 = N | DATnL | Low Byte value of last (n)parameter |

…Write last

**TADR** Telegram address, defined by the Profinet-Master. (Set by the read command)

**SADR** Slave address, if more than one Slave is connected to the Profinet port
1 = standard (single slave)
2 to 32 = also possible (E22 defines the slave address)

**CMD** Command: 3 = Read parameter by the master

**CNT** N*2, Number of data bytes (Maximal 16 data bytes)

**IDH,IDL** H-Byte and Low-Byte of parameter-ID in HEX
0x0000 .. 0x0109 = Valid ID-Range

**DAT1H.. DAT nH**,
**DAT1L.. DAT nL**, Two data bytes in HEX, without decimal sign
0x7FFF == +32767
0x8000 == -32768

### 6.1.3 CMD = 3, Error from Slave (4 bytes)

Write first…

| Byte | Abbreviation | Description |
|---|---|---|
| 0 | TADR | Telegram address, defined by the read command, Range 0..255, 0..0xFF |
| 1 | SADR | Slave address |
| 2 | ERR | Error code |
| 3 | EXCE | Exception |

…Write last

**ERR** Error code: 0x83 = Read failed

**EXCE** Exception:
1 = Command not supported
2 = Wrong ID
3 = CNT== 0 or CNT > 16 (0x10)
4 = Reading of ID failed
5 = timeout slave, address wrong or not installed or
Invalid number of connected slaves
6 = Internal checksum error (Modbus)

Example for error response:

Module 3 is not installed and **SADR** in the master telegram is set to 3

| Byte | Data | Abbreviation | Description |
|---|---|---|---|
| 0 | 0x23 | TADR | Response telegram address (defined by the read command) |
| 1 | **0x03** | SADR | Actual slave address |
| 2 | 0x83 | ERR | Error, Bit „7“ with command 0x03 is set |
| 3 | 0x05 | EXCE | timeout slave, address wrong or not installed |

## 6.2 Telegram example CMD = 3

**Master wants to read parameter „d1.07”:**

ID = 0x0007
N = 1, one parameter
TADR = 0x24
SADR = 0x01

| Byte | Data | Abbreviation |
|---|---|---|
| 0 | 0x24 | TADR |
| 1 | 0x01 | SADR |
| 2 | 0x03 | CMD |
| 3 | 0x00 | IDH |
| 4 | 0x07 | IDL |
| 5 | 0x00 | N-high |
| 6 | 0x01 | N-low |

Response from slave (module)
CNT = 2 (2 Bytes)
DATA = 0x0133 (==0.307Af)

| Byte | Data | Abbreviation |
|---|---|---|
| 0 | 0x24 | TADR |
| 1 | 0x01 | SADR |
| 2 | 0x03 | CMD |
| 3 | 0x02 | CNT |
| 4 | 0x01 | DAT1H |
| 5 | 0x33 | DAT1L |

## 6.3 CMD = 6, Master Write

### 6.3.1 CMD = 6, Master Write single parameter (7 bytes)

Write first…

| Byte | Abbreviation | Description |
|---|---|---|
| 0 | TADR | Telegram address, defined by the user, Range 0..255, 0..0xFF |
| 1 | SADR | Slave address |
| 2 | CMD | Command: 6 = Write single parameter by the master |
| 3 | IDH | High byte parameter ID |
| 4 | IDL | Low byte parameter ID |
| 5 | DATH | High byte of parameter value |
| 6 | DATL | Low byte of parameter value |

…Write last

**TADR** Telegram address, defined by the Profinet-Master.(Set by the user)

**SADR** Slave address, if more than one Slave is connected to the Profinet port
1 = standard (single slave)
2..32 = also possible (E22 defines the slave address)

**CMD** Command: 6 = Write single parameter by the master

**IDH,IDL** H-Byte and Low-Byte of parameter-ID in HEX
0x0000 .. 0x0109 = Valid ID-Range. Please take into account the gaps in the parameter list

**DATH,DATL** Two data bytes in HEX, without decimal sign
0x7FFF == +32767
0x8000 == -32768

### 6.3.2 CMD = 6, Response from Slave (7 bytes)

Write first…

| Byte | Abbreviation | Description |
|---|---|---|
| 0 | TADR | Telegram address, defined by the write command, Range 0..255, 0..0xFF |
| 1 | SADR | Slave address |
| 2 | CMD | Command: 6 = Write single parameter by the master |
| 3 | IDH | High byte parameter ID |
| 4 | IDL | Low byte parameter ID |
| 5 | DATH | High byte of second analogue value |
| 6 | DATL | Low byte of first analogue value |

…Write last

**TADR** Telegram address, defined by the Profinet-Master.( Set by the write command )

**SADR** Slave address, if more than one Slave is connected to the Profinet port
1 = standard (single slave)
2..32 = also possible (E22 defines the slave address)

**CMD** Command: 6 = Write single parameter by the master
**IDH,IDL** H-Byte and Low-Byte of parameter-ID in HEX
0x0000 .. 0x0109 Valid ID-Range

**DATH,DATL** Two data bytes in HEX, without decimal sign
0x7FFF == +32767
0x8000 == -32768

### 6.3.3 CMD = 6, Error from Slave (4 bytes)

Write first…

| Byte | Abbreviation | Description |
|---|---|---|
| 0 | TADR | Telegram address, defined by the write command, Range 0..255, 0..0xFF |
| 1 | SADR | Slave address |
| 2 | ERR | Error code |
| 3 | EXCE | Exception |

…Write last

**ERR** Error code: 0x86 = write failed

**EXCE** Exception:
1 = Command not supported
2 = Wrong ID
3 = Wrong Data Value
4 = Writing of ID failed
5 = timeout slave, address wrong or not installed or
Invalid number of connected slaves
6 = Internal checksum error (Modbus)

Example for error response:

Module 3 is not installed and **SADR** in the master telegram is set to 3

| Byte | Data | Abbreviation | Description |
|---|---|---|---|
| 0 | 0x23 | TADR | Response telegram address (defined by the write command) |
| 1 | **0x03** | SADR | Actual slave address |
| 2 | 0x86 | ERR | Error, Bit „7“ with command 0x06 is set |
| 3 | 0x05 | EXCE | timeout slave, address wrong or not installed |

## 6.4 Telegram example CMD = 6

### 6.4.1 Example, write parameter

**Master wants to write parameter „C1.07” with value 1.000 V:**

ID = 0x0030
DATA = 0x3E8 ( = 1000 in decimal)
TADR = 0x23
SADR = 0x01

| Byte | Data | Abbreviation |
|---|---|---|
| 0 | 0x23 | TADR |
| 1 | 0x01 | SADR |
| 2 | 0x06 | CMD |
| 3 | 0x00 | IDH |
| 4 | 0x30 | IDL |
| 5 | 0x03 | DATL |
| 6 | 0xE8 | DATH |

Response from slave (module), command was successful executed
ID = 0x0030
DATA = 0x3E8 ( = 1.000 V)

| Byte | Data | Abbreviation |
|---|---|---|
| 0 | 0x23 | TADR |
| 1 | 0x01 | SADR |
| 2 | 0x06 | CMD |
| 3 | 0x00 | IDH |
| 4 | 0x30 | IDL |
| 5 | 0x03 | DATL |
| 6 | 0xE8 | DATH |

### 6.4.2 Example, write set value

**Examples for set value via Profinet (ID = 0x0027):**

1.) Command signal 0.000 V = 0x0000:

| Byte | Data | Abbreviation |
|---|---|---|
| 0 | 0x12 | TADR |
| 1 | 0x01 | SADR |
| 2 | 0x06 | CMD |
| 3 | 0x00 | IDH |
| 4 | 0x27 | IDL |
| 5 | 0x00 | DATL |
| 6 | 0x00 | DATH |

2.) Set value 5.000 V = 0x1388:

| Byte | Data | Abbreviation |
|---|---|---|
| 0 | 0x12 | TADR |
| 1 | 0x01 | SADR |
| 2 | 0x06 | CMD |
| 3 | 0x00 | IDH |
| 4 | 0x27 | IDL |
| 5 | 0x13 | DATL |
| 6 | 0x88 | DATH |

3.) Set value 9.999 V = 0x270F:

| Byte | Data | Abbreviation |
|---|---|---|
| 0 | 0x12 | TADR |
| 1 | 0x01 | SADR |
| 2 | 0x06 | CMD |
| 3 | 0x00 | IDH |
| 4 | 0x27 | IDL |
| 5 | 0x27 | DATL |
| 6 | 0x0F | DATH |

4.) Set value –9.999 V = 0xD8F1

| Byte | Data | Abbreviation |
|---|---|---|
| 0 | 0x12 | TADR |
| 1 | 0x01 | SADR |
| 2 | 0x06 | CMD |
| 3 | 0x00 | IDH |
| 4 | 0x27 | IDL |
| 5 | 0xD8 | DATL |
| 6 | 0xF1 | DATH |

# 7 Complex commands

## 7.1 CMD = 15, Master writes multiple parameters (3+5·module bytes) (Fast multi-slave command)

Parameter E22 in the related DMA-22(A)-module has to be set to the correct value within the range of 1 to 5 depending on the number of connected slaves (modules). The slave at the left side has the address „1" and the slave closest to the Profinet node will have the highest address setting.

Example: If 3 modules are used than E22 must be set to „1" for the first module on the left, to „2" for the second module in the middle and to „3" for the third module closest to the Profinet node
See □ Chapter "4.12.2 Multiple module Version (node with multiple modules connected)", page 32

Response telegrams are automatically multiplexed by the Profinet node.
See □ Chapter "7.4.2 Multiple-slave (3 DMA modules)", page 41.

Write first…

| Byte | Structure | | | | | Abbreviation | Description | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0..0xFF | | | | | TADR | Telegram address, defined by the user, Range 0..255, 0..0xFF | |
| 1 | 1 | 2 | 3 | 4 | 5 | SNUM | Number of connected slaves (Maximum 5) | |
| 2 | 15 | | | | | CMD | Command: 15 = Write multiple parameter by the master | |
| 3 | 1 | 1 | 1 | 1 | 1 | DAT1.1 | Control-Byte | Byte for special functions |
| 4 | | | | | | DAT1.2H | Process Value1 | High byte of first „analogue" value |
| 5 | | | | | | DAT1.2L | | Low byte of first „analogue" value |
| 6 | | | | | | DAT1.3H | Process Value2 | High byte of second „analogue" value |
| 7 | | | | | | DAT1.3L | | Low byte of second „analogue" value |
| 8 | | 2 | 2 | 2 | 2 | DAT1.1 | Control-Byte | Byte for special functions |
| 9 | | | | | | DAT1.2H | Process Value1 | High byte of first „analogue" value |
| 10 | | | | | | DAT1.2L | | Low byte of first „analogue" value |
| 11 | | | | | | DAT1.3H | Process Value2 | High byte of second „analogue" value |
| 12 | | | | | | DAT1.3L | | Low byte of second „analogue" value |
| 13 | | | 3 | 3 | 3 | DAT3.1 | Control-Byte | Byte for special functions |
| 14 | | | | | | DAT3.2H | Process Value1 | High byte of first „analogue" value |
| 15 | | | | | | DAT3.2L | | Low byte of first „analogue" value |
| 16 | | | | | | DAT3.3H | Process Value2 | High byte of second „analogue" value |
| 17 | | | | | | DAT3.3L | | Low byte of second „analogue" value |
| 18 | | | | 4 | 4 | DAT4.1 | Control-Byte | Byte for special functions |
| 19 | | | | | | DAT4.2H | Process Value1 | High byte of first „analogue" value |
| 20 | | | | | | DAT4.2L | | Low byte of first „analogue" value |
| 21 | | | | | | DAT4.3H | Process Value2 | High byte of second „analogue" value |
| 22 | | | | | | DAT4.3L | | Low byte of second „analogue" value |
| 23 | | | | | 5 | DAT5.1 | Control-Byte | Byte for special functions |
| 24 | | | | | | DAT5.2H | Process Value1 | High byte of first „analogue" value |
| 25 | | | | | | DAT5.2L | | Low byte of first „analogue" value |
| 26 | | | | | | DAT5.3H | Process Value2 | High byte of second „analogue" value |
| 27 | | | | | | DAT5.3L | | Low byte of second „analogue" value |
| | 8 | 13 | 18 | 23 | 28 | Overall telegram length in bytes | | |

**TADR** Telegram address, defined by the Profinet-Master. (defined by the user) Range 0..255

**SNUM** Number of connected slaves (Maximum 5).
Here for multi slave operation the number of present slaves has to be set properly.

**CMD** Command: 15 = Write multiple parameters by the master

**DATx[1].1** Control-Byte. One data byte in HEX, for special functions
0x00 == 0
0xFF == 255

**DATx[1].2H, DATx[1].2L** Process Value1: Two data bytes in HEX, without decimal sign, for „analogue" values
0x7FFF == +32767
0x8000 == -32768

**DATx[1].3H, DATx[1].3L** Process Value1: Two data bytes in HEX, without decimal sign, for „analogue" values
0x7FFF == +32767
0x8000 == -32768

x[1].= Module address (set by parameter E22 in each DMA-22(A)-Module) range 1 to 5.

## 7.2 CMD = 15, Response from Slave (9 bytes)

Please pay attention to the fact that in case of multiple slave operation SADR will be changed automatically with each new response.

Example:
When using 3 slaves than the slave with the module address „1" (E22 = 1) will respond first. After the next „Master-Write" the slave with module address „2" (E22 = 2) will respond and finally the last slave with address „3" (E22 = 3). Then it starts again with slave „1".

This means that the multiplexer for answering is implemented in the Profinet node. SADR indicates which of the modules has actually send the response.

Write first…

| Byte | Structure | Abbreviation | Description | |
|---|---|---|---|---|
| 0 | Telegram header | TADR | Telegram address, defined by the user, Range 0..255, 0..0xFF | |
| 1 | | SADR | Slave address | |
| 2 | | CMD | Command: 15 = Write multiple parameter by the master | |
| 3 | Data of SADR | DAT.1H | Status Word | High byte of module state |
| 4 | | DAT.1L | | Low byte of module state |
| 5 | | DAT.2H | VALUE1 | High byte of first „analogue" value |
| 6 | | DAT.2L | | Low byte of first „analogue" value |
| 7 | | DAT.3H | VALUE2 | High byte of second „analogue" value |
| 8 | | DAT.3L | | Low byte of second „analogue" value |

…Write last

**TADR** Telegram address, defined by the Profinet-Master.(set by the user)

**SADR** Slave address, if more than one Slave is connected to the Profinet port
1 to max 5 depending on number of slaves

**CMD** Command: 15 = Write multiple parameters by the master

**DAT.1H,DAT.1L** Status word. Two data bytes in HEX, , for module state
0x0000 == 0
0xFFFF == 65535

**DAT.2H, DAT.2L,**
**DAT.3H, DAT.3L** VALUE1
VALUE2. Two data bytes in HEX, without decimal sign, for analogue values
0x7FFF == +32767
0x8000 == -32768

## 7.3 CMD = 15, Error from Slave (4 bytes)

Write first…

| Byte | Abbreviation | Description |
|---|---|---|
| 0 | TADR | Telegram address, defined by the write command, Range 0..255, 0..0xFF |
| 1 | SADR | Slave address |
| 2 | ERR | Error code |
| 3 | EXCE | Exception |

…Write last

**ERR** Error code: 0x8F = write failed

**EXCE** Exception:
1 = Command not supported
2 = Wrong ID
3 = Wrong Data Value
4 = Writing of ID failed
5 = timeout (Modbus)
6 = Internal checksum error (Modbus)
7 = Invalid number of connected slaves
8 = timeout slave, address wrong or not installed
9 = Profinet node (SADR = 0) send, received data length to small. The data length of the selected GSD module is too small.

Example for error response:

Module 3 is not installed and SNUM in the master telegram is set to 3

| Byte | Data | Abbreviation | Description |
|---|---|---|---|
| 0 | 0x23 | TADR | Response telegram address (defined by the write command) |
| 1 | **0x03** | SADR | Actual slave address |
| 2 | 0x8F | ERR | Error, Bit „7" with command 0x0F in decimal 15 is set |
| 3 | 0x08 | EXCE | timeout slave, address wrong or not installed |

## 7.4 Examples, CMD = 15

### 7.4.1 Single slave

E22 of the DMA-22(A) module is set to 1.
Hardware Enable is connected to the DMA-22(A) module.

**Master writes set value of 1.000 V to slave (module):**

| Byte | Data | Abbreviation | Description |
|---|---|---|---|
| 0 | 0x23 | TADR | Telegram address, defined by the user, Range 0..255, 0..0xFF |
| 1 | 0x01 | SNUM | 0x01 means single slave |
| 2 | 0x0F | CMD | 0x0F in decimal 15 |
| 3 | 0x00 | DAT1.1 | **Control byte** , 0x00 (default value, no bus-disable) |
| 4 | (1000) | DAT1.2H | **Process Value1 = A1.01, set value branch 1** |
| 5 | | DAT1.2L | 0x03E8 ( = 1000 in decimal = 1.000V) |
| 6 | (0) | DAT1.3H | **Process Value2 = A1.02, feedback value branch 1** |
| 7 | | DAT1.3L | 0x0000 ( = 0 in decimal = 0.000V) |

**Response slave (module) no.1:**

| Byte | Data | | Abbreviation | Description | |
|---|---|---|---|---|---|
| 0 | 0x23 | | TADR | Response telegram address (defined by the write command) | |
| 1 | 0x01 | | SADR | Actual slave address | |
| 2 | 0x0F | | CMD | 0x0F in decimal 15 | |
| 3 | 0x40 | | DAT1.1H | **Status word** | High byte of module state = 0x40 (HW enable = active) |
| 4 | 0x00 | | DAT1.1L | | Low byte of module state = 0x00 |
| 5 | 0x00 | (0) | DAT1.2H | **VALUE1** = d1.11 | |
| 6 | 0x00 | | DAT1.2L | 0x0000 ( = 0 in decimal = 0.000V ) | |
| 7 | 0x03 | (1000) | DAT1.3H | **VALUE2** = d1.10 | |
| 8 | 0xE8 | | DAT1.3L | 0x03E8 ( = 1000 in decimal = 1.000V) | |

(x) = value in decimal.

