

DMA Profibus Anleitung

Revisionen

Revision	Datum	Beschreibung
Vorläufig	10.08.2006	Erster Ausdruck
Vorläufig	14.08.2006	Slave Adresse SDR, Bytecount CNT hinzugefügt Neue Write, Read, Antwort Telegram Definition Read, Write Befehl geändert (nach Modbus Standard)
Vorläufig	27.08.2006	Master Read geändert von 6 Bytes auf 7 Bytes N geändert in N-High und N-low (nach Modbus Standard) Fehler #5 timeout (z.B.: DMA Protokol timeout) hinzugefügt
Vorläufig	28.08.2006	Parameter Beschreibung übersetzt für englische Version und auf neuen Stand gebracht, DMA SW Version hinzugefügt
Vorläufig	29.08.2006	Parameter Liste auf neuen Stand gebracht
R_1.0	04.09.2006	Erster veröffentlichter Stand
R_1.1	11.09.2006	Möglichkeit zur Vorgabe von Sollwerten über den Profibus implementiert (ID 31,32,39,40)
R_1.2	31.10.2006	Infürmation über Hardware Adressen-Auswahl und Steckverbinder hinzugefügt
R_1.3	05.02.2007	Beispiel hinzugefügt für das Senden von Sollwerten via Profibus
R_1.4	06.02.2007	Infürmationen über Profibus Eigenschaften hinzugefügt
R_1.5	16.03.2007	Infürmation zum Steckverbinder für RS232 am DMA RJ45 Stecker hinzugefügt
R_1.6	25.10.2007	Parameter für SW Vyy.06 und SW V31.xx hinzugefügt, einschließlich Parameter für geregelte Versionen
R_1.7	10.01.2008	Multislave Operation einführt CMD=3, CMD=6 in Kopfzeilen hinzu. Neue Struktur eingeführt. CMD=15, Master Schreibbefehl für mehrere Parameter hinzugefügt
R_1.8	21.01.2008	Neues GSD file HCS0120.GSD eingeführt
R_1.9	14.02.2008	CMD= 15, Beispiel für Fehlerreaktion eingefügt. Allgemeine Beschreibung der Eigenschaften aus Spezifikation für Spanndruckregler übernommen (aus Revision R1.4)
R_2.0	01.03.2008	Überarbeitung. Neue Funktionen und Parameter für SW Versionen Vyy.07 implementiert
R_2.1	16.10.2008	Telegramm Sendebit D_IN1 und D_IN2 ergänzt
R_2.2	19.01.2009	SW Version V32.09x, Parameter E18 zugänglich und Wertebereich definiert
R_2.3	20.02.2009	SW Version ab V32.09c, Bit hinzugefügt, um den Fehler/Komarator Ausgang direkt zu setzen.
R_2.4	10.09.2010	SW Version ab V32.10a, Parameter C1.33 und C2.33 hinzu. Parameter 0xB0 disable card bit hinzugefügt.
R_2.5	16.01.2014	Überarbeitung S7-Implementation diverse Schreibfehler entfernt
R_2.6	27.01.2014	Hinzufügen Fehlerbehebung
R_2.7	14.04.2014	Hinzufügen Fehlertabellen, Reaktion auf Timeout Profibussignal
R_2.8	10.08.2015	Beschreibung Slave Numerierung und E22
R_2.8	22.08.2016	GSD Datei. HCS0220.gsd
R_2.9	26.10.2017	Korrigiertes Beispiel für SW15.xx (modus1) (A1.02 wurde A2.01)
R_3.0	11.12.2017	Korrigierter Antwortstring für SW32.xx (modus3+4) (d1.11 + d1.12 getauscht)
R_3.1	09.03.2019	Konfiguration für das Siemens TIA Portal eingefügt
R_3.1	15.03.2019	Update für CPU 1200,1500 in der TIA Portal Konfiguration

Inhaltsangabe

1	Eigenschaften.....	3
1.1	Verwendete GSD Datei.....	3
1.2	Unterstützte Software Versionen	3
1.3	Spezielle Information für Multi-Slave Nodes	3
1.4	Generelle Information zu TADR (Telegram Adresse)	3
1.5	Generelle Information über CMD (Befehl) und SADR (Slave Adresse)	4
2	Vorderansicht: Adress Anwahl und Steckverbinder	
2.1	Vorderansicht: Einzel-Modul Version (Node mit einem angeschlossenen Modul)	5
2.2	Vorderansicht: Multiple-Slave Version (Node mit mehreren angeschlossenen Slaves).....	6
3	Einfache Befehle.....	
3.1	CMD = 3, Master Read Parameter (7 Bytes)	7
3.2	CMD = 3, Antwort des Slave (4 + CNT Bytes)	7
3.3	CMD = 3, Fehler vom Slave (4 Bytes).....	8
3.4	CMD = 6, Master Write einzelner Parameter (7 Bytes)	9
3.5	CMD = 6, Antwort von Slave (7 bytes).....	9
3.6	CMD = 6, Fehler vom Slave (4 Bytes).....	10
3.7	Telegram Beispiele CMD = 6, CMD = 3.....	11
3.8	Zusätzliche Beispiele für CMD = 6.....	12
4	Komplexe Befehle.....	
4.1	CMD = 15, Master schreibt mehrere Parameter (3 + 5-Modul Bytes)	13
4.2	CMD = 15, Antwort des jeweiligen Slave (9 Bytes).....	14
4.3	CMD = 15, Fehler vom Slave (4 Bytes).....	14
4.4	Beispiele, CMD = 15	15
4.4.1	Single Slave (SW Version 15.xxy)	15
4.4.2	Multiple-Slave (3 Slaves [Module], SW Version 15.xxy)	15
4.5	CMD = 15, Struktur des Profibus Telegrams in Abhängigkeit der SW Version	17
4.5.1	Generell	17
4.5.2	Version: V15.xxy, gesteuert, ein Ventil mit zwei Magneten, , Modus 1	19
4.5.3	Version: V22.xxy, gesteuert, zwei Ventils mit jeweils einem Magnet, Modus 2	20
4.5.4	Version: V32.xxy, geregelt, Modus 3, 4	21
4.5.5	Version: V32.xxy, geregelt, Modus 6	22
4.5.6	Version: V32.xxy, geregelt, Modus 8	23
5	Siemens S7 Implementierung von "Multiple-Slave" (5 DMA Module)	
5.1	Hardware Konfiguration	24
5.1.1	Hardware Konfiguration bei Geräten ohne CMD15-Module in der GSD-Datei	28
5.2	CMD = 15, Protokoll Konfiguration.....	30
6	Parameterliste.....	
7	Fehlerbeseitigung	
7.1	generell:	Fehler! Textmarke nicht definiert.
7.2	Keine Kommunikation zum Modul möglich:	46
7.3	Kommunikation zum Modul ist möglich aber die Ausgänge sind inaktiv:	46

1 Eigenschaften

- Unterstützt Profibus-DP Slave in Übereinstimmung mit IEC 61158
- Unterstützt Profibus DPV1
- Maximal 244 Byte Eingangs- und 244 Byte Ausgangsdaten
- Unterstützt bis zu 12 Mbaud (auto detect)
- Galvanisch getrennt und opto-entkoppelt

1.1 Verwendete GSD Datei

„HCS0220.GSD“

1.2 Unterstützte Software Versionen

Version: V15.xxy, gesteuerte Anwendungen, ein Ventil mit zwei Magneten.
Version: V22.xxy, gesteuerte Anwendungen, zwei Ventile mit je einem Magneten.
Version: V31.xxy, Spannungsregler; siehe hierzu auch separate Dokumentation.
Version: V32.xxy, geregelte Versionen, Modus 3,4,6,8.

1.3 Spezielle Information für Multi-Slave Nodes

In Fällen bei denen ein Profibus-Node mehr als einen Slave (DMA-Modul) angeschlossen hat, müssen all diese Slaves auch aktiv, d.h. an die Versorgungsspannung angeschlossen sein. Das Enable Signal muss ebenso an allen Modulen angeschlossen werden. Andernfalls ist eine Kommunikation nicht möglich und der Node wird mit einer „Time-out“ Fehlermeldung reagieren.