### 7.4.2 Multiple-slave (3 DMA modules)

E22 of the three modules is set to 1, 2, 3.
All modules are powered on, hardware enable is connected all of the DMA-22(A)modules.

**Master first write cycle: set value 1.000 V to slave1, 2.000 V to slave2 and - 3.000 V to slave3:**

| Byte | Data | | Abbreviation | Description |
|---|---|---|---|---|
| 0 | 0x23 | | TADR | Telegram address, defined by the user, Range 0..255, 0..0xFF |
| 1 | 0x03 | | SNUM | 0x03 means three slaves (modules) are connected |
| 2 | 0x0F | | CMD | 0x0F in decimal 15 |
| 3 | 0x00 | | DAT1.1 | **Control byte** , 0x00 (default value, no bus-disable) |
| 4 | 0x03 | (1000) | DAT1.2H | **Process Value1 = A1.01, set value branch 1** |
| 5 | 0xE8 | | DAT1.2L | 0x03E8 ( = 1000 in decimal = 1.000V) |
| 6 | 0x00 | (0) | DAT1.3H | **Process Value2 = A1.02, feedback value branch 1** |
| 7 | 0x00 | | DAT1.3L | 0x0000 ( = 0 in decimal = 0.000V) |
| 8 | 0x00 | | DAT2.1 | **Control byte** , 0x00 (default value, no bus-disable) |
| 9 | 0x07 | (2000) | DAT2.2H | **Process Value1 = A1.01, set value branch 1** |
| 10 | 0xD0 | | DAT2.2L | 0x07D0 ( = 2000 in decimal = 2.000V) |
| 11 | 0x00 | (0) | DAT2.3H | **Process Value2 = A1.02, feedback value branch 1** |
| 12 | 0x00 | | DAT2.3L | 0x0000 ( = 0 in decimal = 0.000V) |
| 13 | 0x00 | | DAT3.1 | **Control byte** , 0x00 (default value, no bus-disable) |
| 14 | 0xF4 | (- | DAT3.2H | **Process Value1 = A1.01, set value branch 1** |
| 15 | 0x48 | 3000) | DAT3.2L | 0xF448 ( = - 3000 in decimal = - 3.000V) |
| 16 | 0x00 | (0) | DAT3.3H | **Process Value2 = A1.02, feedback value branch 1** |
| 17 | 0x00 | | DAT3.3L | 0x0000 ( = 0 in decimal = 0.000V) |

(x) = value in decimal.

The node will only send a response if **any** data in the telegram received from the master has be changed! If the data in a telegram have to remain unchanged than a response from the node to a ”Master write” can be enforced due to a change of the data TADR.
The data of the write command will be immediately forwarded to all slaves (modules). Each response telegram to a write command will automatically contain „new data” from the next slave (module). The slaves are automatically multiplexed.

Response telegrams are automatically multiplexed by the DMA-22(A)-Profinet node.
In this example, module 1 answers first

| Byte | Abbreviation | | Abbreviation | Description |
|---|---|---|---|---|
| 0 | 0x23 | | TADR | Response telegram address (defined by the write command) |
| 1 | **0x01** | | SADR | Actual slave address |
| 2 | 0x0F | | CMD | 0x0F in decimal 15 |
| 3 | 0x40 | | DAT1.1H | High byte of module state, Hardware Enable is active, no error |
| 3 | 0x00 | | DAT1.1L | Low byte of module state, not used, in this case always zero |
| 4 | 0x00 | (0) | DAT1.2H | **VALUE1** = d1.11 |
| 5 | 0x00 | | DAT1.2L | 0x0000 ( = 0 in decimal = 0.000V ) |
| 6 | 0x03 | (1000) | DAT1.3H | **VALUE2** = d1.10 |
| 7 | 0xE8 | | DAT1.3L | 0x03E8 ( = 1000 in decimal = 1.000V) |

(x) = value in decimal.

Change telegram address TADR to 0x24 and send same message as before. Next module no. 2 will answer.

| Byte | Abbreviation | | Abbreviation | Description |
|---|---|---|---|---|
| 0 | 0x24 | | TADR | Response telegram address (defined by the write command) |
| 1 | **0x02** | | SADR | Actual slave address |
| 2 | 0x0F | | CMD | 0x0F in decimal 15 |
| 3 | 0x40 | | DAT1.1H | High byte of module state, Hardware Enable is active, no error |
| 3 | 0x00 | | DAT1.1L | Low byte of module state, not used, in this case always zero |
| 4 | 0x00 | (0) | DAT1.2H | **VALUE1** = d1.11 |
| 5 | 0x00 | | DAT1.2L | 0x0000 ( = 0 in decimal = 0.000V ) |
| 6 | 0x07 | (2000) | DAT1.3H | **VALUE2** = d1.10 |
| 7 | 0xD0 | | DAT1.3L | 0x07D0 ( = 2000 in decimal = 2.000V) |

(x) = value in decimal.

Change telegram address TADR to 0x25 and send same message as before. Next module no. 3 will answer.

| Byte | Abbreviation | | Abbreviation | Description |
|---|---|---|---|---|
| 0 | 0x25 | | TADR | Response telegram address (defined by the write command) |
| 1 | **0x03** | | SADR | Actual slave address |
| 2 | 0x0F | | CMD | 0x0F in decimal 15 |
| 3 | 0x40 | | DAT1.1H | High byte of module state, Hardware Enable is active, no error |
| 3 | 0x00 | | DAT1.1L | Low byte of module state, not used, in this case always zero |
| 4 | 0x00 | (0) | DAT1.2H | **VALUE1** = d1.11 |
| 5 | 0x00 | | DAT1.2L | 0x0000 ( = 0 in decimal = 0.000V ) |
| 6 | 0xF4 | (-3000) | DAT1.3H | **VALUE2** = d1.10 |
| 7 | 0x48 | | DAT1.3L | 0xF448 ( = - 3000 in decimal = - 3.000V) |

(x) = value in decimal.

Change telegram address TADR to 0x26 and send same message as before. Next module no. 1 will answer.

| Byte | Abbreviation | | Abbreviation | Description |
|---|---|---|---|---|
| 0 | 0x26 | | TADR | Response telegram address (defined by the write command) |
| 1 | **0x01** | | SADR | Actual slave address |
| 2 | 0x0F | | CMD | 0x0F in decimal 15 |
| 3 | 0x40 | | DAT1.1H | High byte of module state, Hardware Enable is active, no error |
| 3 | 0x00 | | DAT1.1L | Low byte of module state, not used, in this case always zero |
| 4 | 0x00 | (0) | DAT1.2H | **VALUE1** = d1.11 |
| 5 | 0x00 | | DAT1.2L | 0x0000 ( = 0 in decimal = 0.000V ) |
| 6 | 0x03 | (1000) | DAT1.3H | **VALUE2** = d1.10 |
| 7 | 0xE8 | | DAT1.3L | 0x03E8 ( = 1000 in decimal = 1.000V) |

(x) = value in decimal.

## 7.5 CMD = 15, structure of Profinet telegram definition, depending on SW Versions

### 7.5.1 General

#### 7.5.1.1 Error Message (indication by „Error occurred")

Errors which are indicated in the Bit (Error occurred) like for example „Over Current" or „Cable Fraction" can be reset with the following action :

- Set input signal for the hardware enable at the according module at terminal X3/1 (9) to „Low" and then go back to „High"
- Setting and resetting of the Bits (0x80) Bus-Disable

Mandatory condition for resetting an error is that the failure causing the error has been cleared or removed.

**Error numbers:**
The corresponding error numbers can be found in the respective manual of the DMA-22(A) Modules.
Generally, the following error table is valid:

| SLAVE (MODULE) STATUS | | Description |
|---|---|---|
| Hi-Byte (DAT.1H) | Low-Byte (DAT.1L) | |
| 0x0C | 0x01 | Error of Operation |
| 0x0C | 0x02 | Wire break at FB1 (voltage/current depend on setting) |
| 0x0C | 0x03 | Excess current or short circuit at the output stages (peak current >8A!) |
| 0x0C | 0x04 | Wire break at FB2 (voltage/current depend on setting) |
| 0x0C | 0x05 | Wire break at S1.05 (voltage/current depend on setting) |
| 0x0C | 0x06 | Wire break at S1.06 (voltage/current depend on setting) |
| 0x0C | 0x08 | Underload of the solenoid output (if monitoring is activated) |
| 0x0C | 0x0A | Timeout error -no Profinet command recognized- (defined with parameter E23) |
| 0x0C | 0x0B | Over current at feedback FB1 (current signal >22mA) |
| 0x0C | 0x0C | Over current at feedback FB2 (current signal >22mA) |
| 0x0C | 0x0F | Over current at set point S1.05 (current signal >22mA) |
| 0x0C | 0x10 | Over current at set point S1.06 (current signal >22mA) |
| 0x0C | 0x17 | Selected Sensor need autocalibration. |

Table 7: Error numbers

If parameter E 23 has a value not equal to 0 (Profinet timeout is activated). Depending on parameter E 26, there are different responses from the device.

- E 26 = off (0)
  The device reacts with a blinking enable LED about to long time between valid Profinet telegrams. Those versions continue with work if a new valid telegram is detected. No separate acknowledges with the enable signal is needed.
- E 26 = 1
  The device reacts with a shining Error LED about to long time between valid Profinet telegrams. Those versions do not continue with work if a new valid telegram is detected. A separate acknowledge with the enable signal is needed.

Generally, the electronic discards every set point in error case and the outputs will be disabled.

#### 7.5.1.2 Explanation for „Hardware Enable" and „Software Disable":

Generally speaking, the hardware enable must be present („high" level) in order to get a signal (current) at the output stages. Besides the hardware enable a software disable can be used in order to „override" the hardware enable. So even if the hardware signal is present, with the according software command the module can be disabled!

This means that the hardware enable can be hardwired and the software disable can be used in order to drop the output signal to zero or also in order to quit errors.

### 7.5.1.3 Explanation of special functions and module state bits

Attention: Functions and module state bits depend on the used SW version

| | |
|---|---|
| **HW_ENABLE** | The „Enable" signal (HW Enable X3/1 (9)) is activated at the module. |
| **ERROR** | An Error has occurred in the DMA module. Different possible problems can cause this. Please refer to the manual for the according DMA version. |
| **BUS_DISABLE** | Enable signal was reset by means of Profinet (DMA module is disabled).<br>Switching off the HW enable also resets any pending **BUS_DISABLE** |
| **Din_1** | To enable the function for the „remote loop controller" (function switches from open loop to closed loop) via the comparator „Comp_1" is enabled and can be used. Additional mandatory condition: C1.00 must be set to 2 or 4!<br>Remark: d1.12 and d1.13 remain at „0.000 V" as long as the condition for activation „Din_1" has not be set. This will prevent a premature activation of the comparator „Comp_1". |
| **Dout_1/CL** | Indicates that the function „loop controller" for the branch 1 is active. When the module is working in „remote loop", the module is now working in closed loop mode. Additional mandatory conditions for transition from open loop to closed loop:<br>• C1.00 must be set to 2 or 4<br>• Signal Din_1 must be present<br>• Comparator Comp_1 has to be outside of the window, hence signal Comp_1 = 0.<br>The module will remain in closed loop operation as long as signal Din_1 stays on „High". |
| **Comp_1** | Output signal of comparator „Comp_1", s used in order to monitor the signal defined by C1.25.<br>This signal can also be used in order to switches over from open to closed loop:<br>0 = comparator is not active (out of the determined limits and time delays)<br>1 = comparator is active (inside the determined limits and time delays). |
| **Comp_11** | Output signal of comparator „Comp_11", is used in order to monitor the signal defined by C1.32<br>0 = comparator is not active (out of the determined limits and time delays)<br>1 = comparator is active (inside the determined limits and time delays). |
| **Din_2** | To enable the function for the „remote loop controller" (function switches from open loop to closed loop) via the comparator „Comp_2" is enabled and can be used. Additional mandatory condition: C2.00 must be set to 2 or 4!<br>Remark: d2.12 and d2.13 remain at „0.000 V" as long as the condition for activation „Din_2" has not be set. This will prevent a premature activation of the comparator „Comp_2"... |
| **Dout_2** | Indicates that the function „loop controller" for the branch 2 is active. When the module is working in „remote loop", the module is now working in closed loop mode. Additional mandatory conditions for transition from open loop to closed loop:<br>• C2.00 must be set to 2 or 4<br>• Signal Din_2 must be present<br>• Comparator Comp_2 has to be outside of the window, hence signal Comp_2 = 0.<br>The module will remain in closed loop operation as long as signal Din_2 stays on „High". |
| **Comp_2** | Output signal of comparator „Comp_2", s used in order to monitor the signal defined by C2.25.<br>This signal can also be used in order to switches over from open to closed loop:<br>0 = comparator is not active (out of the determined limits and time delays)<br>1 = comparator is active (inside the determined limits and time delays). |
| **Comp_22** | Output signal of comparator „Comp_22", is used in order to monitor the signal defined by C2.32<br>0 = comparator is not active (out of the determined limits and time delays)<br>1 = comparator is active (inside the determined limits and time delays). |
| **TST_CMP** | This bit can be used to test the Set Error/Comp output. When set, the output is set to 24V, independent of the Comp_1/Comp_2 signal. To activate this function the parameter E 18 must be set to a value from 1 to 8 and no error is present. With E 18 = 9 this bit has no effect; the output is then controlled by CNTRL_4 only. |
| **CNTRL_4** | Set Error/Comp output (terminal X3 / 10) directly via bus: 1 = output 24 V, 0 = output 0 V. Effective only with parameter E 18 = 9 (SW version V4x.02* or higher), independent of errors and comparators. |
| **STAT_1..4** | Reserved for special functions |
| **STAT_5** | Status of the Error/Comp output (terminal X3 / 10): 1 = output 24 V, 0 = output 0 V (SW version V4x.02* or higher). |

### 7.5.2 Version for Mode 1 (open loop, one valve with two solenoids)

**Telegram structure example with 1 DMA module (SNUM = 1),**
**Master writes multiple parameters (3+5·module bytes)**

Write first…

| Comment | Byte | Structure | Abbreviation | Description | |
|---|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by User, Range 0..255, 0..0xFF | |
| | 1 | | SNUM | Number of connected slaves (Maximum 5) | |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master | |
| | 3 | Module Data | DAT1.1 | **Control byte** | Byte for special functions |
| | 4 | | DAT1.2H | **A1.01 Module 1** | |
| | 5 | | DAT1.2L | Set value input (Profinet) | |
| | 6 | | DAT1.3H | **A1.02 Module 1 (normally not used)** | |
| | 7 | | DAT1.3L | Set value input (Profinet) | |
| ….Write last | | 8 | Overall telegram length in bytes | | |

Explanation of Byte for special functions (**Control byte**)

| Description | Abbreviation | Function of each bit |
|---|---|---|
| Byte for special functions | DATx.1 | Can be set only if the hardware enable is active.<br>0x01 = **not used**<br>0x04 = **TST_CMP**; Set Error/Comp output direct to 24V (disable signalizing of error output) (only E18 = 1…8, no effect with E18 = 9)<br>0x10 = **not used**<br>0x80 = **BUS_DISABLE** (Module disable via Profinet)<br>In this mode not applicable, reserved:<br>0x02 = CNTRL_1<br>0x08 = CNTRL_2<br>0x20 = CNTRL_3<br>0x40 = CNTRL_4 |
| | | The special function byte will be set to internal 0x00 (reset) when the hardware enable is inactive. |

**Response telegram:**

Write first…

| Comment | Byte | Structure | Abbreviation | Description | |
|---|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by the write command | |
| | 1 | | SADR | Slave address | |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master | |
| | 3 | Data of SADR Module | DAT.1H | **Status word** | High byte of module state |
| | 4 | | DAT.1L | | Low byte of module state |
| | 5 | | DAT.2H | **VALUE1** = d1.07 from module with related SADR | |
| | 6 | | DAT.2L | Actual current A | |
| | 7 | | DAT.3H | **VALUE2** = d1.08 from module with related SADR | |
| | 8 | | DAT.3L | Actual current B | |
| ….Write last | | 9 | Overall telegram length in bytes | | |

Explanation of High and low byte of module state (**Status word**):

| Description | Abbreviation | Explanation of each bit |
|---|---|---|
| High byte of module state | DAT.1H | 0x01 = Digital input **SP1** (S1.01) active *)<br>0x02 = Digital input **SP2** (S1.02) active *)<br>0x04 = **HW_ENABLE**, hardware enable active<br>0x08 = **ERROR**, Error is pending<br>0x10 = Digital input **SP3** (S1.03 active *)<br>0x20 = Digital input **SP4** (S1.04 active *)<br>0x80 = **BUS_DISABLE** is set<br>In this mode not applicable, reserved:<br>0x40 = STAT_5 |
| Low byte of module state | DAT.1L | if „Error occurred" bit is set: error number<br>otherwise:<br>0x01 = **not used**<br>0x02 = **not used**<br>0x04 = **not used**<br>0x08 = **not used**<br>0x10 = **not used**<br>0x20 = **not used**<br>0x40 = **not used**<br>0x80 = **not used** |

*) The designation and availability of the input depends on the module variant.

### 7.5.3 Version for Mode 2 (open loop, two valves with one solenoid each)

**Telegram structure example with 1 DMA module (SNUM = 1),**
**Master writes multiple parameters (3+5·module bytes)**

Write first…

| Comment | Byte | Structure | Abbreviation | Description |
|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by User, Range 0..255, 0..0xFF |
| | 1 | | SNUM | Number of connected slaves (Maximum 5) |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master |
| | 3 | Module Data | DAT1.1 | **Control byte** / Byte for special functions |
| | 4 | | DAT1.2H | **A1.01,** set value input (Profinet) for (Branch 1) |
| | 5 | | DAT1.2L | |
| | 6 | | DAT1.3H | **A2.01,** set value input (Profinet) for (Branch 2) |
| | 7 | | DAT1.3L | |
| ….Write last | | 8 | Overall telegram length in bytes | |

Explanation of Byte for special functions (**Control byte**)

| Description | Abbreviation | Function of each bit |
|---|---|---|
| Byte for special functions | DATx.1 | Can be set only if the hardware enable is active.<br>0x01 = not used<br>0x04 = **TST_CMP**; Set Error/Comp output direct to 24V (disable signalizing of error output) (only E18 = 1…8, no effect with E18 = 9)<br>0x10 = not used<br>0x80 = **BUS_DISABLE** (Module disable via Profinet)<br>In this mode not applicable, reserved:<br>0x02 = CNTRL_1<br>0x08 = CNTRL_2<br>0x20 = CNTRL_3<br>0x40 = CNTRL_4 |
| | | The special function byte will be set to internal 0x00 (reset) when the hardware enable is inactive. |

**Response telegram:**

Write first…

| Comment | Byte | Structure | Abbreviation | Description |
|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by the write command |
| | 1 | | SADR | Slave address |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master |
| | 3 | Data of SADR Module | DAT.1H | **Status word** / High byte of module state |
| | 4 | | DAT.1L | Low byte of module state |
| | 5 | | DAT.2H | **VALUE1** = d1.07 from module with related SADR |
| | 6 | | DAT.2L | Actual current A |
| | 7 | | DAT.3H | **VALUE2** = d1.08 from module with related SADR |
| | 8 | | DAT.3L | Actual current B |
| ….Write last | | 9 | Overall telegram length in bytes | |

Explanation of High and low byte of module state (**Status word**):

| Description | Abbreviation | Explanation of each bit |
|---|---|---|
| High byte of module state | DAT.1H | 0x01 = Digital input **SP1** (S1.01) active *)<br>0x02 = Digital input **SP2** (S1.02) active *)<br>0x04 = **HW_ENABLE**, hardware enable active<br>0x08 = **ERROR**, Error is pending<br>0x10 = Digital input **SP3** (S2.01) active *)<br>0x20 = Digital input **SP4** (S2.02) active *)<br>0x80 = **BUS_DISABLE** is set<br>In this mode not applicable, reserved:<br>0x40 = STAT_5 |
| Low byte of module state | DAT.1L | if „Error occurred" bit is set: error number<br>otherwise:<br>0x01 = not used<br>0x02 = not used<br>0x04 = not used<br>0x08 = not used<br>0x10 = not used<br>0x20 = not used<br>0x40 = not used<br>0x80 = not used |

*) The designation and availability of the input depends on the module variant.

### 7.5.4 Version for Mode 2 with separate enable (open loop, two valves with one solenoid each)

**Telegram structure example with 1 DMA module (SNUM = 1),**
**Master writes multiple parameters (3+5·module bytes)**

Write first…

| Comment | Byte | Structure | Abbreviation | Description | |
|---|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by User, Range 0..255, 0..0xFF | |
| | 1 | | SNUM | Number of connected slaves (Maximum 5) | |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master | |
| | 3 | Module Data | DAT1.1 | **Control byte** | Byte for special functions |
| | 4 | | DAT1.2H | **A1.01,** set value input (Profinet) for (Branch 1) | |
| | 5 | | DAT1.2L | | |
| | 6 | | DAT1.3H | **A2.01,** set value input (Profinet) for (Branch 2) | |
| | 7 | | DAT1.3L | | |
| ….Write last | | 8 | Overall telegram length in bytes | | |

Explanation of Byte for special functions (**Control byte**)

| Description | Abbreviation | Function of each bit |
|---|---|---|
| Byte for special functions | DATx.1 | Can be set only if the hardware enable is active.<br>0x01 = **BusEnable_1**; Set signal to active solenoid output 1(A)<br>0x04 = **TST_CMP**; Set Error/Comp output direct to 24V (disable signalizing of error output) (only E18 = 1…8, no effect with E18 = 9)<br>0x10 = **BusEnable_2**; Set signal to active solenoid output 2(B)<br>0x80 = **BUS_DISABLE** (Module disable via Profinet)<br>In this mode not applicable, reserved:<br>0x02 = CNTRL_1<br>0x08 = CNTRL_2<br>0x20 = CNTRL_3<br>0x40 = CNTRL_4 |
| | | The special function byte will be set to internal 0x00 (reset) when the hardware enable is inactive. |

**Response telegram:**

Write first…

| Comment | Byte | Structure | Abbreviation | Description | |
|---|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by the write command | |
| | 1 | | SADR | Slave address | |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master | |
| | 3 | Data of SADR Module | DAT.1H | **Status word** | High byte of module state |
| | 4 | | DAT.1L | | Low byte of module state |
| | 5 | | DAT.2H | **VALUE1** = d1.07 from module with related SADR | |
| | 6 | | DAT.2L | Actual current A | |
| | 7 | | DAT.3H | **VALUE2** = d1.08 from module with related SADR | |
| | 8 | | DAT.3L | Actual current B | |
| ….Write last | | 9 | Overall telegram length in bytes | | |

Explanation of High and low byte of module state (**Status word**):

| Description | Abbreviation | Explanation of each bit |
|---|---|---|
| High byte of module state | DAT.1H | 0x01 = Digital input Enable**1** (A) active *)<br>0x02 = Digital input Enable**2** (B) active *)<br>0x04 = **HW_ENABLE**, hardware enable active<br>0x08 = **ERROR**, Error is pending<br>0x10 = not used<br>0x20 = not used<br>0x80 = **BUS_DISABLE** is set<br>In this mode not applicable, reserved:<br>0x40 = STAT_5 |
| Low byte of module state | DAT.1L | if „Error occurred" bit is set: error number<br>otherwise:<br>0x01 = **BusEnable_1** active<br>0x02 = not used<br>0x04 = not used<br>0x08 = not used<br>0x10 = **BusEnable_2** active<br>0x20 = not used<br>0x40 = not used<br>0x80 = not used |

### 7.5.5 Version for Mode 3, 4, 10 (closed loop)

**Telegram structure example with 1 DMA module (SNUM = 1), Master writes multiple parameters (3+5·module bytes)**

Write first…

| Comment | Byte | Structure | Abbreviation | Description |
|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by User, Range 0..255, 0..0xFF |
| | 1 | | SNUM | Number of connected slaves (Maximum 5) |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master |
| | 3 | Module Data | DAT1.1 | **Control byte** \| Byte for special functions |
| | 4 | | DAT1.2H | **Process Value1 = A1.01** |
| | 5 | | DAT1.2L | Set value input (Profinet) for (Branch 1) |
| | 6 | | DAT1.3H | **Process Value2 = A1.02** |
| | 7 | | DAT1.3L | Feedback value input (Profinet) for (Branch 1) |
| ….Write last | | 8 | Overall telegram length in bytes | |

Explanation of Byte for special functions (**Control byte**)

| Description | Abbreviation | Function of each bit |
|---|---|---|
| Byte for special functions | DATx.1 | Can be set only if the hardware enable is active.<br>0x01 = **Din_1**; Set signal Din_1 to active<br>0x04 = **TST_CMP**; Set Error/Comp output direct to 24V (disable signalizing of error output) (only E18 = 1…8, no effect with E18 = 9)<br>0x10 = **Din_2**; Set signal Din_2 to active<br>0x80 = **BUS_DISABLE** (Module disable via Profinet)<br>In this mode not applicable, reserved:<br>0x02 = CNTRL_1<br>0x08 = CNTRL_2<br>0x20 = CNTRL_3<br>0x40 = CNTRL_4<br>Special since V4x.02*<br>0x40 = CNTRL_4: Set Error/Comp Output |
| | | The special function byte will be set to internal 0x00 (reset) when the hardware enable is inactive. |

**Response telegram:**

Write first…

| Comment | Byte | Structure | Abbreviation | Description |
|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by the write command |
| | 1 | | SADR | Slave address |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master |
| | 3 | Data of SADR Module | DAT.1H | **Status word** \| High byte of module state |
| | 4 | | DAT.1L | **Status word** \| Low byte of module state |
| | 5 | | DAT.2H | **VALUE1** = d1.11 from module with related SADR |
| | 6 | | DAT.2L | Actual value, feedback value (Branch 1) |
| | 7 | | DAT.3H | **VALUE2** = d1.10 from module with related SADR |
| | 8 | | DAT.3L | Set value (internal value) (Branch 1) |
| ….Write last | | 9 | Overall telegram length in bytes | |

Explanation of High and low byte of module state (**Status word**):

| Description | Abbreviation | Explanation of each bit |
|---|---|---|
| High byte of module state | DAT.1H | 0x01 = Digital input **SP1** (S1.01) active *)<br>0x02 = Digital input **SP2** (S1.02) active *)<br>0x04 = **HW_ENABLE**, hardware enable active<br>0x08 = **ERROR**, Error is pending<br>0x10 = Digital input **SP3** (S2.01) active *)<br>0x20 = Digital input **SP4** (S2.02) active *)<br>0x80 = **BUS_DISABLE** is set<br>In this mode not applicable, reserved:<br>0x40 = STAT_5<br>Special since V4x.02*<br>0x40 = STAT_5: Status of Error/Comp Output |
| Low byte of module state | DAT.1L | if „Error occurred" bit is set: error number<br>otherwise:<br>0x01 = **Din_1** active<br>0x02 = **Dout_1** active<br>0x04 = **Comp_1** active<br>0x08 = **Comp_11** active<br>0x10 = **Din_2** active<br>0x20 = **Dout_2** active<br>0x40 = **Comp_2** active<br>0x80 = **Comp_22** active |

*) The designation and availability of the input depends on the module variant.

### 7.5.6 Version for Mode 6, 11 (two closed loops)

**Telegram structure example with 1 DMA module (SNUM = 1), Master writes multiple parameters (3+5·module bytes)**

Write first…

| Comment | Byte | Structure | Abbreviation | Description |
|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by User, Range 0..255, 0..0xFF |
| | 1 | | SNUM | Number of connected slaves (Maximum 5) |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master |
| | 3 | Module Data | DAT1.1 | **Control byte** Byte for special functions |
| | 4 | | DAT1.2H | **Process Value1 = A1.01** |
| | 5 | | DAT1.2L | Set value input (Profinet) for (Branch 1) |
| | 6 | | DAT1.3H | **Process Value2 = A2.02** |
| | 7 | | DAT1.3L | Feedback value input (Profinet) for (Branch 2) |
| ….Write last | | 8 | Overall telegram length in bytes | |

Explanation of Byte for special functions (**Control byte**)

| Description | Abbreviation | Function of each bit |
|---|---|---|
| Byte for special functions | DATx.1 | Can be set only if the hardware enable is active.<br>0x01 = **Din_1**; Set signal Din_1 to active<br>0x04 = **TST_CMP**; Set Error/Comp output direct to 24V (disable signalizing of error output) (only E18 = 1…8, no effect with E18 = 9)<br>0x10 = **Din_2**; Set signal Din_2 to active<br>0x80 = **BUS_DISABLE** (Module disable via Profinet)<br>In this mode not applicable, reserved:<br>0x02 = CNTRL_1<br>0x08 = CNTRL_2<br>0x20 = CNTRL_3<br>0x40 = CNTRL_4<br>Special since V4x.02*<br>0x40 = CNTRL_4: Set Error/Comp Output |
| | | The special function byte will be set to internal 0x00 (reset) when the hardware enable is inactive. |

**Response telegram:**

Write first…

| Comment | Byte | Structure | Abbreviation | Description |
|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by the write command |
| | 1 | | SADR | Slave address |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master |
| | 3 | Data of SADR Module | DAT.1H | **Status word** High byte of module state |
| | 4 | | DAT.1L | Low byte of module state |
| | 5 | | DAT.2H | **VALUE1** = d1.11 from module with related SADR |
| | 6 | | DAT.2L | Actual value, feedback value (Branch 1) |
| | 7 | | DAT.3H | **VALUE2** = d2.11 from module with related SADR |
| | 8 | | DAT.3L | Actual value, feedback value (Branch 2) |
| ….Write last | | 9 | Overall telegram length in bytes | |

Explanation of High and low byte of module state (**Status word**):

| Description | Abbreviation | Explanation of each bit |
|---|---|---|
| High byte of module state | DAT.1H | 0x01 = Digital input **SP1** (S1.01) active *)<br>0x02 = Digital input **SP2** (S1.02) active *)<br>0x04 = **HW_ENABLE**, hardware enable active<br>0x08 = **ERROR**, Error is pending<br>0x10 = Digital input **SP3** (S2.01) active *)<br>0x20 = Digital input **SP4** (S2.02) active *)<br>0x80 = **BUS_DISABLE** is set<br>In this mode not applicable, reserved:<br>0x40 = STAT_5<br>Special since V4x.02*<br>0x40 = STAT_5: Status of Error/Comp Output |
| Low byte of module state | DAT.1L | if „Error occurred" bit is set: error number<br>otherwise:<br>0x01 = **Din_1** active<br>0x02 = **Dout_1** active<br>0x04 = **Comp_1** active<br>0x08 = **Comp_11** active<br>0x10 = **Din_2** active<br>0x20 = **Dout_2** active<br>0x40 = **Comp_2** active<br>0x80 = **Comp_22** active |

*) The designation and availability of the input depends on the module variant.

### 7.5.7 Version for Mode 8 (closed loop)

**Telegram structure example with 1 DMA module (SNUM = 1), Master writes multiple parameters (3+5·module bytes)**

Write first…

| Comment | Byte | Structure | Abbreviation | Description |
|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by User, Range 0..255, 0..0xFF |
| | 1 | | SNUM | Number of connected slaves (Maximum 5) |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master |
| | 3 | Module Data | DAT1.1 | **Control byte** Byte for special functions |
| | 4 | | DAT1.2H | **Process Value1 = A1.01** |
| | 5 | | DAT1.2L | Set value input (Profinet) for (Branch 1) |
| | 6 | | DAT1.3H | **Process Value2 = A2.01 S** |
| | 7 | | DAT1.3L | Set value input (Profinet) for (Branch 2) |
| ….Write last | | 8 | Overall telegram length in bytes | |

Explanation of Byte for special functions (**Control byte**)

| Description | Abbreviation | Function of each bit |
|---|---|---|
| Byte for special functions | DATx.1 | Can be set only if the hardware enable is active.<br>0x01 = **Din_1**; Set signal Din_1 to active<br>0x04 = **TST_CMP**; Set Error/Comp output direct to 24V (disable signalizing of error output) (only E18 = 1…8, no effect with E18 = 9)<br>0x10 = **Din_2**; Set signal Din_2 to active<br>0x80 = **BUS_DISABLE** (Module disable via Profinet)<br>Not applicable, reserved:<br>0x02 = CNTRL_1<br>0x08 = CNTRL_2<br>0x20 = CNTRL_3<br>0x40 = CNTRL_4<br>Special since V4x.02*<br>0x40 = CNTRL_4: Set Error/Comp Output |
| | | The special function byte will be set to internal 0x00 (reset) when the hardware enable is inactive. |

**Response telegram:**

Write first…

| Comment | Byte | Structure | Abbreviation | Description |
|---|---|---|---|---|
| | 0 | Telegram header | TADR | Telegram address, defined by the write command |
| | 1 | | SADR | Slave address |
| | 2 | | CMD | Command: 15 = Write multiple parameter by the master |
| | 3 | Data of SADR Module | DAT.1H | **Status word** High byte of module state |
| | 4 | | DAT.1L | Low byte of module state |
| | 5 | | DAT.2H | **VALUE1** = d1.11 from module with related SADR |
| | 6 | | DAT.2L | Actual value, feedback value (Branch 1) |
| | 7 | | DAT.3H | **VALUE2** = d2.11 from module with related SADR |
| | 8 | | DAT.3L | Actual value, feedback value (Branch 2) |
| ….Write last | | 9 | Overall telegram length in bytes | |

Explanation of High and low byte of module state (**Status word**):

| Description | Abbreviation | Explanation of each bit |
|---|---|---|
| High byte of module state | DAT.1H | 0x01 = Digital input **SP1** (S1.01) active *)<br>0x02 = Digital input **SP2** (S1.02) active *)<br>0x04 = **HW_ENABLE**, hardware enable active<br>0x08 = **ERROR**, Error is pending<br>0x10 = Digital input **SP3** (S2.01) active *)<br>0x20 = Digital input **SP4** (S2.02) active *)<br>0x80 = **BUS_DISABLE** is set<br>In this mode not applicable, reserved:<br>0x40 = STAT_5<br>Special since V4x.02*<br>0x40 = STAT_5: Status of Error/Comp Output |
| Low byte of module state | DAT.1L | if „Error occurred" bit is set: error number<br>otherwise:<br>0x01 = **Din_1** active<br>0x02 = **Dout_1** active<br>0x04 = **Comp_1** active<br>0x08 = **Comp_11** active<br>0x10 = **Din_2** active<br>0x20 = **Dout_2** active<br>0x40 = **Comp_2** active<br>0x80 = **Comp_22** active |

*) The designation and availability of the input depends on the module variant.