Die Profibus Adressauswahl an der Front des Node-Moduls muss richtig eingestellt sein, dann erlischt auch die LED „Bus-Fehler“.

Wenn z.B. 3 (Slave) Modul verwendet werden, dann muss E22 beim ersten Modul links vom Node-Modul auf „1“, für das nächste Slave-Modul auf „2“ und für das dritte auf „3“ gestellt werden (siehe Kapitel 2.2).

1.4 Generelle Information zu TADR (Telegram Adresse)

TADR ist ein Datum, welches mit jedem Zyklus geschrieben (verändert) werden kann. Aber eine Änderung des Wertes ist nur Optional und nicht zwangsläufig erforderlich. Der Wert wird durch den DP-Master definiert.

Der Zweck liegt darin, entweder die Telegramm durch den Master zu kontrollieren oder um bei den Nodes das Senden eines neuen Antworttelegramms zu erreichen.

Das Design des Profibus-Node ist so ausgeführt, dass der Slave oder die Slaves (das DMA-Modul oder die DMA-Modul im Fall von Multi-Slave-Modulen) nur antworten, wenn eine empfangene Nachricht vom Master veränderte Daten – im Vergleich zur vorhergegangenen Nachricht – enthält.

Deshalb kann TADR verwendet werden, um eine Antwort von einem Slave zu „erzwingen“ und damit ein Update des augenblicklichen Status des Slave zu erhalten, obwohl der Rest der Daten des Telegrams unverändert geblieben ist.
Der gültige Wertebereich für TADR ist 0 bis 255 oder in Hexadezimal 0x00 bis 0xFF.

1.5 Generelle Information über CMD (Befehl) und SADR (Slave Adresse)

Verschiedene Wege der Kommunikation zwischen Master und Slaves sind möglich. Der Unterschied liegt im CMD Befehl.

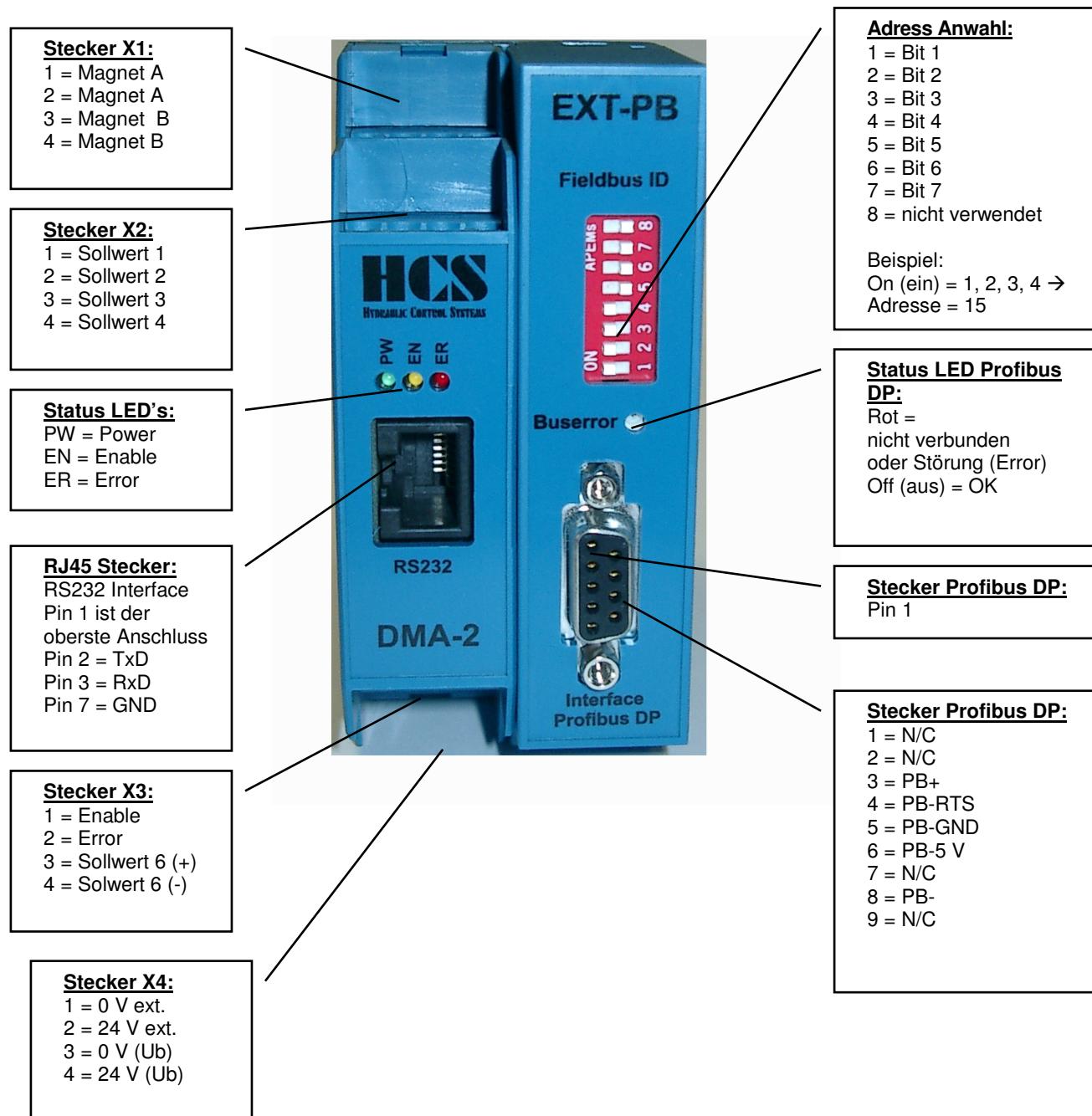
- CMD = 6: Schreiben einer einzelnen Parameter-ID eines Slaves (Modul) an einem Node. Die ausgewählte SADR im Telegram definiert welcher der Slaves für die Kommunikation gewählt wird. Im Slave (Modul) selbst ist die Adresse definiert durch den Wert in Parameter E22.
Sonderfall: Der Profibus-Node hat nur einen Slave (Modul) angeschlossen. In diesem Fall ist SADR auf "1" gesetzt (im Parameter E22). SADR könnte auch auf jeden anderen Wert zwischen 1 und 32 gesetzt werden.
Theoretisch können bis zu 32 Slaves (Modul) an einem Profibus-Node angeschlossen werden. Welcher der Slaves (Modul) dann ausgewählt wird, ist im Telegram wiederum durch SADR definiert und in den Slaves durch Parameter E22. Mehr Information hierzu finden sich im Abschnitt 3.1.
- CMD = 3: Lesen eines oder mehrerer Parameter mit Parameter-ID in steigender Folge. Alle anderen Definitionen von CMD = 6 sind ebenfalls anzuwenden. Mehr Information hierzu finden sich im Abschnitt 3.1.
- CMD = 15: Dieser Befehl ermöglicht das Schreiben eines (vordefinierten) Satzes von Parametern. Dieser Satz von Parametern selbst ist abhängig von der Software Version der Slaves (der Modul). Siehe auch hierzu auch Abschnitt 4.5.
Mit diesem Befehl können alle Slaves (Modul) an einem Profibus-Node mit der Adresse 1 bis 5 erreicht werden. Dieser Befehl kann nicht benutzt werden für Slaves (Modul) mit SADR > 5. In diesem Fall muss CMD = 3 und CMD = 6 verwendet werden.
Vorzugsweise ist SADR in den Slaves (Modulen) in ansteigender und durchgängiger Reihenfolge beginnend mit 1 (E22 = 1) zu setzen.
Im Telegram selbst ist die Nummer des Slaves (Moduls) am Profibus-Node definiert durch SNUM. Dies wird zugleich die Länge des Telegrams selbst bestimmen (Anzahl der Bytes im Telegram). Mehr Information findet sich im Abschnitt 4.

Es ist möglich die Befehle CMD = 3, CMD = 6 und CMD = 15 an einem Profibus-Node zu "mischen". So werden z.B. die Slaves (Modul) mit E22 = 1, E22 = 2 und E22 = 3 mit CMD = 15 oder/und CMD = 3 oder/und CMD = 6 angesprochen, aber ein anderer Slave (Modul) mit Adresse E22 größer als 5 kann nur mit den Befehlen CMD = 3 oder CMD = 6 angesprochen werden.