## 7.6 Digital inputs

The DMA-22(A) has three digital inputs

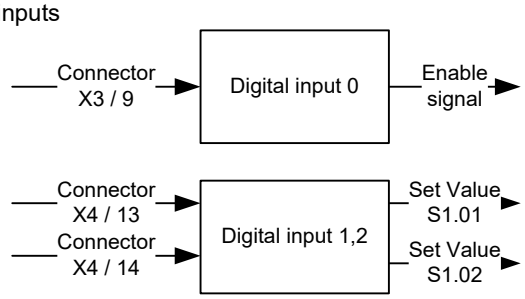



Figure 25: Digital inputs

### 7.6.1 Digital input 0 (enable signal) input 1,2 (set Value selection)

The digital enable signal incorporates the following functions:

- Removes the hardware locking of the output stages
- Start the solenoid current controller
- Start the control loop of branch 1 and/or branch 2

The set value selection signal incorporates the following functions:

- Select the set value of parameter S1.01, S1.02
- Set the signal Din_1, Din_2 when the “remote controller” is selected.

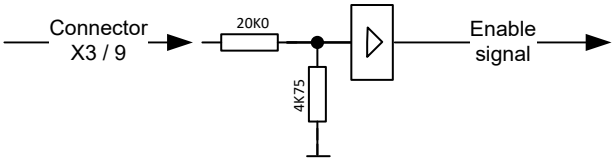



Figure 26: schematic of digital input 0,1,2

| Enable Signal | Description |
|---|---|
| ON | Signal of digital input > 8V |
| OFF | Signal of digital input < 3V |

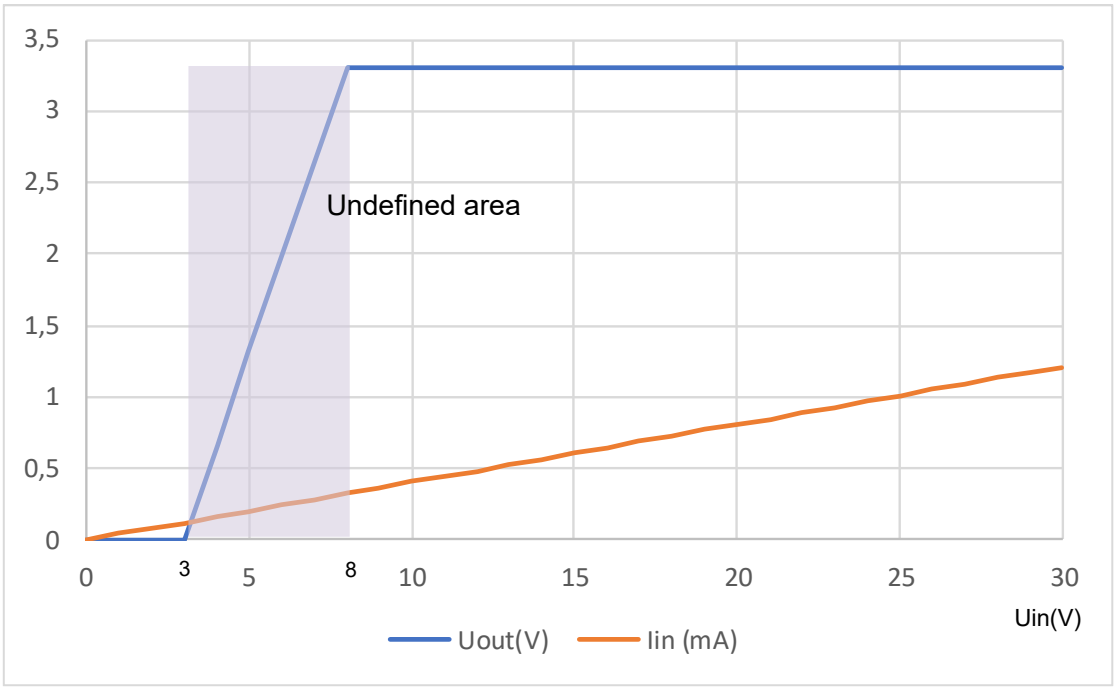



Figure 27: Input Signal curve

## 7.7 Digital outputs

The following digital output is available.

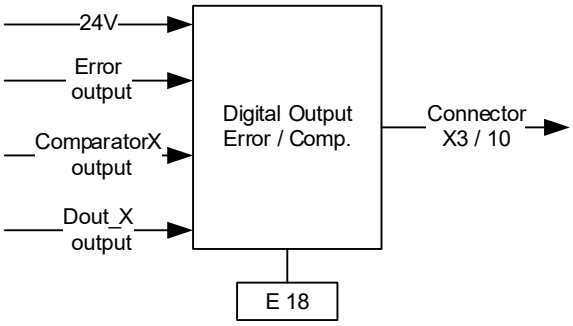



Figure 28: Digital outputs

### 7.7.1 Digital output: \Error or Comparator output

At the terminal X3 / 10 the error output or comparator is available (Depending on setting E18)
If an error occurred, the output is low (negative logic).

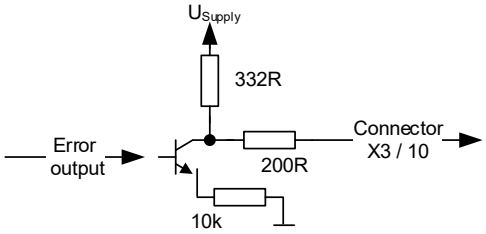



Figure 29: schematic of digital output \Error/comp.

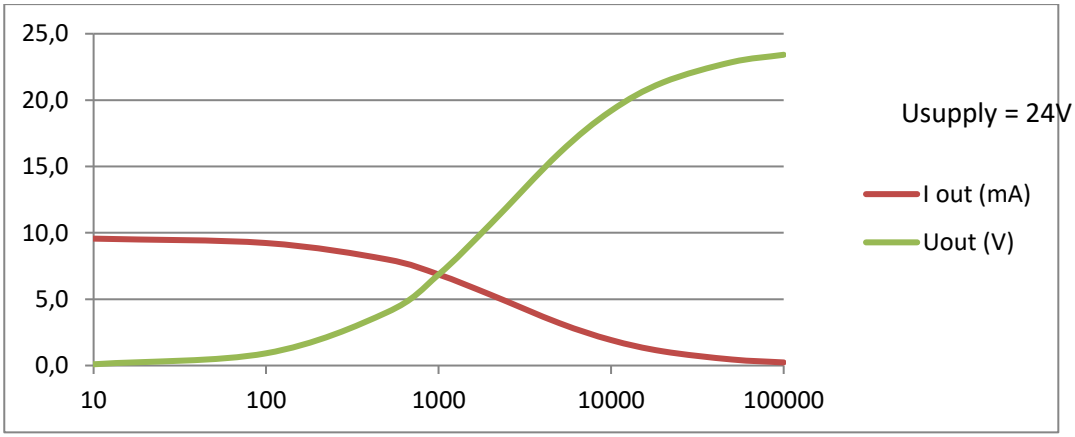



Figure 30: Output high, voltage and current curve at 24V supply

On the output pin X3 / 10 can also be used as a universal output.

**Load capability:**

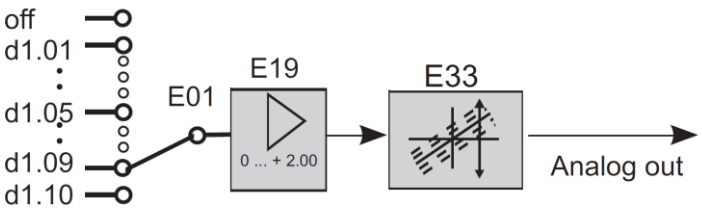
The \Error/Comp output is a signal output, not a power output. It is pulled up to the supply voltage via internal resistors (332 Ω + 200 Ω, see Figure 29). The output current is limited to approx. 10 mA (short circuit) and the output voltage decreases with increasing load (see Figure 30). Without load approx. 23 V are measured at 24 V supply. Do not connect relay coils, solenoid valves or other loads directly to terminal X3 / 10. Use a coupling device with low input current (approx. ≤ 3 mA at 24 V), e.g. an optocoupler / solid-state relay module, an interface relay with electronic input or a 24 V digital input of the PLC. With a load resistance of approx. 10 kΩ or higher, the output voltage is approx. 18 … 23 V.

### 7.7.2 Internal supply output

Normally the output voltage is fixed to 10V ref. (on customer request the output voltage can be changed). The connector pin for the outputis not fix. Please refer the label on the DMA module.

### 7.7.3 Analog voltage output

Maximal output current 6mA.
The connector pin for the output is not fix. Please refer the label on the DMA module.



## 7.8 Status display LEDs

The network and the amplifier states are indicated by light emitting diodes (status display LEDs) on the electronics housing.

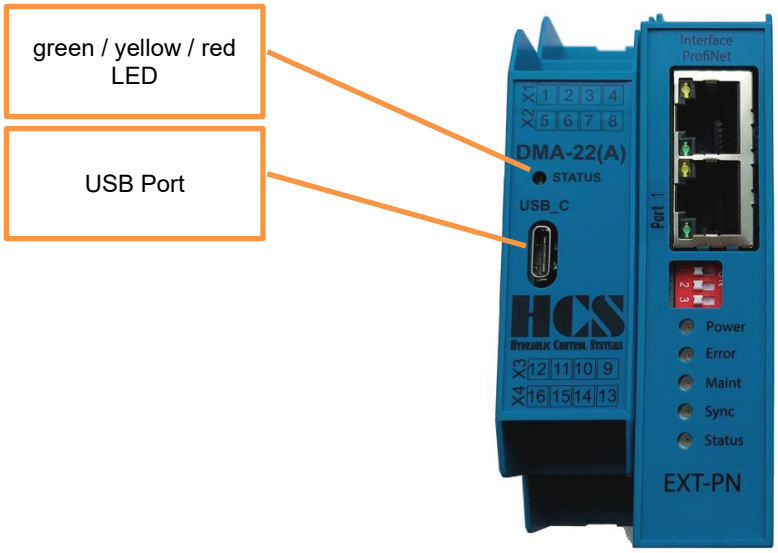



Figure 31: Top view, status LEDs and USB Port

### 7.8.1 LED definition

| LED | Definition |
|---|---|
| ON | constantly ON |
| OFF | constantly OFF |
| Blinking | equal ON and OFF times with a frequency of 1.0 Hz: ON/OFF = 500 ms. |
| Fast Blinking | equal ON and OFF times with a frequency of 4.0 Hz: ON/OFF = 125 ms. |
| Flashing | Different ON and OFF times: ON = 50 ms, OFF = 250 ms. |
| Flicker flash | Different ON and OFF times: ON = 100 ms, OFF = 400 ms. |

Table 8: LED definitions

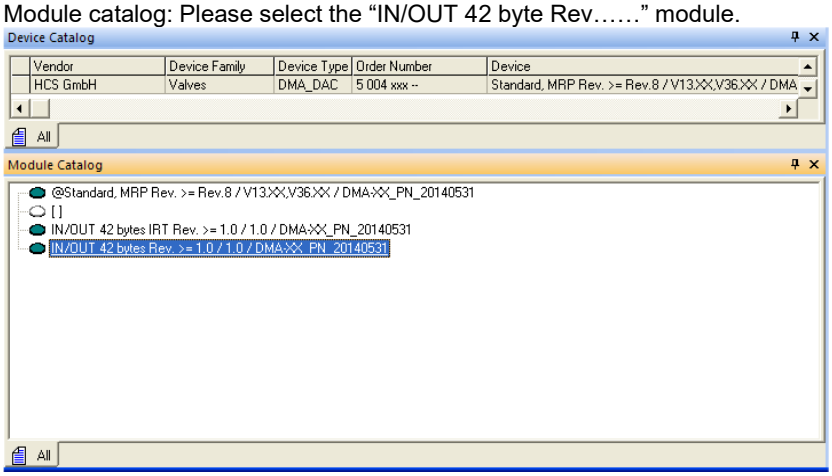
### 7.8.2 LED status definition

Definition « Enable Signal » see □ Chapter “7.6.1 Digital input 0 (enable signal)“, page 51

| EXT-PB | DMA-22(A) | | | Enable Signal | Description |
|---|---|---|---|---|---|
| Red LED | Green LED | Yellow LED | Red LED | | |
| OFF | OFF | OFF | OFF | --- | No power is applied to the amplifier |
| OFF | ON | OFF | OFF | OFF | Power is applied via external source or the USB interface. |
| OFF | OFF | Blinking | OFF | ON | The device is in the semiautomatic sensor calibration process. |
| OFF | OFF | Fast Blinking | OFF | ON | Enable signal is on, and the parameter E 09 is set to a certain value. Or, the device is disabled by bus command. |
| OFF | OFF | Flashing | OFF | ON | Timeout occurred, parameter E 26 is set to 0. (ProfiNet timeout causes no error) |
| OFF | OFF | ON | OFF | ON | Enable of the device is active. |
| -- | OFF | OFF | ON | ON | Internal error occurred. |
| -- | OFF | OFF | Blinking | ON | Internal error occurred, communication with the PROFINET board disturbed. |

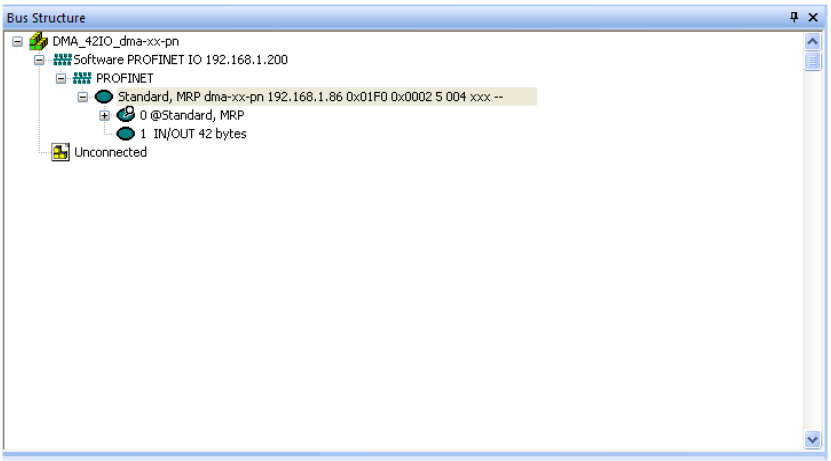
# 8 PLC configurator

Module catalog: Please select the “IN/OUT 42 byte Rev……” module.

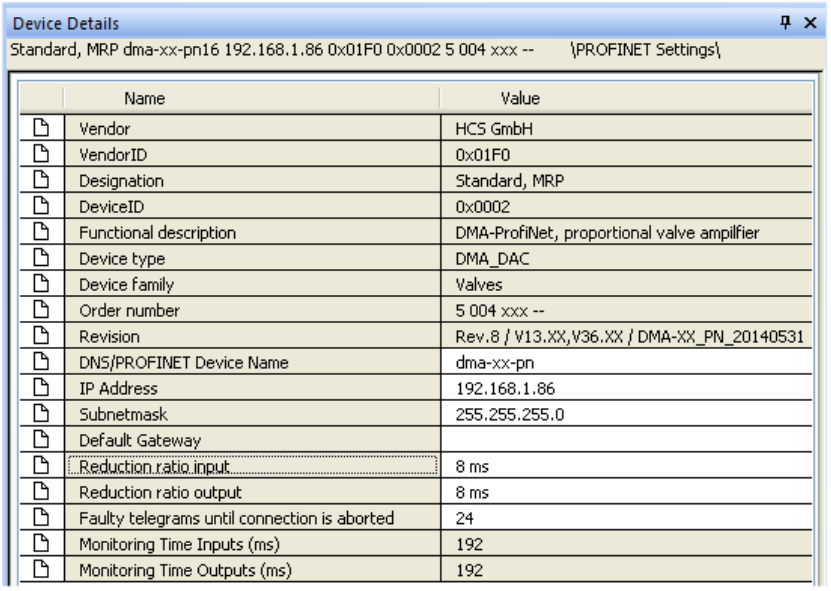



Automatically an 42 byte in- output buffer will be reserved.
So all kind of DMA module configuration can processed.

And insert this module into the bus structure.




Device details:




| Name | Value |
|---|---|
| Vendor | HCS GmbH |
| VendorID | 0x01F0 |
| Designation | Standard, MRP |
| DeviceID | 0x0002 |
| Functional description | DMA-ProfiNet, proportional valve ampilfier |
| Device type | DMA_DAC |
| Device family | Valves |
| Order number | 5 004 xxx -- |
| Revision | Rev.8 / V13.XX,V36.XX / DMA-XX_PN_20140531 |
| DNS/PROFINET Device Name | dma-xx-pn |
| IP Address | 192.168.1.86 |
| Subnetmask | 255.255.255.0 |
| Default Gateway | |
| Reduction ratio input | 8 ms |
| Reduction ratio output | 8 ms |
| Faulty telegrams until connection is aborted | 24 |
| Monitoring Time Inputs (ms) | 192 |
| Monitoring Time Outputs (ms) | 192 |

# 9 Configuration for SIEMENS TIA portal

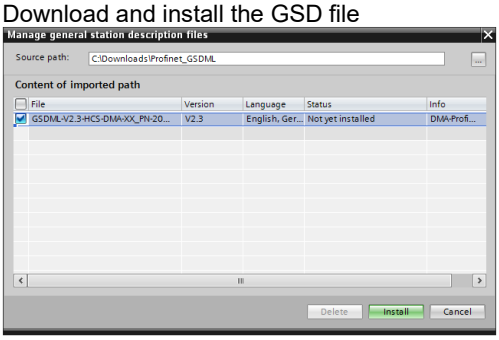
Two Profinet function blocks for the Siemens TIA portal are available, and can be downloaded from the HCS web side.
Which one, the user should take, is depending on the used CPU family.

- Siemens CPU series S7-300,400 PN_DMA_2_CLASSIC_DB(Vxx).scl
- Siemens CPU series S7-1200,1500 PN_DMA_2_DB.scl(Vxx).scl

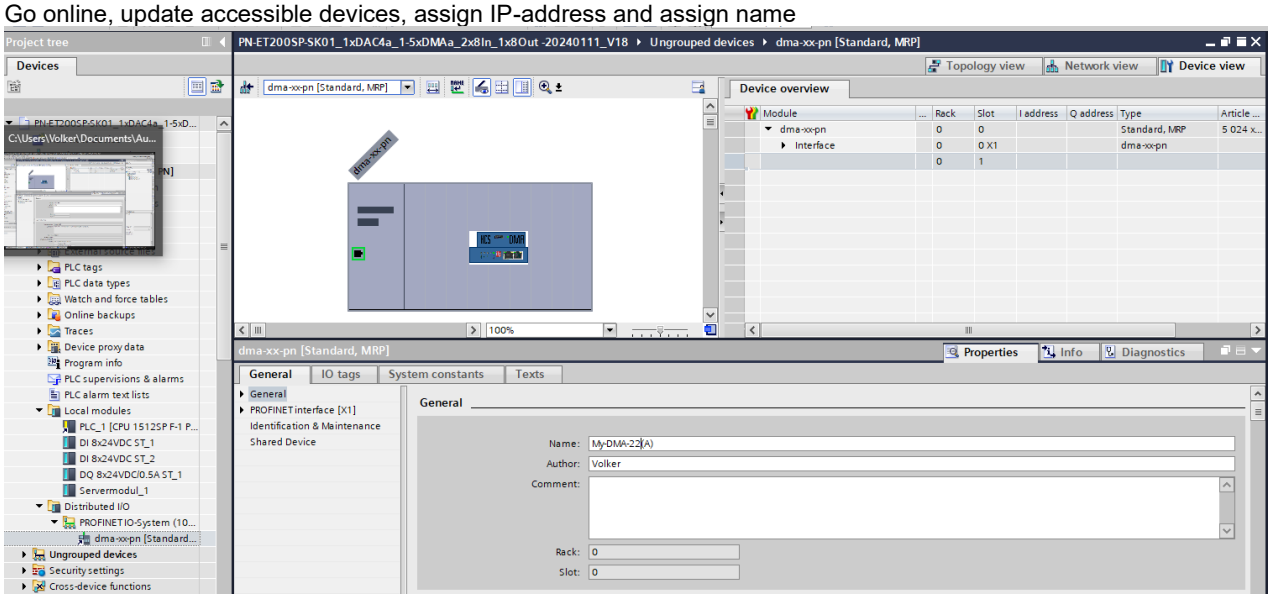
(If the DMA fold is mixed with DMA's, working in different mode's, only the bits for the control-byte and the status-word for the certain DMA module is useable).

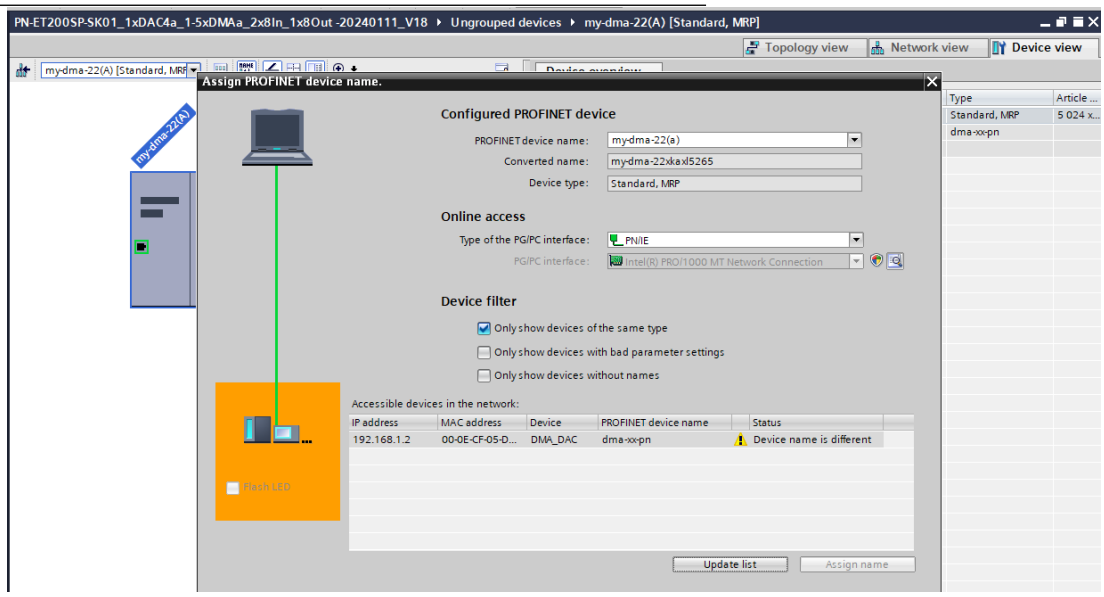
## 9.1 Setup the network configuration for the DMA module

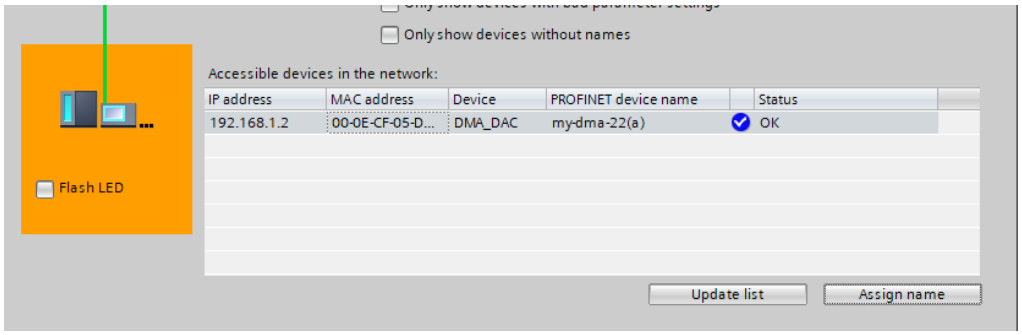
Download and install the GSD file



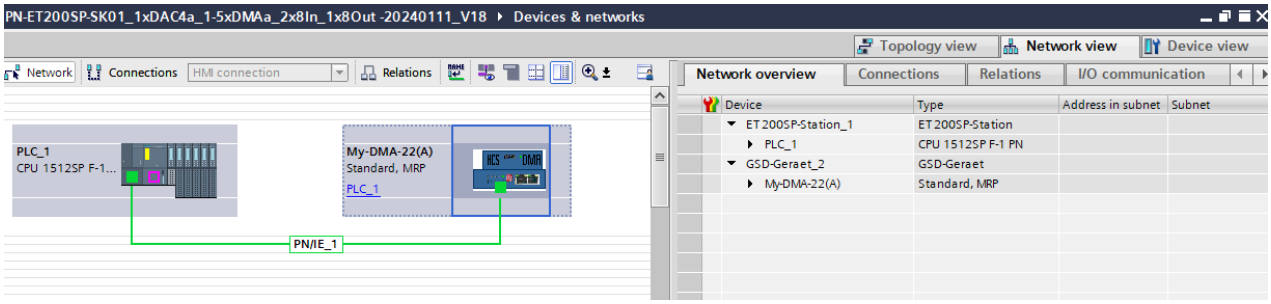

Go online, update accessible devices, assign IP-address and assign name



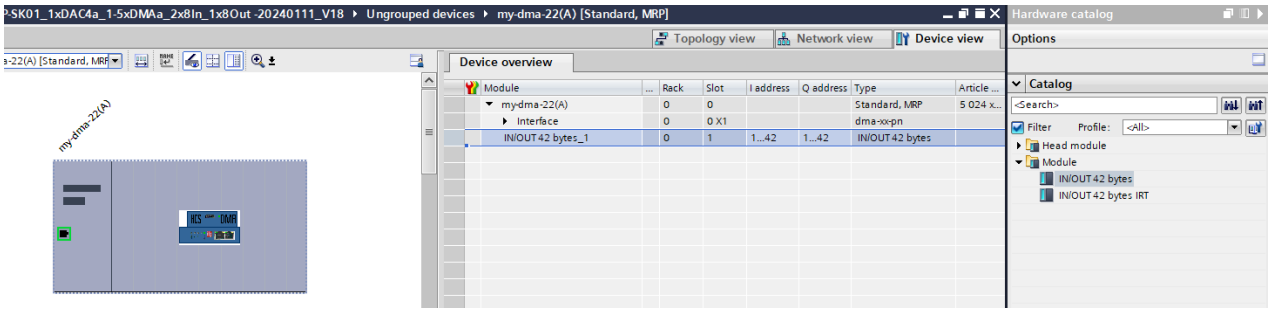



Add new “Distributed I/O” and connect The DMA to the network.



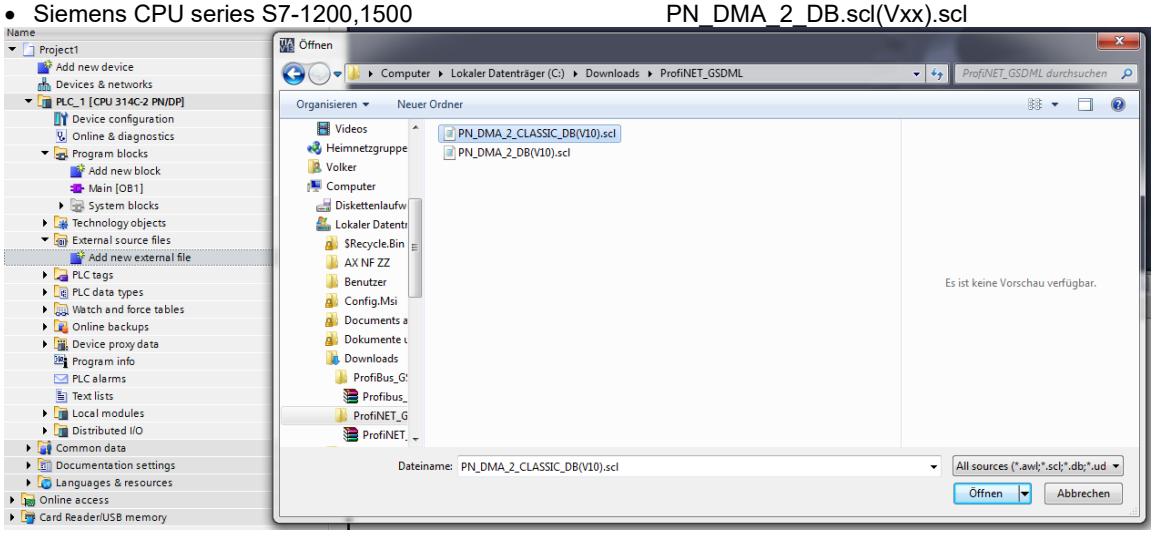
## 9.2 Assign the IO-Module from the GSD to the device



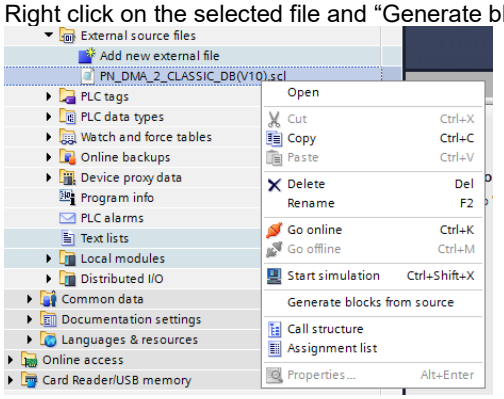

## 9.3 Install the function block

Add new external file, Chose the function block according your used CPU series

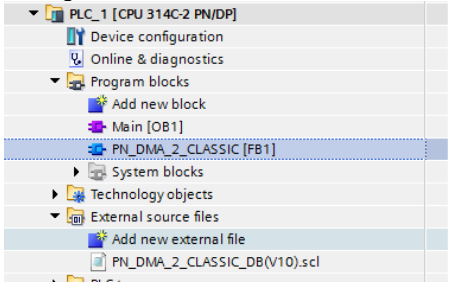
- Siemens CPU series S7-300,400 PN_DMA_2_CLASSIC_DB(Vxx).scl
- Siemens CPU series S7-1200,1500 PN_DMA_2_DB.scl(Vxx).scl




Right click on the selected file and “Generate blocks from the source”



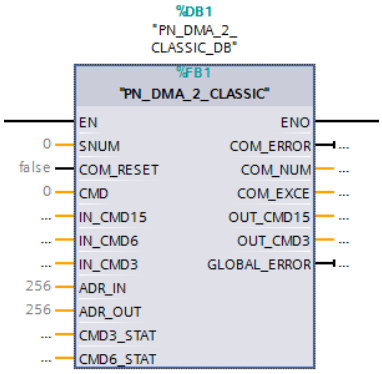

After generation the functionsblock is now available



## 9.4 Explanation of the function block

### 9.4.1 General

In this chapter we describe, how to use the function block in the different modes.
The structure and function for CMD 3, CMD 6 is the same in every working mode of the DMA-2. Only the CMD 15 differs depending on the DMA-2 operation mode.




The Bus slave address SNUM of every connected DMA-2 is defined by the parameter E 22.

| | |
|---|---|
| SNUM | Int |

The COM_ERROR is obtained from the telegram header, which is transmitted from the Profinet node to the PLC. When this error is pending, the slave address source and the cause can be found out by analyze the COM_EXCE code.
(See □ Chapter “6.1.3 CMD = 3, Error from Slave (4 bytes)“, page 35
or □ Chapter “6.3.3 CMD = 6, Error from Slave (4 bytes)”, page 37

Reset an communication error can by set the input #COM_RESET
From function block version V14 (PN_DMA_2a, version 1.4) on: if a module answers with an error (e.g. while the modules are still initialising after power-up), the telegram address TADR is changed automatically. A new telegram is thereby forced and the communication resumes automatically as soon as the cause has been removed. Short errors are filtered (20 function block calls). COM_ERROR remains latched for diagnostics until it is acknowledged once with #COM_RESET (TRUE, then FALSE).
Note: With function block versions up to V13, the communication may remain in error state after an error response, because the same telegram is sent again and the node only answers to changed telegrams (see chapter 5.2). #COM_RESET does not clear this state. Please update to V14.

| | |
|---|---|
| COM_RESET | Bool |

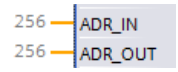
An GLOBAL_ERROR occurs if any communication error or/and internal DMA-2 error is pending.

| | |
|---|---|
| GLOBAL_ERROR | Bool |

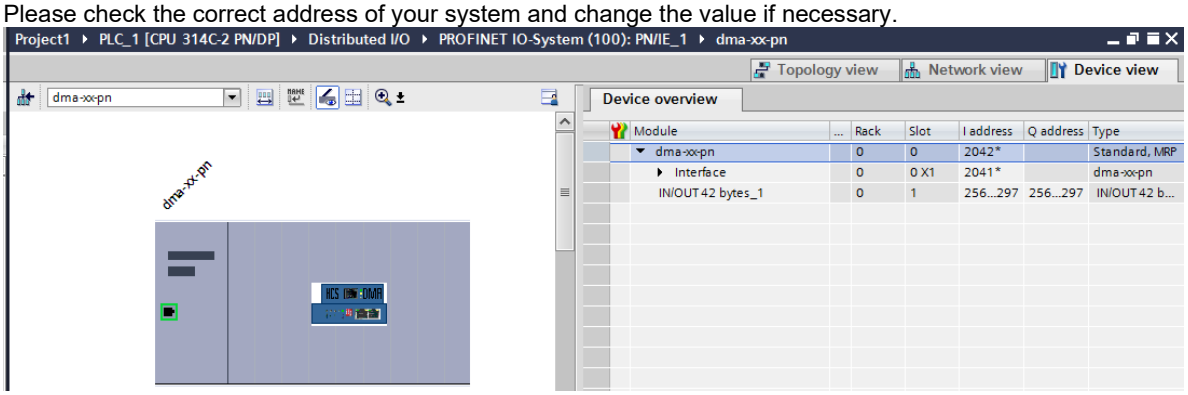
### 9.4.2 Classic CPU 300,400: Definition of the data buffer address

Unlike the CPU 1200, 1500, the E-address and A-address must be assigned to the variables #ADR_IN and #ADR_OUT here.