2 Vorderansicht: Adress Anwahl und Steckverbinder

2.1 Vorderansicht: Einzel-Modul Version (Node mit einem angeschlossenen Modul)

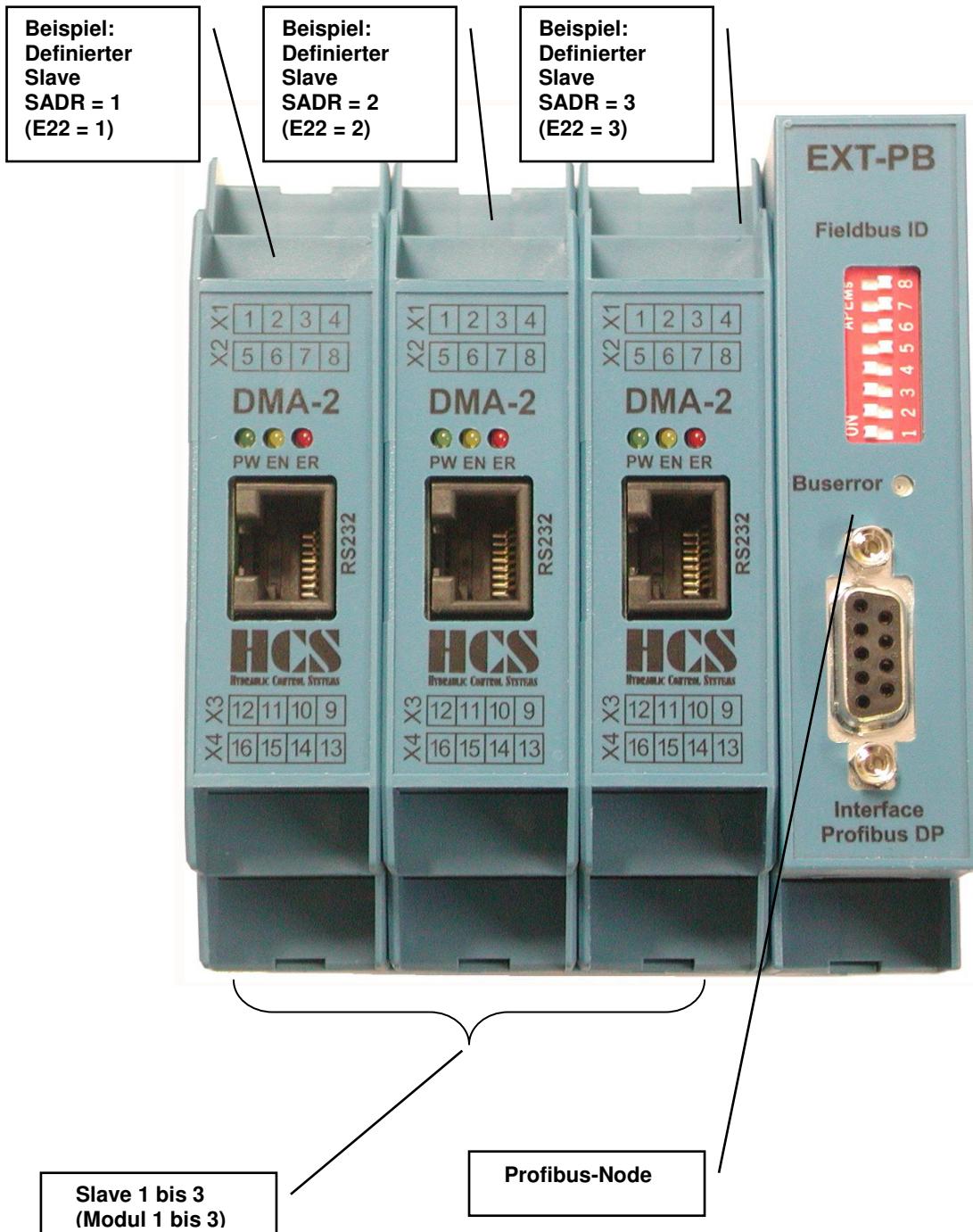
Beispiel für ein Modul in Ausführung DMA-22-M1-01-x!



2.2 Vorderansicht: Multiple-Slave Version (Node mit mehreren angeschlossenen Slaves)

Beispiel:

1 Profibus-Node und 3 DMA Modul (Slaves) angeschlossen



3 Einfache Befehle

3.1 CMD = 3, Master Read Parameter (7 Bytes)

Zuerst schreiben ...

Byte	Abkürzung	Beschreibung
0	TADR	Telegram Adresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
1	SADR	Slave Adresse
2	CMD	Befehl: 3 = Lesen mehrerer Parameter durch den Master
3	IDH	High byte Parameter ID des ersten Parameters
4	IDL	Low byte Parameter ID des ersten Parameters
5	N-high	High-Byte Anzahl der Parameter
6	N-low	Low-Byte Anzahl der Parameter

...zuletzt schreiben

TADR Telegram Adresse, definiert durch den DP-Master (Vorgabe durch Anwender).

SADR Slave Adresse, falls mehr als ein Slave am Profibus-Node angeschlossen ist
1 = Standard (single slave)
2 bis 32 = auch möglich (E22 legt die Slave Adresse fest)

CMD Befehl: 3 = Parameter lesen durch den Master

IDH,IDL H-Byte und Low-Byte der Parameter-ID in HEX
0x0000 .. 0x00A2 = gültiger ID-Bereich

N-high High-Byte Anzahl der Parameter (Wörter), normalerweise "0"

N-low Low-Byte Anzahl der Parameter (Wörter), 1 bis 8 (0x08)
(maximal sind 8 Parameter auf einmal zu lesen)

3.2 CMD = 3, Antwort des Slave (4 + CNT Bytes)

Zuerst schreiben ...

Byte	Abkürzung	Beschreibung
0	TADR	Telegram Adresse, durch den Lesebefehl vorgegeben, Bereich 0..255, 0..0xFF
1	SADR	Slave Adresse
2	CMD	Befehl: 3 = Lesen mehrerer Parameter durch den Master
3	CNT	N*2, Anzahl der Daten Bytes
4	DAT1H	High byte Wert des ersten Parameters
5	DAT1L	Low Byte Wert des ersten Parameters

falls CNT > 2:

CNT+2 = N	DATnH	High byte Wert des letzten Parameters
CNT+3 = N	DATnL	Low Byte Wert des letzten Parameters

...zuletzt schreiben

TADR Telegram Adresse, definiert durch den DP-Master (Vorgabe durch Lesebefehl).

SADR Slave Adresse, falls mehr als ein Slave am Profibus-Node angeschlossen ist
1 = Standard (single slave)
2 bis 32 = auch möglich (E22 legt die Slave Adresse fest)

CMD Befehl: 3 = Parameter lesen durch den Master

CNT N*2, Anzahl der Daten Bytes (Maximal 16 Daten Bytes)

IDH,IDL H-Byte und Low-Byte der Parameter-ID in HEX
0x0000 .. 0x00B0 = gültiger ID-Bereich

**DAT1H.. DAT nH,
DAT1L.. DAT nL,** Zwei Daten Bytes in HEX, ohne Dezimalzeichen
0x7FFF == +32767
0x8000 == -32768

3.3 CMD = 3, Fehler vom Slave (4 Bytes)

Zuerst schreiben ...