The data buffer address #ADR_IN, #ADR_OUT is predefined to 256.



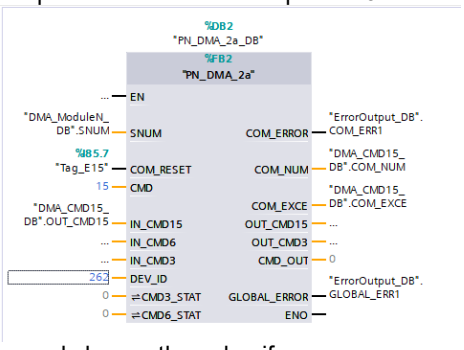

Please check the correct address of your system and change the value if necessary.



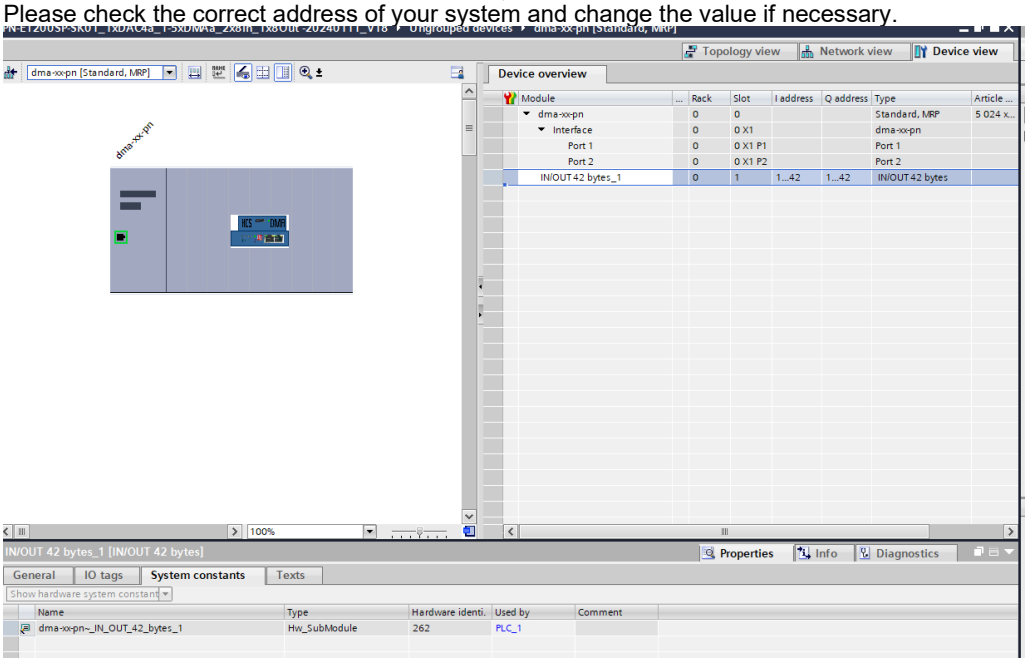

### 9.4.3 CPU 1200,1500: Specification of the data buffer address based on the hardware identifier

In contrast to CPU 300, 400, the hardware ID for the GSD I / O module must be assigned to the variable #DEV_ID.
The data buffer address #ADR_IN, #ADR_OUT is predefined in this example at 262.

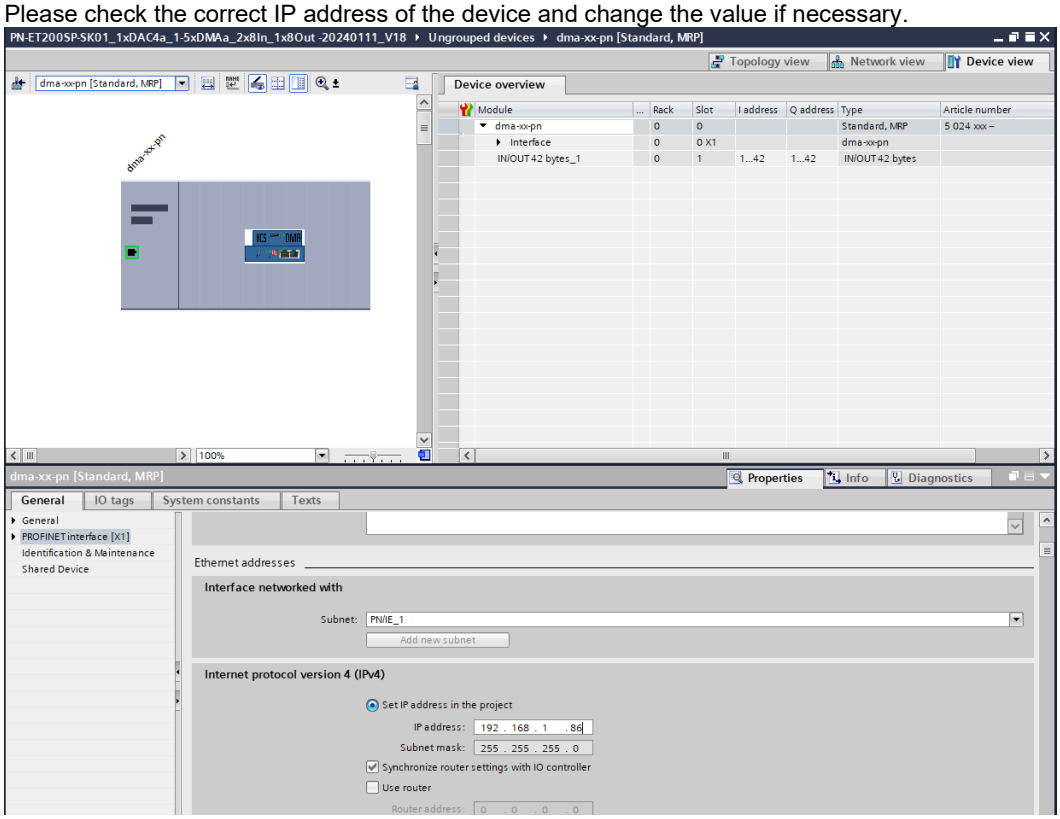



Please check the correct address of your system and change the value if necessary.



Please check the correct IP address of the device and change the value if necessary.




### 9.4.4 CMD 3, Reading values from one DMA-2 node

Every DMA-2 module connected to the Profinet node has an own SNUM (slave number). Together with this number it is possible to read an set of parameter of this specific DMA-2.
The parameter-ID's are shown in table □ Chapter "10 List of parameters", page 65

An reading access is done as followed:

- Define the slave number SNUM of the DMA-2 from which the parameters are to be read.

| SNUM | Int |
|---|---|

E.g. for slave 1 #SNUM := 1;.

- Setup the CMD: # CMD := 3;

| CMD | Int |
|---|---|

- Write the starting address of the first parameter ID which should be read by the structure

| IN_CMD3 | Struct |
|---|---|
| PAR_ID | Int |

E.g. parameter 22, S1.01 #IN_CMD3.PAR_ID := 22;

- Define the number of parameters to be read (maximum 8).

| IN_CMD3 | Struct |
|---|---|
| PAR_ID | Int |
| PAR_N | Int |

E.g. 4 parameters S1.01..S1.04 #IN_CMD3.PAR_N := 4;

- Definition of the states in #CMD3_STAT:

| | | | |
|---|---|---|---|
| CMD_UNSED | Int | | 0 |
| CMD_START | Int | | 1 |
| CMD_PROGRESS | Int | | 2 |
| CMD_FINISH | Int | | 4 |
| CMD_FAIL | Int | | 8 |

0: no action
1: Start of reading
2: Reading in process
4: Reading finished
8: Reading error occurred.

- Start the process, by writing #CMD3_STAT := 1; (CMD_START)

| | |
|---|---|
| CMD3_STAT | Int |

- When #CMD3_STAT reaches the value 4 (CMD_FINISH), Reading is finished, and the values can be read in the structure #OUT_CMD.VALUE[1..8]. In addition the number of received bytes will be shown I the structure #OUT_CMD.CNT. Every read parameter needs to 2 bytes.

| | |
|---|---|
| OUT_CMD3 | Struct |
| CNT | Int |
| VALUE | Array[1..8] of Int |
| VALUE[1] | Int |
| VALUE[2] | Int |
| VALUE[3] | Int |

- If an reading error occurred, the reason for the error can be analyzed by checking the #COM_xxxxx outputs of the function block.

| | |
|---|---|
| COM_ERROR | Bool |
| COM_NUM | Int |
| COM_EXCE | Int |

### 9.4.5 CMD 6, Writing Parameters to one DMA-2 node

**The written values are stored in a non-volatile memory.**
Every DMA-2 module connected to the Profinet node has an own SNUM (slave number). Together with this number it is possible to write an new parameter value to the parameter ID of this specific DMA-2.
The parameter-ID's are shown in table □ Chapter "10 List of parameters", page 65

An parameter writing access is done as followed:

- Define the slave number SNUM of the DMA-2 whose parameters are to be changed.

| | |
|---|---|
| SNUM | Int |

E.g. for slave 1 #SNUM := 1;.

- Setup the CMD: # CMD := 6;

| | |
|---|---|
| CMD | Int |

- Define the parameter ID and the value which should be changed.

| | |
|---|---|
| IN_CMD6 | Struct |
| PAR_ID | Int |
| PAR_VALUE | Int |

E.g. parameter 22, S1.01 =1000 (1.0 V) #IN_CMD3.PAR_ID := 22;
#IN_CMD3.PAR_VALUE := 1000;

- Definition of the states in #CMD6_STAT:

| | | | |
|---|---|---|---|
| CMD_UNSED | Int | | 0 |
| CMD_START | Int | | 1 |
| CMD_PROGRESS | Int | | 2 |
| CMD_FINISH | Int | | 4 |
| CMD_FAIL | Int | | 8 |

0: no action
1: Start of writing
2: Writing in process
4: Writing finished
8: Writing error occurred.

- Start the process, by writing #CMD6_STAT := 1; (CMD_START)

| | |
|---|---|
| CMD6_STAT | Int |

- When #CMD6_STAT reaches the value 4 (CMD_FINISH), writing is finished

- If an writing error occurred, the reason for the error can be analyzed by checking the #COM_xxxxx outputs of the function block.

| | |
|---|---|
| COM_ERROR | Bool |
| COM_NUM | Int |
| COM_EXCE | Int |

### 9.4.6 CMD 15, Writing and Reading once to all connected DMA nodes

The CMD 15 makes it possible to send and receive process data from all DMAs connected to the Profinet node. In this case, the variable SNUM (slave number) contains the maximum number of connected DMA-2s. (deviating from CMD 3 or CMAD6).
The meaning of the process values depending on the mode of the DMA-2 working at.
See □ Chapter “7.5 CMD = 15, structure of Profinet telegram definition, depending on SW Versions“, page 43

This also may be different to each module due to the DMA-2 fold composition. Also, the Control-Byte is depending on the working mode. Explanation of the bits
see □ Chapter “7.5.1.3 Explanation of special functions and module state bits“, page 44

An CMD 15 access is done as followed:

- Define the number of slaves SNUM of the modules connected to the Profinet node.

| | |
|---|---|
| SNUM | Int |

E.g. If using a 3-fold DMA-2 #SNUM := 3;.

- Setup the CMD: # CMD := 15;

| | |
|---|---|
| CMD | Int |

- Write the process data you want to send to the DMA-2 modules in the structure provided. The array index corresponds to the SADR (parameter E 22) of the connected modules

| | |
|---|---|
| IN_CMD15 | Array[1..5] of Struct |
| IN_CMD15[1] | Struct |
| IN_CMD15[2] | Struct |
| IN_CMD15[3] | Struct |
| IN_CMD15[4] | Struct |
| IN_CMD15[5] | Struct |

Module with SADR = 1 (parameter E22 = 1)

Module with SADR = 5 (parameter E22 = 5)

Structure of the transmit data set:

Standard SW

| | |
|---|---|
| IN_CMD15 | Array[1..5] o |
| IN_CMD15[1] | Struct |
| BUS_DISABLE | Bool |
| Din_1 | Bool |
| Din_2 | Bool |
| TST_CMP | Bool |
| CNTRL_1 | Bool |
| CNTRL_2 | Bool |
| CNTRL_3 | Bool |
| CNTRL_4 | Bool |
| VALUE1 | Int |
| VALUE2 | Int |

Special SW for Mode2.
The bits Din_1 and Din_2 are used for control the separate Enable

| | |
|---|---|
| IN_CMD15 | Array[1..5] of Struct |
| IN_CMD15[1] | Struct |
| BUS_DISABLE | Bool |
| Enable_1 | Bool |
| Enable_2 | Bool |
| TST_CMP | Bool |
| CNTRL_1 | Bool |
| CNTRL_2 | Bool |
| CNTRL_3 | Bool |
| CNTRL_4 | Bool |
| VALUE1 | Int |
| VALUE2 | Int |

Control Byte

Process values to DMA

- The internal telegram address is changed automatically so an return telegram is initiating every function block call. The array index corresponds to the SADR (parameter E 22) of the connected modules
Structure of the receive data set.

Standard SW

| | |
|---|---|
| OUT_CMD15 | Array[1..5] o... |
| OUT_CMD15[1] | Struct |
| HW_ENABLE | Bool |
| BUS_DISABLE | Bool |
| STAT_1 | Bool |
| STAT_2 | Bool |
| STAT_3 | Bool |
| STAT_4 | Bool |
| STAT_5 | Bool |
| ERROR | Bool |
| ERROR_NO | Byte |
| Din_1 | Bool |
| Dout_1 | Bool |
| DKOMP_1 | Bool |
| DKOMP_11 | Bool |
| Din_2 | Bool |
| Dout_2 | Bool |
| DKOMP_2 | Bool |
| DKOMP_22 | Bool |
| VALUE1 | Int |
| VALUE2 | Int |

Special SW for Mode2.
The bits Din_1 and Din_2 are used to display the separate Enable setting

| | |
|---|---|
| DMA_OUT_CMD15 | Array[1..5] of Struct |
| DMA_OUT_CMD1... | Struct |
| HW_ENABLE | Bool |
| BUS_DISABLE | Bool |
| STAT_1 | Bool |
| STAT_2 | Bool |
| STAT_3 | Bool |
| STAT_4 | Bool |
| STAT_5 | Bool |
| ERROR | Bool |
| ERROR_NO | Byte |
| Enable_1 | Bool |
| Dout_1 | Bool |
| DKOMP_1 | Bool |
| DKOMP_11 | Bool |
| Enable_2 | Bool |
| Dout_2 | Bool |
| DKOMP_2 | Bool |
| DKOMP_22 | Bool |
| VALUE1 | Int |
| VALUE2 | Int |

Status Word

Process values from DMA

- If an error has occurred, the cause of the error can be analyzed by evaluating the outputs #COM_xxxxx of the function block.

| | |
|---|---|
| COM_ERROR | Bool |
| COM_NUM | Int |
| COM_EXCE | Int |

# 10 List of parameters

Parameters in grey letters are not yet implemented in the DMA Software.
All values in decimal, for usage as data values they must be converted into Hex
R = read only parameter.
X = implemented, but not active (no access)
W/R = writing and reading possible.

| $ID_{Hex}$ | $ID_{dec}$ | W/R | Name | Function | Unit | Min | Max | Def | Description |
|---|---|---|---|---|---|---|---|---|---|
| 0x00 | 0 | R | Vers | Software version | --- | xx.xx | xx.xx | 10.01 | Depends on HW and SW version |
| 0x01 | 1 | R | d1.01 | Sum of analogue set value | V | -9999 | 9999 | - | -1000 == -1.000 V; |
| 0x02 | 2 | R | d1.02 | Sum of all post ramp set values | V | -9999 | 9999 | - | 1000 == 1.000 V |
| 0x03 | 3 | R | d1.03 | Set values after linearization | V | -9999 | 9999 | - | |
| 0x04 | 4 | R | d1.04 | Value after gain adjustment. | V | -9999 | 9999 | - | |
| 0x05 | 5 | R | d1.05 | Signal A | V | -9999 | 9999 | - | |
| 0x06 | 6 | R | d1.06 | Signal B | V | -9999 | 9999 | - | |
| 0x07 | 7 | R | d1.07 | Current A | A | 0 | 5000 | - | 1000 == 1.000 A |
| 0x08 | 8 | R | d1.08 | Current B | A | 0 | 5000 | - | |
| 0x09 | 9 | R | d1.09 | Total current | A | 0 | 5000 | - | |
| 0x0A | 10 | R | d1.10 | Desired value | V | -9999 | 9999 | - | -1000 == -1.000 V; 1000 == 1.000 V |
| 0x0B | 11 | R | d1.11 | Actual value, feedback value | V | -9999 | 9999 | - | |
| 0x0C | 12 | R | d1.12 | Lag error | V | -9999 | 9999 | - | |
| 0x0D | 13 | R | d1.13 | Controller output | V | -9999 | 9999 | - | |
| 0x0E | 14 | R | d2.01 | Sum of analogue set value | V | -9999 | 9999 | - | |
| 0x0F | 15 | R | d2.02 | Sum of all post ramp set values | V | -9999 | 9999 | - | |
| 0x10 | 16 | R | d2.03 | Set values after linearization | V | -9999 | 9999 | - | |
| 0x11 | 17 | R | d2.04 | Value after gain adjustment. | V | -9999 | 9999 | - | |
| 0x12 | 18 | R | d2.10 | Desired value | V | -9999 | 9999 | - | |
| 0x13 | 19 | R | d2.11 | Actual value, feedback value | V | -9999 | 9999 | - | |
| 0x14 | 20 | R | d2.12 | Lag error | V | -9999 | 9999 | - | |
| 0x15 | 21 | R | d2.13 | Controller output | V | -9999 | 9999 | - | |
| 0x16 | 22 | R/W | S1.01 | Set Value1 | V | -9999 | 9999 | 0 | |
| 0x17 | 23 | R/W | S1.02 | Set Value2 | V | -9999 | 9999 | 0 | |
| 0x18 | 24 | R/W | S1.03 | Set Value3 | V | -9999 | 9999 | 0 | |
| 0x19 | 25 | R/W | S1.04 | Set Value4 | V | -9999 | 9999 | 0 | |
| 0x1A | 26 | --- | S1.08 | reserved | | --- | --- | --- | --- |
| 0x1B | 27 | R/W | r1.01 | Ramp from 0 ⇒ - | s | 0 | 3950 | 0 | 1 == 10ms |
| 0x1C | 28 | R/W | r1.02 | Ramp from – ⇒ 0 | s | 0 | 3950 | 0 | 0 == Ramp function off |
| 0x1D | 29 | R/W | r1.03 | Ramp from 0 ⇒ + | s | 0 | 3950 | 0 | |
| 0x1E | 30 | R/W | r1.04 | Ramp from + ⇒ 0 | s | 0 | 3950 | 0 | |
| 0x1F | 31 | R/W | A1.01 | Profinet set value (Branch 1) | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V This value is, when written, always active. Cleared by reset, or writing “0”. For switching off of the analogue set-value, set E17 = “2” |
| 0x20 | 32 | R/W | A1.02 | Profinet feedback value (Branch 1) | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V C1.09 has to be switched off in order to avoid interferences |
| 0x21 | 33 | R/W | S2.01 | Set Value1 (Branch 2) | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V |
| 0x22 | 34 | R/W | S2.02 | Set Value2 (Branch 2) | V | -9999 | 9999 | 0 | |
| 0x23 | 35 | R/W | r2.01 | Ramp from 0 ⇒ - | s | 0 | 3950 | 0 | 1 == 10ms |
| 0x24 | 36 | R/W | r2.02 | Ramp from – ⇒ 0 | s | 0 | 3950 | 0 | 0 == Ramp function off |
| 0x25 | 37 | R/W | r2.03 | Ramp from 0 ⇒ + (Branch 2) | S | 0 | 3950 | 0 | |
| 0x26 | 38 | R/W | r2.04 | Ramp from + ⇒ 0 (Branch 2) | S | 0 | 3950 | 0 | |
| 0x27 | 39 | R/W | A2.01 | Profinet set value (Branch 2) | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V This value is, when written, always active. Cleared by reset, or writing “0”. For switching off of the analogue set-value, set E17 = “2” |
| 0x28 | 40 | R/W | A2.02 | Profinet feedback value (Branch 2) | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V C2.09 has to be switched off in order to avoid interferences |
| 0x29 | 41 | R/W | C1.00 | Controller selection | --- | 0 | 4 | 1 | 0 = off<br>1 = P-$PT_1$-I-$DT_1$<br>2 =Remote<br>3 =dff<br>4 =Remote + dff |
| 0x2A | 42 | R/W | C1.01 | Safety function | --- | 0 | 1 | 0 | 0 = off; 1 = on |
| 0x2B | 43 | R/W | C1.02 | Linearization | --- | 0 | 6 | 0 | 0 = linear<br>1 ... 5 = standard curves<br>6 = customized curve |

| $ID_{Hex}$ | $ID_{dec}$ | W/R | Name | Function | Unit | Min | Max | Def | Description |
|---|---|---|---|---|---|---|---|---|---|
| 0x2C | 44 | R/W | C1.03 | Gain A | V/V | 0 | 200 | 100 | 100 == Factor 1.00 |
| 0x2D | 45 | R/W | C1.04 | Gain B | V/V | 0 | 200 | 100 | 100 == Factor 1.00 |
| 0x2E | 46 | R/W | C1.05 | Set value sign and gain | V/V | -400 | 400 | 100 | 100 == Factor 1.00<br>Sign and gain! |
| 0x2F | 47 | R/W | C1.06 | Set value offset | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V |
| 0x30 | 48 | R/W | C1.07 | Dead band compensation A | V | 0 | 9999 | 0 | 1000 == 1.000 V |
| 0x31 | 49 | R/W | C1.08 | Dead band compensation B | V | 0 | 9999 | 0 | 9.999 V = max. current depending on solenoid selection |
| 0x32 | 50 | R/W | C1.09 | Sensor type<br><br>Attention:<br>No negative controller output possible when 10, 11 or 12 is selected! | --- | 1 | 28 | 4 | off = Deactivated<br>1 = 0...20mA<br>2 = 4...20mA<br>3 = 12mA +-8mA<br>4 = 0...10V<br>5 = 0...+-10V<br>6 = 6V +-2.5V<br>7 = 7.5V +-2.5V<br>8 = 6V +-4V<br>9 = reserved<br>10 = 0..20mA (positive contr. outp. only)<br>11 = 4..20mA (positive contr. outp. only)<br>12 = 0..10V (positive contr. outp. only)<br>13 = reserved<br>14 = 5V+-3.0V<br>15..20 = reserved<br>21 = Bipolar voltage semiautomatic calibration<br>22 = Unipolar voltage semiauto. calibration<br>23 = Unipolar volt. semiauto. calibration (positive contr. outp. only)<br>24 = Bipolar (internal) current semiauto. calibration<br>25 = Unipolar current semiauto. calibration<br>26 = Unipolar current semiauto. calibration (positive contr. outp. only)<br>27 = reserved<br>28 = 5V+-2.6V |
| 0x33 | 51 | R/W | C1.10 | Actual value gain | V/V | 0 | 400 | 100 | 100 == Factor 1.00 |
| 0x34 | 52 | R/W | C1.11 | Actual value offset | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V |
| 0x35 | 53 | R/W | C1.12 | Actual value sign | --- | - 1 | + 1 | 1 | - 1 = negative<br>0 = off<br>+ 1 = positive |
| 0x36 | 54 | R/W | C1.13 | P-Portion $K_{P1}$ | V/V | 0 | 400 | 0 | 100 == Factor 1.00 |
| 0x37 | 55 | R/W | C1.14 | T-Portion for PT1 (to C1.16) | S | 0 | 1000 | 0 | 100 == 1.00 |
| 0x38 | 56 | R/W | C1.15 | Threshold (C1.13, C1.16) | V | 0 | 9999 | 9999 | 1000 == 1.000 V |
| 0x39 | 57 | R/W | C1.16 | P-Portion $K_{P2}$ | V/V | 0 | 400 | 0 | 100 == Factor 1.00 |
| 0x3A | 58 | R/W | C1.17 | I-Portion | V/s | 0 | 4000 | 0 | 1000 == 1.000 |
| 0x3B | 59 | R/W | C1.18 | D-Portion | Vs | 0 | 400 | 0 | 100 == 1.00 |
| 0x3C | 60 | R/W | C1.19 | T-Portion for DT1 | S | 0 | 1000 | 0 | 100 == 1.00 |
| 0x3D | 61 | R/W | C1.20 | Gain ( C1.13 and C1.16) | V/V | 1 | 32 | 1 | 2 = Factor 2.00 |
| 0x3E | 62 | R/W | C1.21 | Comparator upper level | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V |
| 0x3F | 63 | R/W | C1.22 | Comparator lower level | V | -9999 | 9999 | 0 | |
| 0x40 | 64 | R/W | C1.23 | Comparator delay into window | S | 0 | 9999 | 0 | 1 == 10 ms |
| 0x41 | 65 | R/W | C1.24 | Comparator delay out of window | S | 0 | 9999 | 0 | 0 == no delay |
| 0x42 | 66 | R/W | C1.25 | Comparator selection COMP_1 | --- | 0 | 3 | 0 | 0 = off<br>1 = Set value<br>2 = Actual value<br>3 =Lag error |
| 0x43 | 67 | R/W | C1.26 | Cable fracture detection feedback | --- | 0 | 1 | 0 | 0 = off; 1 = active |
| 0x44 | 68 | R/W | C2.00 | Controller selection | --- | 0 | 4 | 0 | 0 = off<br>1 = P-$PT_1$-I-$DT_1$<br>2 =Remote<br>3 =dff<br>4 =Remote + dff |
| 0x45 | 69 | R/W | C2.01 | Safety function | --- | 0 | 1 | 0 | 0 = off; 1 = on |
| 0x46 | 70 | R/W | C2.02 | Linearization | --- | 0 | 6 | 0 | 0 = linear<br>1 ... 5 = standard curves<br>6 = customized curve |
| 0x47 | 71 | R/W | C2.03 | Gain A | V/V | 0 | 200 | 100 | 100 == Factor 1.00 |
| 0x48 | 72 | R/W | C2.04 | Gain B | V/V | 0 | 200 | 100 | 100 == Factor 1.00 |
| 0x49 | 73 | R/W | C2.05 | Set value sign and gain | V/V | -400 | 400 | 100 | 100 == Factor 1.00<br>Sign and gain! |

| $ID_{Hex}$ | $ID_{dec}$ | W/R | Name | Function | Unit | Min | Max | Def | Description |
|---|---|---|---|---|---|---|---|---|---|
| 0x4A | 74 | R/W | C2.06 | Set value offset | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V |
| 0x4B | 75 | R/W | C2.07 | Dead band compensation A | V | 0 | 9999 | 0 | 1000 == 1.000 V |
| 0x4C | 76 | R/W | C2.08 | Dead band compensation B | V | 0 | 9999 | 0 | 9.999 V = max. current depending on solenoid selection |
| 0x4D | 77 | R/W | C2.09 | Sensor type<br><br>Attention:<br>No negative controller output possible when 10, 11 or 12 is selected! | --- | 0 | 12 | 4 | off = Deactivated<br>1 = 0...20mA<br>2 = 4...20mA<br>3 = 12mA +-8mA<br>4 = 0...10V<br>5 = 0...+-10V<br>6 = 6V +-2.5V<br>7 = 7.5V +-2.5V<br>8 = 6V +-4V<br>9 = reserved<br>10 = 0..20mA (positive contr. outp. only)<br>11 = 4..20mA (positive contr. outp. only)<br>12 = 0..10V (positive contr. outp. only)<br>13 = reserved<br>14 = 5V+-3.0V<br>15..27 = reserved<br>28 = 5V+-2.6V |
| 0x4E | 78 | R/W | C2.10 | Actual value gain | V/V | 0 | 400 | 100 | 100 == Factor 1.00 |
| 0x4F | 79 | R/W | C2.11 | Actual value offset | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V |
| 0x50 | 80 | R/W | C2.12 | Actual value sign | --- | - 1 | + 1 | 1 | - 1 = negative<br>0 = off<br>+ 1 = positive |
| 0x51 | 81 | R/W | C2.13 | P-Portion $K_{P1}$ | V/V | 0 | 400 | 0 | 100 == Factor 1.00 |
| 0x52 | 82 | R/W | C2.14 | T-Portion for PT1 (to C1.16) | S | 0 | 1000 | 0 | 100 == 1.00 |
| 0x53 | 83 | R/W | C2.15 | Threshold (C1.13, C1.16) | V | 0 | 9999 | 9999 | 1000 == 1.000 V |
| 0x54 | 84 | R/W | C2.16 | P-Portion $K_{P2}$ | V/V | 0 | 400 | 0 | 100 == Factor 1.00 |
| 0x55 | 85 | R/W | C2.17 | I-Portion | V/s | 0 | 4000 | 0 | 1000 == 1.000 |
| 0x56 | 86 | R/W | C2.18 | D-Portion | Vs | 0 | 400 | 0 | 100 == 1.00 |
| 0x57 | 87 | R/W | C2.19 | T-Portion for DT1 | S | 0 | 1000 | 0 | 100 == 1.00 |
| 0x58 | 88 | R/W | C2.20 | Gain (C1.13 and C1.16) | V/V | 1 | 32 | 1 | 2 = Factor 2.00 |
| 0x59 | 89 | R/W | C2.21 | Comparator upper level | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V |
| 0x5A | 90 | R/W | C2.22 | Comparator lower level | V | -9999 | 9999 | 0 | |
| 0x5B | 91 | R/W | C2.23 | Comparator delay into window | S | 0 | 9999 | 0 | 1 == 10 ms |
| 0x5C | 92 | R/W | C2.24 | Comparator delay out of window | S | 0 | 9999 | 0 | 0 == no delay |
| 0x5D | 93 | R/W | C2.25 | Comparator selection COMP_2 | --- | 0 | 3 | 0 | 0 = off<br>1 = Set value<br>2 = Actual value<br>3 =Lag error |
| 0x5E | 94 | R/W | C2.26 | Cable fracture detection feedback | --- | 0 | 1 | 0 | 0 = off; 1 = active |
| 0x5F | 95 | R | E00 | Operation mode<br>(depends on HW + SW version) | --- | 1 | 11 | 3 | 1 = Open loop one valve<br>2 = Open loop two valves<br>3 = Closed loop one valve<br>4 = Closed loop on application<br>6 = Closed loop valve/application<br>8 = Closed loop application/ application<br>10 = Controller function only, one feedback<br>11 = Controller function only, two feedback |
| 0x60 | 96 | R/W | E01 | Analogue output<br>(depends on HW + SW version) | --- | 1<br><br>and<br>14 | 13<br><br>and<br>21 | 1 | 1 = d1.01 to<br>13 = d1.13 and<br>14 = d2.01 to<br>21 = d2.13 |
| 0x61 | 97 | R/W | E02 | Push-Pull function<br>(depends on HW + SW version) | --- | 0 | 1 | 0 | 0 = off v(no push pull)<br>1 = active<br>2 = common “+” for solenoids<br>(Remark: only for max 0,8 A current )<br>3 = full bridge<br>4 = off + detection low current to solenoid<br>5 = active + detection low current to solenoid |
| 0x62 | 98 | R/W | E03<br>E1.03 | Solenoid selection<br>Solenoid selection A<br>(depends on HW + SW version) | --- | 1 | 7 | 6 | Standard solenoid definition<br>1 = 0,8 A<br>2 = 1,1 A<br>3 = 1,3 A<br>4 = 1,6 A<br>5 = 2,4 A<br>6 = 2,7 A<br>7 = 3,5 A<br>(depends on HW + SW version) |