Byte	Abkürzung	Beschreibung
0	TADR	Telegram Adresse, durch den Lesebefehl vorgegeben, Bereich 0..255, 0..0xFF
1	SADR	Slave Adresse
2	ERR	Fehler Code
3	EXCE	Ausnahme

...zuletzt schreiben

ERR Fehler Code: 0x83 = Lesen fehlgeschlagen

EXCE Ausnahme:
1 = Befehl (Befehl) nicht unterstützt
2 = Falsche ID
3 = CNT== 0 oder CNT > 16 (0x10)
4 = Lesen der ID fehlgeschlagen
5 = timeout Slave, Adresse falsch oder nicht installiert oder falsche Anzahl angeschlossener Slaves (Modul)
6 = Interner Checksum Fehler (Modbus)

Beispiel an Fehler Antworten:

Modul 3 ist nicht installiert und **SADR** im Master Telegram ist auf „3“ gesetzt

Byte	Abkürzung	Abkürzung	Beschreibung
0	0x23	TADR	Antwort Telegram Adresse (durch den Lesebefehl vorgegeben)
1	0x03	SADR	Aktuelle Slave Adresse
2	0x83	ERR	Fehler, Bit „7“ mit mit Befehl 0x03 ist gesetzt
3	0x05	EXCE	Timeout Slave, Adresse falsch oder nicht installiert

3.4 CMD = 6, Master Write einzelner Parameter (7 Bytes)

Zuerst schreiben...

Byte	Abkürzung	Beschreibung
0	TADR	Telegram Adresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
1	SADR	Slave Adresse
2	CMD	Befehl: 6 = Schreiben eines einzelnen Parameters durch den Master
3	IDH	High Byte Parameter ID
4	IDL	Low Byte Parameter ID
5	DATH	High Byte des Parameter Wertes
6	DATL	Low Byte des Parameter Wertes

...zuletzt schreiben

TADR	Telegram Adresse, definiert durch den DP-Master (Vorgabe durch Anwender).
SADR	Slave Adresse, falls mehr als ein Slave am Profibus-Node angeschlossen ist 1 = Standard (single slave) 2 bis 32 = auch möglich (E22 legt die Slave Adresse fest)
CMD	Befehl: 6 = Einzelnen Parameter schreiben durch den Master
IDH,IDL	H-Byte und Low-Byte der Parameter-ID in HEX 0x0000 .. 0x00B0 = gültiger ID-Bereich
DATH,DATL	Zwei Daten Bytes in HEX, ohne Dezimalzeichen 0x7FFF == +32767 0x8000 == -32768

3.5 CMD = 6, Antwort von Slave (7 bytes)

Zuerst schreiben...

Byte	Abkürzung	Beschreibung
0	TADR	Telegram Adresse, durch den Schreibbefehl vorgegeben, Bereich 0..255, 0..0xFF
1	SADR	Slave Adresse
2	CMD	Befehl: 6 = Schreiben eines einzelnen Parameters durch den Master
3	IDH	High byte parameter ID
4	IDL	Low byte parameter ID
5	DATH	High Byte des Parameter Wertes
6	DATL	Low Byte des Parameter Wertes

...Zuletzt schreiben

TADR	Telegram Adresse, definiert durch den DP-Master (Vorgabe durch Schreibbefehl)..
SADR	Slave Adresse, falls mehr als ein Slave am Profibus-Node angeschlossen ist 1 = Standard (single slave) 2 bis 32 = auch möglich (E22 legt die Slave Adresse fest)
CMD	Befehl: 6 = Einzelnen Parameter schreiben durch den Master
IDH,IDL	H-Byte und Low-Byte der Parameter-ID in HEX 0x0000 .. 0x00A2 gültiger ID-Bereich
DATH,DATL	Zwei Daten Bytes in HEX, ohne Dezimalzeichen 0x7FFF == +32767 0x8000 == -32768

3.6 CMD = 6, Fehler vom Slave (4 Bytes)

Zuerst schreiben...

Byte	Abkürzung	Beschreibung
0	TADR	Telegram Adresse, durch den Schreibbefehl vorgegeben, Bereich 0..255, 0..0xFF
1	SADR	Slave Adresse
2	ERR	Fehler Code
3	EXCE	Ausnahme

...Zuletzt schreiben

ERR Fehler Code: 0x86 = schreiben fehlgeschlagen

EXCE Ausnahme:
1 = Befehl (Befehl) nicht unterstützt
2 = Falsche ID
3 = CNT== 0 oder CNT > 16 (0x10)
4 = Lesen der ID fehlgeschlagen
5 = timeout Slave, Adresse falsch oder nicht installiert oder falsche Anzahl angeschlossener Slaves (Modul)
6 = Interner Checksum Fehler (Modbus)

Beispiel an Fehler Antworten:

Modul 3 ist nicht installiert und **SADR** im Master Telegram ist auf „3“ gesetzt

Byte	Abkürzung	Abkürzung	Beschreibung
0	0x23	TADR	Antwort Telegram Adresse (durch den Schreibbefehl vorgegeben)
1	0x03	SADR	Aktuelle Slave Adresse
2	0x86	ERR	Fehler, Bit „7“ mit mit Befehl 0x03 ist gesetzt
3	0x05	EXCE	Timeout Slave, Adresse falsch oder nicht installiert

3.7 Telegram Beispiele CMD = 6, CMD = 3

Master möchte den Parameter “C1.07” mit dem Wert 1.000 V schreiben:

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

Byte	Wert	Abkürzung
0	0x23	TADR
1	0x01	SADR
2	0x06	CMD
3	0x00	IDH
4	0x30	IDL
5	0x03	DATL
6	0xE8	DATH

Antwort vom Slave (Modul), Befehl wurde erfolgreich ausgeführt.

ID = 0x0030
DATA = 0x3E8 (= 1.000 V)

Byte	Wert	Abkürzung
0	0x23	TADR
1	0x01	SADR
2	0x06	CMD
3	0x00	IDH
4	0x30	IDL
5	0x03	DATL
6	0xE8	DATH

Master möchte Parameter “d1.07” lesen:

ID = 0x0007
N = 1, ein Parameter
TADR = 0x24
SADR = 0x01

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

Antwort vom Slave (Modul)

CNT = 2 (2 Bytes)

DATA = 0x0133 (==0.307Af)

Byte	Wert	Abkürzung
0	0x24	TADR
1	0x01	SADR
2	0x03	CMD
3	0x02	CNT
4	0x01	DAT1H
5	0x33	DAT1L

3.8 Zusätzliche Beispiele für CMD = 6

Beispiele für Sollwertvorgabe über den Profibus (ID = 0x0027):

1.) Sollwert 0.000 V = 0x0000:

Byte	Wert	Abkürzung
0	0x12	TADR
1	0x01	SADR
2	0x06	CMD
3	0x00	IDH
4	0x27	IDL
5	0x00	DATL
6	0x00	DATH

2.) Sollwert 5.000 V = 0x1388:

Byte	Wert	Abkürzung
0	0x12	TADR
1	0x01	SADR
2	0x06	CMD
3	0x00	IDH
4	0x27	IDL
5	0x13	DATL
6	0x88	DATH

3.) Sollwert 9.999 V = 0x270F:

Byte	Wert	Abkürzung
0	0x12	TADR
1	0x01	SADR
2	0x06	CMD
3	0x00	IDH
4	0x27	IDL
5	0x27	DATL
6	0x0F	DATH

4.) Sollwert -9.999 V = 0xD8F1

Byte	Wert	Abkürzung
0	0x12	TADR
1	0x01	SADR
2	0x06	CMD
3	0x00	IDH
4	0x27	IDL
5	0xD8	DATL
6	0xF1	DATH

4 Komplexe Befehle

4.1 CMD = 15, Master schreibt mehrere Parameter (3 + 5-Modul Bytes) (Schneller Multi-Slave Befehl)

Parameter E22 in jedem DMA-Modul muss auf den korrekten Wert im Bereich von 1 bis 5, abhängig von der Anzahl der angeschlossenen Slaves (Module), gesetzt werden. Der Slave an der linken Seite bekommt die Adresse „1“ und der Slave am nächsten zum Profibusknoten bekommt die höchste Adresse.

Beispiel: Wenn 3 Module verwendet werden, dann muss E22 für das erste Modul an der linken Seite auf „1“ gesetzt werden. E22 = „2“ für das zweite Modul in der Mitte, und „3“ für das dritte Modul am nächsten zum Profibusknoten (siehe auch 2.2).

Antworttelegramme werden automatisch vom Profibusknoten gemultiplext (siche auch 4.4.2).

Zuerst schreiben...

Adresse	Byte	Struktur					Abkürzung	Beschreibung	
0		0..0xFF					TADR	Telegram Adresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF	
1		1	2	3	4	5	SNUM	Anzahl der angeschlossenen Slaves (Maximal 5)	
2		15					CMD	Befehl: 15 = schreiben mehrerer Parameter durch den Master	
3		1	1	1	1	1	DAT1.1	Byte für spezielle Funktionen	
4							DAT1.2H	High byte des ersten „analogen“ Wertes	
5							DAT1.2L	Low byte des ersten „analogen“ Wertes	
6							DAT1.3H	High byte des zweiten „analogen“ Wertes	
7							DAT1.3L	Low byte des zweiten „analogen“ Wertes	
8		2	2	2	2	2	DAT2.1	Byte für spezielle Funktionen	
9							DAT2.2H	High byte des ersten „analogen“ Wertes	
10							DAT2.2L	Low byte des ersten „analogen“ Wertes	
11							DAT2.3H	High byte des zweiten „analogen“ Wertes	
12							DAT2.3L	Low byte des zweiten „analogen“ Wertes	
13		3	3	3	3	3	DAT3.1	Byte für spezielle Funktionen	
14							DAT3.2H	High byte des ersten „analogen“ Wertes	
15							DAT3.2L	Low byte des ersten „analogen“ Wertes	
16							DAT3.3H	High byte des zweiten „analogen“ Wertes	
17							DAT3.3L	Low byte des zweiten „analogen“ Wertes	
18		4	4	4	4	4	DAT4.1	Byte für spezielle Funktionen	
19							DAT4.2H	High byte des ersten „analogen“ Wertes	
20							DAT4.2L	Low byte des ersten „analogen“ Wertes	
21							DAT4.3H	High byte des zweiten „analogen“ Wertes	
22							DAT4.3L	Low byte des zweiten „analogen“ Wertes	
23		5	5	5	5	5	DAT5.1	Byte für spezielle Funktionen	
24							DAT5.2H	High byte des ersten „analogen“ Wertes	
25							DAT5.2L	Low byte des ersten „analogen“ Wertes	
26							DAT5.3H	High byte des zweiten „analogen“ Wertes	
27							DAT5.3L	Low byte des zweiten „analogen“ Wertes	
....Zuletzt schreiben		8	13	18	23	28	Gesamte Telegramlänge in Bytes		

TADR

Telegram Adresse, definiert durch den DP-Master (Vorgabe durch Anwender).. Range 0..255

SNUM

Anzahl der angeschlossenen Slaves (Maximal 5).

Hier muss für Multi-Slave Anwendungen die Anzahl vorhandener Slaves richtig eingetragen sein.

CMD

Befehl: 15 = schreiben mehrerer Parameter durch den Master

DATx¹.1

Ein Daten Byte in HEX, für spezielle Funktionen

0x00 == 0

0xFF == 255

DATx¹.2H, DATx¹.2L

Zwei Daten Bytes in HEX, ohne Dezimalzeichen, für „analogen“ Wert

0x7FFF == +32767

0x8000 == -32768

DATx¹.3H, DATx¹.3L

Zwei Daten Bytes in HEX, ohne Dezimalzeichen, für „analogen“ Wert

0x7FFF == +32767

0x8000 == -32768

¹) = Moduladresse (gesetzt im Parameter E22 in jedem der DMA-Modul); Bereich 1 bis 5.

4.2 CMD = 15, Antwort des jeweiligen Slave (9 Bytes)

Es ist zu beachten, dass bei Multiple-Slave Anwendungen SADR automatisch mit jeder neuen Antwort geändert wird.

Beispiel:

Bei Verwendung von 3 Slaves wird der Slave (Modul) mit der Adresse „1“ (E22 = 1) als erstes antworten. Nach dem nächsten „Master-Write“ kommt die Antwort dann vom Slave (Modul) mit der Adresse „2“ (E22 = 2) und zuletzt dann der Slave (Modul) mit der Adresse „3“ (E22 = 3). Dann fängt der Ablauf wieder mit Slave „1“ an.

D.h., der Multiplexer für die Antworten ist im Profibus-Node implementiert. SADR zeigt an, welcher der Slaves (Modul) die aktuelle Antwort geschickt hat.

Write first...

Adresse	Byte	Struktur	Abkürzung	Beschreibung	
0	Telegram header		TADR	Telegram Adresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF	
1			SADR	Slave Adresse	
2			CMD	Befehl: 15 = Schreiben mehrerer Parameter durch den Master	
3	Data of SADR		DAT.1H	Status Word	High byte des Slave (Modul) Status
4			DAT.1L		
5			DAT.2H	VALUE1	High byte des ersten „analogen“ Wertes
6			DAT.2L		
7			DAT.3H	VALUE2	High byte des zweiten „analogen“ Wertes
8			DAT.3L		

...Write last

TADR Telegram Adresse, definiert durch den DP-Master (Vorgabe durch Anwender)..

SADR Slave Adresse, falls mehr als ein Slave am Profibus-Node angeschlossen ist
1 bis 5 in Abhängigkeit von der Anzahl der Slaves

CMD Befehl: 15 = Schreiben mehrerer Parameter durch den Master

DAT.1H,DAT.1L Zwei Daten Bytes in HEX, für Slave (Modul) Status
0x0000 == 0
0xFFFF == 65535

**DAT.2H, DAT.2L,
DAT.3H, DAT.3L** VALUE1
VALUE2. Zwei Daten Bytes in HEX, ohne Dezimalzeichen, für analogen Wert
0x7FFF == +32767
0x8000 == -32768

4.3 CMD = 15, Fehler vom Slave (4 Bytes)

Zuerst schreiben...

Adresse	Byte	Abkürzung	Beschreibung
0	TADR	Telegram Adresse, durch den Schreibbefehl vorgegeben, Range 0..255, 0..0xFF	
1	SADR	Slave Adresse	
2	ERR	Fehler code	
3	EXCE	Exception	

...Zuletzt schreiben

ERR Fehler code: 0x8F = write failed

EXCE Exception:
1 = Befehl not supported
2 = Wrong ID
3 = Wrong Data Wert
4 = Writing of ID failed
5 = timeout (Modbus)
6 = Internal checksum Fehler (Modbus)
7 = Invalid number of connected slaves
8 = timeout slave, Adresse wrong or not installed
9 = Profibus Knoten (SADR = 0) schickt „Empfangs Datenpuffer zu klein“. Die definierte Datenlänge des GSD modes ist zu gering gewählt.

Beispiel für eine Fehler Antwort:

Modul 3 ist nicht installiert und SNUM im Mastertelegram ist auf „3“ gesetzt

Byte	Abkürzung	Abkürzung	Beschreibung
0	0x23	TADR	Telegram Adresse, durch den Schreibbefehl vorgegeben, Bereich 0..255, 0..0xFF
1	0x03	SADR	Adresse des aktuellen Slave (Modul)
2	0x8F	ERR	Fehler, Bit „7“ mit Befehl 0x0F in Dezimal 15 ist gesetzt
3	0x08	EXCE	Timeout Slave, Adresse falsch oder nicht installiert

4.4 Beispiele, CMD = 15

4.4.1 Single Slave (SW Version 15.xxy)

Der Adressenschalter an der Vorderseite des Profibus-Node ist auf den richtigen Wert gesetzt (BusFehler LED ist aus). E22 im Slave (Modul) ist auf 1 gesetzt. Hardware Enable ist mit dem Slave (Modul) verbunden.