| $ID_{Hex}$ | $ID_{dec}$ | W/R | Name | Function | Unit | Min | Max | Def | Description |
|---|---|---|---|---|---|---|---|---|---|
| 0x63 | 99 | R | E04<br>E1.04 | P-Portion current contr. Energization A | --- | 500 | 9999 | 500 | Default for 2,700 A solenoid |
| 0x64 | 100 | R | E05<br>E1.05 | I-Portion current contr. Energization A | --- | 500 | 9999 | 500 | |
| 0x65 | 101 | R | E06<br>E1.06 | P-Portion cur. contr. de- energization A | --- | 1300 | 9999 | 1300 | |
| 0x66 | 102 | R | E076<br>E1.07 | I-Portion cur. contr. De- energization A | --- | 500 | 9999 | 500 | |
| 0x67 | 103 | R/W | E08 | Ramp selection | --- | 1 | 2 | 1 | 0 = digital set v. (time constant)<br>1 = all set v. (rise constant.)<br>2 = selectable ramps |
| 0x68 | 104 | R/W | E09 | Time delay enable signal | s | 0 | 9999 | 0 | 1000 = 1.00s |
| 0x69 | 105 | R/W | E10<br>E1.10 | Solenoid current adaptation A | --- | 50 | 110 | 100 | Variable adjustment of max. current<br>100 == Factor 1.00 |
| 0x6A | 106 | R/W | E11 | Initial current solenoid A | V | 0 | 9999 | 0 | 3.000 V = 30 % of max. rated current |
| 0x6B | 107 | R/W | E12 | Initial current solenoid B | V | 0 | 9999 | 0 | |
| 0x6C | 108 | R/W | E13<br>E1.13 | Dither Amplitude A | V | 0 | 3000 | 0 | |
| 0x6D | 109 | R/W | E14<br>E1.14 | Dither Frequency A | Hz | 1 | 300 | 0 | --- |
| 0x6E | 110 | R/W | E15 | Selection set point S1.06 (U/I) | --- | 0 | 11 | 1 | off = Deactivated<br>1 = Voltage -10V..+10V<br>2 = Voltage 0..10V<br>3 = Voltage 5V+-5V<br>4 = Voltage 5V+-4.5V w/o signal observation<br>5 = Voltage 5V+-4.5V with signal observation<br>6 = 0..20mA w/o cable fract. det.<br>7 = 10+-10mA w/o cable fract. det.<br>8 = 4..20mA w/o cable fract. det.<br>9 = 4..20mA with cable fract. det.<br>10 = 12+-8mA w/o cable fract. det.<br>11 = 12+-8mA with cable fract. det. |
| 0x6F | 111 | R/W | E16 | Reserved | --- | --- | --- | --- | --- |
| 0x70 | 112 | R/W | E17 | Set value activation mode | --- | 0 | 2 | 0 | 0 = digital and analog set value input active<br>1 = reserved<br>2 = only digital set value input active |
| 0x71 | 113 | R/W | E18 | Error/Comparator Output | --- | 0 | 9 | 0 | 0 = Error<br>1 = Comp_1 positive logic<br>2 = Comp_1 negative logic<br>3 = Comp_2 positive logic<br>4 = Comp_2 negative logic<br>5 = Dout_1 positive logic<br>6 = Dout_1 negative logic<br>7 = Dout_2 positive logic<br>8 = Dout_2 negative logic<br>9 = Output is controlled by bus bit: CNTRL_4 (CMD15 Control-Byte 0x40), status in STAT_5 (SW version V4x.02* or higher) |
| 0x72 | 114 | R/W | E19 | Factor for analog output<br>(depends on HW + SW version) | V/V | -400 | 400 | 100 | 100 == Factor 1.00<br>Sign and gain! |
| 0x73 | 115 | X | E20 | Reserved | --- | --- | --- | --- | |
| .. | .. | | | | | | | | |
| 0x9B | 155 | R/W | E2.03 | Solenoid selection<br>Solenoid selection B<br>(depends on HW + SW version) | --- | | | | Standard solenoid definition<br>1 = 0,8 A<br>2 = 1,1 A<br>3 = 1,3 A<br>4 = 1,6 A<br>5 = 2,4 A<br>6 = 2,7 A<br>7 = 3,5 A<br>(depends on HW + SW version) |
| 0x9C | 156 | R | E2.04 | P-Portion current contr. Energization B | --- | 0 | 9999 | 500 | Default for 2,700 A solenoid |
| 0x9D | 157 | R | E2.05 | I-Portion current contr. Energization B | --- | 0 | 9999 | 500 | |
| 0x9E | 158 | R | E2.06 | P-Portion cur. contr. De-energization B | --- | 0 | 9999 | 1300 | |
| 0x9F | 159 | R | E2.07 | I-Portion cur. contr. De-energization B | --- | 0 | 9999 | 500 | |
| 0xA0 | 160 | R/W | E2.10 | Solenoid current adaption B | --- | 50 | 110 | 100 | Variable adjustment of max. current<br>100 == Factor 1.00 |
| 0xA1 | 161 | R/W | E2.13 | Dither Amplitude B | V | 0 | 300 | 0 | |
| 0xA2 | 162 | R/W | E2.14 | Dither Frequency B | Hz | 0 | 300 | 0 | |
| 0xA3 | 163 | R/W | E 22 | Profinet Slave Address | --- | 0 | 32 | 0 | 0 = No bus function<br>1 = Standard Profinet |

| $ID_{Hex}$ | $ID_{dec}$ | W/R | Name | Function | Unit | Min | Max | Def | Description |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | 2..5 = Multiple Slave Profinet |
| 0xA4 | 164 | R/W | C1.27 | Hysteresis command A | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V |
| 0xA5 | 165 | R/W | C2.27 | Hysteresis command B | 1 | -9999 | 9999 | 0 | |
| 0xA6 | 166 | R/W | C1.28 | Comparator upper level | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V |
| 0xA7 | 167 | R/W | C1.29 | Comparator lower level | V | -9999 | 9999 | 0 | |
| 0xA8 | 168 | R/W | C1.30 | Comparator delay into window | S | 0 | 9999 | 0 | 1 == 10 ms |
| 0xA9 | 169 | R/W | C1.31 | Comparator delay out of window | S | 0 | 9999 | 0 | 0 == no delay |
| 0xAA | 170 | R/W | C1.32 | Comparator selection Comp_11 | --- | 0 | 3 | 0 | 0 = off<br>1 = Set value<br>2 = Actual value<br>3 =Lag error |
| 0xAB | 171 | R/W | C2.28 | Comparator upper level | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V |
| 0xAC | 172 | R/W | C2.29 | Comparator lower level | V | -9999 | 9999 | 0 | |
| 0xAD | 173 | R/W | C2.30 | Comparator delay into window | S | 0 | 9999 | 0 | 1 == 10 ms |
| 0xAE | 174 | R/W | C2.31 | Comparator delay out of window | S | 0 | 9999 | 0 | 0 == no delay |
| 0xAF | 175 | R/W | C2.32 | Comparator selection Comp_22 | --- | 0 | 3 | 0 | 0 = off<br>1 = Set value<br>2 = Actual value<br>3 =Lag error |
| 0xB0 | 176 | R/W | E Int | Internal Digital Switches (Read/Set) | --- | 0 | 0xFFFF | | 0x0001 = Din_1 active (Read/Write)<br>0x0002 = Dout_1 active (Read only )<br>0x0004 = Comp_1 active (Read only )<br>0x0008 = Comp_11 active (Read only )<br>0x0010 = Din_2 active (Read/Write)<br>0x0020 = Dout_2 active (Read only )<br>0x0040 = Comp_2 active (Read only )<br>0x0080 = Comp_22 active (Read only )<br>0x0100 = Input SP1 active (Read only )<br>0x0200 = Input SP2 active (Read only )<br>0x0400 = Enable active (Read only )<br>0x0800 = Error active (Read only )<br>0x1000 = Input SP3 active (Read only )<br>0x2000 = Input SP4 active (Read only )<br>0x8000 = Bus Disable Card (Read/Write) |
| 0xB1 | 177 | R/W | E 23 | Profinet telegram timeout window | S | 0 | 9999 | 0 | 0 == function deactivated<br>1 == 10 ms |
| 0xB2 | 178 | R/W | E 26 | Profinet Timeout Error handling | --- | 0 | 1 | 0 | 0 = Blinking Enable Led shows Profinet timeout<br>1 = Profinet timeout causes error. |
| 0xB3 | 179 | R/W | C1.33 | I-portion limitation | V | 0 | 9999 | | 0 == 0.000 V; 9999 == 9.999 V |
| 0xB4 | 180 | R/W | C2.33 | I-portion limitation | V | 0 | 9999 | | 0 == 0.000 V; 9999 == 9.999 V |
| .. | .. | | | | | | | | |
| 0xD4 | 212 | R/W | E 33 | Offset for analog output (depends on HW + SW version) | V | -9999 | 9999 | 0 | -1000 == -1.000 V; 1000 == 1.000 V |
| .. | .. | | | | | | | | |
| 0xDA | 218 | R/W | C1.36 | Sensor signal correction factor for signal < 0. branch 1 | V/V | -100 | 100 | 100 | 100 == Factor 1.00 |
| 0xDB | 219 | R/W | C2.36 | Sensor signal correction factor for signal < 0. branch 2 | V/V | -100 | 100 | 100 | 100 == Factor 1.00 |
| 0xDC | 220 | R/W | C1.37 | Spool overlap compensation A | V | 0 | 9999 | 0 | 1000 == 1.000 V |
| 0xDD | 221 | R/W | C1.38 | Spool overlap compensation B | V | 0 | 9999 | 0 | 9.999 V = max. current depending on solenoid selection |
| 0xDE | 222 | R/W | C2.37 | Spool overlap compensation A | V | 0 | 9999 | 0 | 1000 == 1.000 V |
| 0xDF | 223 | R/W | C2.38 | Spool overlap compensation B | V | 0 | 9999 | 0 | 9.999 V = max. current depending on solenoid selection |
| .. | .. | | | | | | | | |
| 0xE5 | 229 | R/W | C1.41 | Hysteresis threshold for resetting the I regulation value (the function is deactivated at 0.00) | V | 0 | 9999 | 0 | 1000 == 1.000 V |
| 0xE6 | 230 | R/W | L1.x0 | Linearization curve branch 1 [0,0] | V | 0 | 0 | 0 | 1000 == 1.000 V |
| 0xE7 | 232 | R/W | L1.y0 | Linearization curve [0,0] | V | 0 | 9999 | 0 | |
| 0xE8 | 232 | R/W | L1.x1 | Linearization curve [1,1] | V | 0 | 9999 | 1250 | |
| 0xE9 | 233 | R/W | L1.y1 | Linearization curve [1,1] | V | 0 | 9999 | 1250 | |
| 0xEA | 234 | R/W | L1.x2 | Linearization curve [2,2] | V | 0 | 9999 | 2500 | |
| 0xEB | 235 | R/W | L1.y2 | Linearization curve [2,2] | V | 0 | 9999 | 2500 | |
| 0xEC | 236 | R/W | L1.x3 | Linearization curve [3,3] | V | 0 | 9999 | 3750 | |
| 0xED | 237 | R/W | L1.y3 | Linearization curve [3,3] | V | 0 | 9999 | 3750 | |
| 0xEE | 238 | R/W | L1.x4 | Linearization curve [4,4] | V | 0 | 9999 | 5000 | |
| 0xEF | 239 | R/W | L1.y4 | Linearization curve [4,4] | V | 0 | 9999 | 5000 | |
| 0xF0 | 240 | R/W | L1.x5 | Linearization curve [5,5] | V | 0 | 9999 | 6250 | |
| 0xF1 | 241 | R/W | L1.y5 | Linearization curve [5,5] | V | 0 | 9999 | 6250 | |

| $ID_{Hex}$ | $ID_{dec}$ | W/R | Name | Function | Unit | Min | Max | Def | Description |
|---|---|---|---|---|---|---|---|---|---|
| 0xF2 | 242 | R/W | L1.x6 | Linearization curve [6,6] | V | 0 | 9999 | 7500 | |
| 0xF3 | 243 | R/W | L1.y6 | Linearization curve [6,6] | V | 0 | 9999 | 7500 | |
| 0xF4 | 244 | R/W | L1.x7 | Linearization curve [7,7] | V | 0 | 9999 | 8750 | |
| 0xF5 | 245 | R/W | L1.y7 | Linearization curve [7,7] | V | 0 | 9999 | 8750 | |
| 0xF6 | 246 | R/W | L1.x8 | Linearization curve [8,8] | V | 0 | 9999 | 9999 | |
| 0xF7 | 247 | R/W | L1.y8 | Linearization curve [8,8] | V | 9999 | 9999 | 9999 | |
| 0xF8 | 248 | R/W | L2.x0 | Linearization curve branch 2 [0,0] | V | 0 | 0 | 0 | 1000 == 1.000 V |
| 0xF9 | 249 | R/W | L2.y0 | Linearization curve [0,0] | V | 0 | 9999 | 0 | |
| 0xFA | 250 | R/W | L2.x1 | Linearization curve [1,1] | V | 0 | 9999 | 1250 | |
| 0xFB | 251 | R/W | L2.y1 | Linearization curve [1,1] | V | 0 | 9999 | 1250 | |
| 0xFC | 252 | R/W | L2.x2 | Linearization curve [2,2] | V | 0 | 9999 | 2500 | |
| 0xFD | 253 | R/W | L2.y2 | Linearization curve [2,2] | V | 0 | 9999 | 2500 | |
| 0xFE | 254 | R/W | L2.x3 | Linearization curve [3,3] | V | 0 | 9999 | 3750 | |
| 0xFF | 255 | R/W | L2.y3 | Linearization curve [3,3] | V | 0 | 9999 | 3750 | |
| 0x100 | 256 | R/W | L2.x4 | Linearization curve [4,4] | V | 0 | 9999 | 5000 | |
| 0x101 | 257 | R/W | L2.y4 | Linearization curve [4,4] | V | 0 | 9999 | 5000 | |
| 0x102 | 258 | R/W | L2.x5 | Linearization curve [5,5] | V | 0 | 9999 | 6250 | |
| 0x103 | 259 | R/W | L2.y5 | Linearization curve [5,5] | V | 0 | 9999 | 6250 | |
| 0x104 | 260 | R/W | L2.x6 | Linearization curve [6,6] | V | 0 | 9999 | 7500 | |
| 0x105 | 261 | R/W | L2.y6 | Linearization curve [6,6] | V | 0 | 9999 | 7500 | |
| 0x106 | 262 | R/W | L2.x7 | Linearization curve [7,7] | V | 0 | 9999 | 8750 | |
| 0x107 | 263 | R/W | L2.y7 | Linearization curve [7,7] | V | 0 | 9999 | 8750 | |
| 0x108 | 264 | R/W | L2.x8 | Linearization curve [8,8] | V | 0 | 9999 | 9999 | |
| 0x109 | 265 | R/W | L2.y8 | Linearization curve [8,8] | V | 9999 | 9999 | 9999 | |
| .. | .. | | | | | | | | |

Table 9: Parameter list

# 11 Troubleshooting

## 11.1 General

The status LED in the Front of the DMA don't lit:  or 
=> Are all modules supplied with power? (Supply 24V / Supply 0V)
To be sure the power is applied, disconnect the USB interface!

The status LED at the slave module don't lit yellow: 
=> Is the Enable input powered with 24 V?
=> Is the Software enable switched off? (See the special byte in CMD-15-mode)

The current of the solenoid did not reach the desired value:
=> Could the power supply provide enough power?
=> Is the selected type of solenoid type correct?

The status LED of one of the DMA slave modules show red color: 
=> The slave module is in error state.
=> To acknowledge this error, the enable signal of the module must be switched off/on - Then the error should disappear.

## 11.2 No Profinet communication is possible

**THE RED LED "BUS ERROR" IS LIT:**

=> Check the wiring of the Profinet (termination, length of cables, connectors)

The "universal module" is used in PLC-project:
=> This is not allowed, only the module listed in GDS-file are supported. (The universal module is automatically displayed in the S7 configurator, please not chose this one)

## 11.3 Communication ok, but current outputs are inactive

Could the set point/feedback-value be monitored by the DMA module? (parameter d1.11, d1.01, d1.02):

Yes => Please check the setting of the Ex.xx, Cx.xx parameters.
Check also the “Bus slave address” E22 of each module.

No => Apply a manual signal to the analogue inputs, please be sure that the parameter E17 is set to “off”.
Otherwise analogue inputs would be ignored.

Are the set point/feedback-value now visible?

Yes => The PLC don’t send a correct telegram. Please check you PLC program, and modify the PLC commands in your program.

No => Check general errors, see □ Chapter “11.1 General”, page 70.

## 11.4 PNO Certificate

PROFIBUS • PROFINET

# Certificate

PROFIBUS Nutzerorganisation e.V. grants to

**HCS Hydraulic Control Systems GmbH**
**Neuffener Str. 29, 72636 Frickenhausen, Germany**

the Certificate No: **Z01871** for the PROFIBUS device:

| | |
|---|---|
| **Model Name:** | **DMA-22-VPC3** |
| **Revision:** | **V2.4; SW/FW: V0103a; HW: revision 02** |
| **GSD:** | **HCS0EA7.GSD, File Version: V2.4 17.01.2014** |

This certificate confirms that the product has successfully passed the certification tests with the following scope:

| | |
|---|---|
| ☑ DP-V0 | MS0, Sync, Freeze, Fail_Safe, Set_Slave_Add |
| ☑ DP-V1 | MS1, MS2, I&M |
| ☑ Physical Layer | RS485 |

Test Report Number: **FZI0001**
Authorized Test Laboratory: **FZI Forschungszentrum Informatik, Karlsruhe, Germany**

The tests were executed in accordance with the following documents:
"Test Specifications for PROFIBUS DP Slaves, Version 3.09, September 2008".
This certificate is granted according to the document:
"Framework for testing and certification of PROFIBUS and PROFINET products".
For all products that are placed in circulation by July 14, 2017 the certificate is valid for life.

(Official in Charge)

Board of PROFIBUS Nutzerorganisation e. V.

(Karsten Schneider)

(K.-P. Lindner)

## 12 Declaration of Conformity

# EC Declaration of Conformity in accordance with EMC Directive 2014/30/EU

**HCS Hydraulic Control Systems GmbH**
**Neuffener Str. 29**
**D-72636 Frickenhausen**

Hereby declares that the product described as follows complies in terms of its design, as well as in the version placed in the stream of commerce by us, with the relevant requirements of the directive. This declaration is void in the event of any change to the product without our written agreement.

| | |
|---|---|
| Product: | Digital Amplifier and Controller Module |
| Intended use | Automation systems (industrial applications) |
| Model: | **DMA-22(A)-x** |
| Rated voltage: | 24 V DC; SELV, PELV |
| Rated power: | max. 100 VA |
| Protection class: | III |
| Protection degree: | IP00 (IP20 on request) |
| Relevant EU Directive: | EMC Directive 2014/30/EU |
| Applicable EU Standards: | |
| Emissions: | EN 61000-6-4:2007 + A1:2011;<br>EN 61000-6-3:2007 + A1:2011 |
| Immunity: | EN 61000-6-2: 2019 |

Date/manufacturers signature



15.01.2024
Details of signatory: Dipl.-Ing. (FH) Peter Deuschle (General Manager)

## *13 Notes:*

## *14* HCS distributors and partners

*Please refer to:*

**https://www.h-c-s-gmbh.de/en/sales-partners**

on our web side.

- End -