Master schreibt Sollwert von 1.000 V zum Slave (Modul):

Byte	Abkürzung	Abkürzung	Beschreibung
0	0x23	TADR	Telegram Adresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
1	0x01	SNUM	0x01 bedeutet einzelner Slave (Modul)
2	0x0F	CMD	0x0F in Dezimal 15
3	0x00	DAT1.1	Byte für spezielle Funktionen , 0x00 (Default Wert, kein Bus-Disable)
4	(1000)	DAT1.2H	Sollwert A1.01
5		DAT1.2L	0x03E8 (= 1000 in Dezimal = 1.000V)
6	(0)	DAT1.3H	Sollwert A2.01
7		DAT1.3L	0x0000 (= 0 in Dezimal = 0.000V)

Antwort Slave (Modul) Nummer 1:

Byte	Abkürzung	Abkürzung	Beschreibung
0	0x23	TADR	Antwort Telegram Adresse, durch den Schreibbefehl vorgegeben, Bereich 0..255, 0..0xFF
1	0x01	SADR	Aktuelle Slave (Modul) Adresse
2	0x0F	CMD	0x0F in Dezimal 15
3	0x40	DAT1.1H	High Byte des Slave (Modul) Status, Hardware Enable ist Aktiv, kein Fehler
4	0x00	DAT1.1L	Low Byte des Slave (Modul) Status, nicht verwendet, in diesem Fall immer = 0
5	(0)	DAT1.2H	Aktueller Strom A
6		DAT1.2L	0x0000 (= 0 in Dezimal = 0.000A)
7	(270)	DAT1.3H	Aktueller Strom B
8		DAT1.3L	0x010E (= 270 in Dezimal = 0.270A)

(x) = Wert in Dezimal.

4.4.2 Multiple-Slave (3 Slaves [Module], SW Version 15.xxy)

Der Adressenschalter an der Vorderseite des Profibus-Node ist auf den richtigen Wert gesetzt (BusFehler LED ist aus). E22 der drei Slaves (Module) ist auf 1, 2 und 3 gesetzt.

Alle Slaves (Module) sind eingeschaltet und Hardware Enable ist mit allen Slaves (Modulen) verbunden.

Master erster Schreibzyklus: Sollwert 1.000 V an Slave1, 2.000 V an Slave2 und - 3.000 V an Slave3:

Byte	Abkürzung	Abkürzung	Beschreibung
0	0x23	TADR	Telegram Adresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
1	0x03	SNUM	0x03 bedeutet, dass 3 Slaves (Module) angeschlossen sind
2	0x0F	CMD	0x0F in Dezimal 15
3	0x00	DAT1.1	Byte für spezielle Funktionen, 0x00 (Default Wert, kein Bus-Disable)
4	(1000)	DAT1.2H	Sollwert A1.01
5		DAT1.2L	0x03E8 (= 1000 in Dezimal = 1.000V)
6	(0)	DAT1.3H	Sollwert A2.01
7		DAT1.3L	0x0000 (= 0 in Dezimal = 0.000V)
8	0x00	DAT2.1	Byte für spezielle Funktionen, 0x00 (Default Wert, kein Bus-Disable)
9	(2000)	DAT2.2H	Sollwert A1.01
10		DAT2.2L	0x07D0 (= 2000 in Dezimal = 2.000V)
11	(0)	DAT2.3H	Sollwert A2.01
12		DAT2.3L	0x0000 (= 0 in Dezimal = 0.000V)
13	0x00	DAT3.1	Byte für spezielle Funktionen, 0x00 (Default Wert, kein Bus-Disable)
14	(-3000)	DAT3.2H	Sollwert A1.01
15		DAT3.2L	0xF448 (= -3000 in Dezimal = - 3.000V)
16	(0)	DAT3.3H	Sollwert A2.01
17		DAT3.3L	0x0000 (= 0 in Dezimal = 0.000V)

(x) = Wert in Dezimal.

Der Profibus-Node wird nur eine Antwort senden, wenn irgendein Datum im Telegram vom Master anders ist als im vorhergehenden Telegram! Falls die Daten in einem Telegram an sich unverändert bleiben sollen, dann kann eine Antwort vom Profibus-Node dadurch „erzwungen“ werden, dass das Datum TADR verändert wird.

Das Datum des Write Befehls wird sofort an alle Slaves (Module) weitergegeben. Jedes Antwort-Telegram auf einen Write-Befehl wird automatisch „neue Daten“ vom nächsten Slave (Modul) enthalten. Die Slaves (Module) werden automatisch gemultiplexed.

Antwort Telegrame sind Automatisch gemultiplexed durch den Profibus-Node.

In diesem Beispiel antwortet Slave (Modul) 1 zuerst

Byte	Abkürzung	Abkürzung	Beschreibung
0	0x23	TADR	Antwort Telegram Adresse, durch den Schreibbefehl vorgegeben, Bereich 0..255, 0..0xFF
1	0x01	SADR	Aktuelle Slave (Modul) Adresse
2	0x0F	CMD	0x0F in Dezimal 15
3	0x40	DAT1.1H	High Byte des Slave (Modul) Status, Hardware Enable is Aktiv, no Fehler
3	0x00	DAT1.1L	Low Byte des Slave (Modul) Status, nicht verwendet, in diesem Fall immer = 0
4	(0)	DAT1.2H	Aktueller Strom A, d1.07 of Modul 1
5		DAT1.2L	0x0000 (= 0 in Dezimal = 0.000A)
6	(270)	DAT1.3H	Aktueller Strom B, d1.08 of Modul 1
7		DAT1.3L	0x010E (= 270 in Dezimal = 0.270A)

(x) = Wert in Dezimal.

Verändern der Telegram Adresse TADR auf 0x24 und die selbe Nachricht senden wie zuvor.

Der nächste Slave (Modul) Nummer "2" wird antworten.

Byte	Abkürzung	Abkürzung	Beschreibung
0	0x24	TADR	Antwort telegram Adresse (durch den Schreibbefehl vorgegeben)
1	0x02	SADR	Aktuelle Slave Adresse
2	0x0F	CMD	0x0F in Dezimal 15
3	0x40	DAT1.1H	High byte des Slave (Modul) Status, Hardware Enable is Aktiv, no Fehler
3	0x00	DAT1.1L	Low byte des Slave (Modul) Status, nicht verwendet, in diesem Fall immer = 0
4	(0)	DAT1.2H	Aktueller Strom A, d1.07 des Slave (Modul) 2
5		DAT1.2L	0x0000 (= 0 in Dezimal = 0.000A)
6	(540)	DAT1.3H	Aktueller Strom B, d1.08 des Slave (Modul) 2
7		DAT1.3L	0x021C (= 540 in Dezimal = 0.540A)

(x) = Wert in Dezimal.

Verändern der Telegram Adresse TADR auf 0x25 und die selbe Nachricht senden wie zuvor.

Der nächste Slave (Modul) Nummer "3" wird antworten.

Byte	Abkürzung	Abkürzung	Beschreibung
0	0x25	TADR	Antwort Telegram Adresse, durch den Schreibbefehl vorgegeben, Bereich 0..255, 0..0xFF
1	0x03	SADR	Aktuelle Slave Adresse
2	0x0F	CMD	0x0F in Dezimal 15
3	0x40	DAT1.1H	High Byte des Slave (Modul) Status, Hardware Enable is Aktiv, kein Fehler
3	0x00	DAT1.1L	Low Byte des Slave (Modul) Status, nicht verwendet, in diesem Fall immer = 0
4	(810)	DAT1.2H	Aktueller Strom A, d1.07 des Slave (Modul) 3
5		DAT1.2L	0x032A (= 810 in Dezimal = 0.810A)
6	(0)	DAT1.3H	Aktueller Strom B, d1.08 des Slave (Modul) 3
7		DAT1.3L	0x0000 (= 0 in Dezimal = 0.000A)

(x) = Wert in Dezimal.

Verändern der Telegram Adresse TADR auf 0x26 und die selbe Nachricht senden wie zuvor.

Der nächste Slave (Modul) Nummer "1" wird antworten.

Byte	Abkürzung	Abkürzung	Beschreibung
0	0x26	TADR	Antwort Telegram Adresse, durch den Schreibbefehl vorgegeben, Bereich 0..255, 0..0xFF
1	0x01	SADR	Aktuelle Slave Adresse
2	0x0F	CMD	0x0F in Dezimal 15
3	0x40	DAT1.1H	High Byte des Slave (Modul) Status, Hardware Enable is Aktiv, kein Fehler
3	0x00	DAT1.1L	Low Byte des Slave (Modul) Status, nicht verwendet, in diesem Fall immer = 0
4	(0)	DAT1.2H	Aktueller Strom A, d1.07 des Slave (Modul) 1
5		DAT1.2L	0x000 (= 0 in Dezimal = 0.000A)
6	(270)	DAT1.3H	Aktueller Strom B, d1.08 des Slave (Modul) 1
7		DAT1.3L	0x010E (= 270 in Dezimal = 0.270A)

(x) = Wert in Dezimal.

4.5 CMD = 15, Struktur des Profibus Telegrams in Abhängigkeit der SW Version

4.5.1 Generell

4.5.1.1 Fehlermeldung (angezeigt durch “Fehler occurred”)

Fehler wie z.B. „Überstrom“ oder „Drahtbruch“, welche durch das Bit “Fehler occurred” angezeigt werden, können durch einen der folgenden Schritte quittiert werden:

- Eingangssignal für Hardware-Enable am zugehörigen Slave (Modul) am Terminal X3/1 (9) zuerst auf “Low” setzen und danach wieder auf “High”
oder
- setzen und rücksetzen des Bits (0x80) Bus-Disable

Notwendige Bedingung für das rücksetzen eines Fehler ist, dass die Ursache für den Fehler zuerst behoben wurde.

Fehlernummern:

Die entsprechenden Fehlernummern sind im jeweiligen Handbuch der DMA-Baugruppen zu finden.

Generell gilt aber folgende Fehlertabelle:

Slave (Modul) Status	Hi-Byte (DAT.1H)	Low-Byte (DAT.1L)	Description
	Hi-Byte (DAT.1H)	Low-Byte (DAT.1L)	
0x0C	0x01		Allgemeiner Fehler
0x0C	0x02		Drahtbruch des Analog-Einganges S1.06 erkannt (Stromeingang <2mA)
0x0C	0x03		Kurzschluss der Ausgangsstufen erkannt (Spitzenstrom >8A!)
0x0C	0x04		Drahtbruch des Analog-Einganges S1.05 erkannt (Stromeingang <2mA)
0x0C	0x05		Überstrom am Analog-Eingang S1.06 erkannt (Stromeingang >25mA)
0x0C	0x06		Überstrom am Analog-Eingang S1.05 erkannt (Stromeingang >25mA)
0x0C	0x07		Nicht verwendet-
0x0C	0x08		Überstrom, Kurzschluss oder Unterbrechung an den Endstufen. Die Differenz zwischen Sollstrom und Iststrom liegt außerhalb des definierten Bereichs. Bedingungen für die Fehlerauslösung: $ Fehler > 250 \text{ mA}$ für $t_{Fehler} \text{ ca. } 100 \text{ ms}$.
0x0C	0x09		Eingangssignal @ Sollwert S1.06 und Auswahl E15 = 8 (5 ...+/- 4,5 V) auserhalb des gültigen Bereiches (approx. < 0,43 V / > 9,58 V)
0x0C	0x0A		TimeoutFehler -no Profibus Fehler erkannt- (Modul mit Slave.Adresse antwortet nicht (Parameter E23 der angeschlossenen Module überprüfen))

Softwareversionen welche den Fehler „10“ bzw. „0x0A“ **nicht** unterstützen, reagieren mit blinkender Enable LED auf zu langen Pausen zwischen gültigen Profibustelegrammen. Diese Versionen arbeiten bei Erhalt eines gültigen Profibustelegrammes normal weiter. Hier ist keine separate Quittierung mittels Enablesignal notwendig.

Softwareversionen welche den Fehler „10“ bzw. „0x0A“ unterstützen, reagieren mit leuchtender Fehler LED auf zu langen Pausen zwischen gültigen Profibustelegrammen. Diese Versionen arbeiten bei Erhalt eines gültigen Profibustelegrammes **nicht** normal weiter. Hier muss eine separate Quittierung mittels Enablesignal erfolgen.

Generell werden im Fehlerzustand angelegte Sollwerte von der Elektronik verworfen und die Ausgänge abgeschaltet.

4.5.1.2 Erklärung für “Hardware Enable” und “Software Disable”:

Generell gesagt muss das Signal „Hardware Enable“ vorhanden sein („High“ Level), um ein Signal (Strom) an den Endstufen erhalten zu können. Neben dem Hardware Enable kann auch ein „Software Disable“ verwendet werden, um das Hardware Enable Signal zu umgehen. Damit kann der Slave (Modul) disabled werden, Das bedeutet, dass das Signal „Hardware Enable“ fest verdrahtet werden kann und dann die Funktion „Software Disable“ verwendet werden kann, um die Endstufen abzuschalten oder auch um Fehler zu quittieren!

4.5.1.3 Erklärung spezieller Funktionen und Slave (Modul) Status Bits

Achtung: Funktions- und Slave (Modul) Status Bits hängen von der verwendeten Software version ab

HW_ENABLE	Das Signal "D-In Enable" (HW Enable) ist Aktiv am Slave (Modul). Ein Fehler ist im Slave (Modul) aufgetreten. Verschiedene Ursachen können dies verursachen.
ERROR	Weitere Informationen finden sich in den zugehörigen Bedienungsanleitungen für die DMA-Module.
BUS_DISABLE	Das Enable Signal wurde über den Profibus zurückgesetzt (Slave [Modul] ist disabled).
Din_1	Die "Enable Funktion" für den „schaltenden Regler“ (diese Funktion schaltet um, zwischen gesteuert und geregelt) mittels des Komparator „KOMP_1“ ist freigegeben und kann verwendet werden. Zusätzliche notwendige Bedingung: C1.00 muss auf 2 oder 4 gesetzt sein! Anmerkung: d1.12 und d1.13 bleiben auf „0.000 V“ solange die Bedingung zur Aktivierung „Din_1“ nicht gesetzt wurde. Dies verhindert eine vorzeitige Aktivierung des Komparator „KOMP_1“.
Dout_1	Zeigt an, dass die Funktion "schaltender Regler" aktiv ist. Der Slave (Modul) arbeitet nun in der Regelung. Zusätzliche notwendige Bedingung für den Übergang von offenem zu geschlossenem Regelkreis: C1.00 muss entweder auf 2 oder 4 gesetzt sein. Signal Din_1 muss vorhanden sein Komparator KOMP_1 muss außerhalb des Fensters sein und demzufolge ist das Signal KOMP_1 = 0.
DKOMP_1	Der Slave (Modul) bleibt solange im Zustand "geregelt", solange auch das Signal Din_1 auf "High" bleibt. Ausgangsignal des Komparator „KOMP_1“. Wird verwendet um das Signal, definiert durch C1.25 zu überwachen. Dieses Signal kann auch verwendet werden, um von gesteuert auf geregelt zu schalten. 0 = Komparator ist nicht Aktiv (ausserhalb der definierten Grenzen und Zeitverzögerungen) 1 = Komparator ist Aktiv (innerhalb der definierten Grenzen und Zeitverzögerungen).
DKOMP_11	Ausgangsignal des Komparator „KOMP_11“, Wird verwendet um das Signal, definiert durch C1.32 zu überwachen. 0 = Komparator ist nicht Aktiv (ausserhalb der definierten Grenzen und Zeitverzögerungen) 1 = Komparator ist Aktiv (innerhalb der definierten Grenzen und Zeitverzögerungen).
Din_2	Die "Enable Funktion" für den „schaltenden Regler“ (diese Funktion schaltet um, zwischen gesteuert und geregelt) mittels des Komparator „KOMP_2“ ist freigegeben und kann verwendet werden. Zusätzliche notwendige Bedingung: C2.00 muss auf 2 oder 4 gesetzt sein! Anmerkung: d2.12 und d2.13 bleiben auf „0.000 V“ solange die Bedingung zur Aktivierung „Din_2“ nicht gesetzt wurde. Dies verhindert eine vorzeitige Aktivierung des Komparator „KOMP_2“.
Dout_2	Zeigt an, dass die Funktion "schaltender Regler" aktiv ist. Der Slave (Modul) arbeitet nun in der Regelung. Zusätzliche notwendige Bedingung für den Übergang von offenem zu geschlossenem Regelkreis: C2.00 muss entweder auf 2 oder 4 gesetzt sein. Signal Din_2 muss vorhanden sein Komparator KOMP_2 muss außerhalb des Fensters sein und demzufolge ist das Signal KOMP_2 = 0.
DKOMP_2	Der Slave (Modul) bleibt solange im Zustand "geregelt", solange auch das Signal Din_2 auf "High" bleibt. Ausgangsignal des Komparator „KOMP_2“. Wird verwendet um das Signal, definiert durch C2.25 zu überwachen. Dieses Signal kann auch verwendet werden, um von gesteuert auf geregelt zu schalten. 0 = Komparator ist nicht Aktiv (ausserhalb der definierten Grenzen und Zeitverzögerungen) 1 = Komparator ist Aktiv (innerhalb der definierten Grenzen und Zeitverzögerungen).
DKOMP_22	Ausgangsignal des Komparator „KOMP_22“, Wird verwendet um das Signal, definiert durch C2.32 zu überwachen. 0 = Komparator ist nicht Aktiv (ausserhalb der definierten Grenzen und Zeitverzögerungen) 1 = Komparator ist Aktiv (innerhalb der definierten Grenzen und Zeitverzögerungen).
TST_CMP	Damit wird der Error / Comp-Ausgang direkt auf 24V gesetzt (Die Signalisierung des Fehlerausgangs wird damit deaktiviert). Dient zum Testen des Komparator oder Fehler-ausganges.
CNTRL_1..4	Reserviert für Sonderfunktionen
STAT_1..5	Reserviert für Sonderfunktionen

4.5.2 Version: V15.xxy, gesteuert, ein Ventil mit zwei Magneten, , Modus 1

Telegram Struktur Beispiel mit einem Slave (Modul, SNUM = 1),
Master schreibt mehrere Parameter (3 + 5·Modul Bytes)

Zuerst schreiben...

Anmerkung	Byte	Struktur	Abkürzung	Beschreibung
	0	Telegramm Kopf	TADR	Telegrammadresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
	1		SNUM	Anzahl der angeschlossenen Slaves (Module) (Maximal 5)
	2		CMD	Befehl: 15 = schreiben mehrerer Parameter durch den Master
	3	Modul Daten	DAT1.1	Steuer-Byte Byte für spezielle Funktionen
	4		DAT1.2H	VALUE1 = A1.01
	5		DAT1.2L	Sollwert Eingang (Profibus)
	6		DAT1.3H	VALUE2 = A2.01 (normal nicht benötigt)
	7		DAT1.3L	Sollwert Eingang (Profibus)
....Zuletzt schreiben	8			Gesamte Telegrammlänge in Bytes

Erklärung des Byte für spezielle Funktionen (**Steuer-Byte**)

Beschreibung	Abkürzung	Funktion des einzelnen Bits
Byte für spezielle Funktionen	DATx.1	<p>Kann nur gesetzt werden wenn HW_ENABLE aktiv ist. 0x80 = BUS_DISABLE (Freigabe via Profibus sperren) In diesem Modus nicht verfügbar, reserviert:</p> <p>0x01 = Din_1 0x02 = CNTRL_1 0x04 = TST_CMP 0x08 = CNTRL_2 0x10 = Din_2 0x20 = CNTRL_3 0x40 = CNTRL_4</p> <p>Reset = 0x00 (Enable nicht aktiv)</p>

Antwort Telegram:

Zuerst schreiben...

Anmerkung	Byte	Struktur	Abkürzung	Beschreibung
	0	Telegramm Kopf	TADR	Telegrammadresse, durch den Schreibbefehl vorgegeben, Bereich 0..255, 0..0xFF
	1		SADR	Slave (Modul) Adresse
	2		CMD	Befehl: 15 = Schreiben mehrerer Parameter durch den Master
	3	Daten des SADR Modules	DAT.1H	Statuswort High Byte des Slave (Modul) Status
	4		DAT.1L	Low Byte des Slave (Modul) Status
	5		DAT.2H	VALUE1 = d1.07 von dem Modul mit der entsprechenden SADR
	6		DAT.2L	Aktueller Strom A
	7		DAT.3H	VALUE2 = d1.08 von dem Modul mit der entsprechenden SADR
	8		DAT.3L	Aktueller Strom B
....Zuletzt schreiben	9			Gesamte Telegrammlänge in Bytes

Erklärung des High and Low Byte des Slave-Modules (**Statuswort**):

Beschreibung	Abkürzung	Funktion des einzelnen Bits
High byte des Statuswortes	DAT.1H	<p>0x04 = HW_ENABLE, hardware enable ist aktiv 0x08 = ERROR, ein Fehler steht an 0x80 = BUS_DISABLE ist aktiv In diesem Modus nicht verfügbar, reserviert:</p> <p>0x01 = STAT_1 0x02 = STAT_2 0x10 = STAT_3 0x20 = STAT_4 0x40 = STAT_5</p>
Low byte des Statuswortes	DAT.1L	<p>Im Falle eines Fehlers, d.h. wenn das "ERROR" Bit gesetzt ist: Fehler Nummer, sonst: immer "0" In diesem Modus nicht verfügbar, reserviert:</p> <p>0x01 = Din_1 active 0x02 = Dout_1 active 0x04 = DKOMP_1 active 0x08 = DKOMP_11 active 0x10 = Din_2 active 0x20 = Dout_2 active 0x40 = DKOMP_2 active 0x80 = DKOMP_22 active</p>

4.5.3 Version: V22.xxy, gesteuert, zwei Ventils mit jeweils einem Magnet, Modus 2

Telegram Struktur Beispiel mit einem Slave (Modul, SNUM = 1),
Master schreibt mehrere Parameter (3 + 5·Modul Bytes)

Zuerst schreiben...

Anmerkung	Byte	Struktur	Abkürzung	Beschreibung
	0	Telegramm Kopf	TADR	Telegrammadresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
	1		SNUM	Anzahl der angeschlossenen Slaves (Module) (Maximal 5)
	2		CMD	Befehl: 15 = schreiben mehrerer Parameter durch den Master
	3	Modul Daten	DAT1.1	Steuer-Byte Byte für spezielle Funktionen
	4		DAT1.2H	VALUE1 = A1.01
	5		DAT1.2L	Sollwert Eingang (Profibus) für (Zweig 1)
	6		DAT1.3H	VALUE2 = A2.01
	7		DAT1.3L	Sollwert Eingang (Profibus) für (Zweig 2)
....Zuletzt schreiben	8			Gesamte Telegrammlänge in Bytes

Erklärung des Byte für spezielle Funktionen (**Steuer-Byte**)

Beschreibung	Abkürzung	Funktion des einzelnen Bits
Byte für spezielle Funktionen	DATx.1	<p>Kann nur gesetzt werden wenn HW_ENABLE aktiv ist. 0x80 = BUS_DISABLE (Freigabe via Profibus sperren) In diesem Modus nicht verfügbar, reserviert:</p> <p>0x01 = Din_1 0x02 = CNTRL_1 0x04 = TST_CMP 0x08 = CNTRL_2 0x10 = Din_2 0x20 = CNTRL_3 0x40 = CNTRL_4</p> <p>Das Steuer-Byte wird auf intern 0x00 (reset) gesetzt, wenn die Hardware-Freigabe inaktiv ist.</p>

Antwort Telegram:

Zuerst schreiben...

Anmerkung	Byte	Struktur	Abkürzung	Beschreibung
	0	Telegramm Kopf	TADR	Telegrammadresse, durch den Schreibbefehl vorgegeben, Bereich 0..255, 0..0xFF
	1		SADR	Slave (Modul) Adresse
	2		CMD	Befehl: 15 = Schreiben mehrerer Parameter durch den Master
	3	Modul Daten	DAT.1H	Statuswort High Byte des Slave (Modul) Status
	4		DAT.1L	Low Byte des Slave (Modul) Status
	5		DAT.2H	VALUE1 = d1.07 von dem Modul mit der entsprechenden SADR
	6		DAT.2L	Aktueller Strom A
	7		DAT.3H	VALUE2 = d1.08 von dem Modul mit der entsprechenden SADR
	8		DAT.3L	Aktueller Strom B
....Zuletzt schreiben	9			Gesamte Telegrammlänge in Bytes

Erklärung des High and Low Byte des Slave-Modules (**Statuswort**):

Beschreibung	Abkürzung	Funktion des einzelnen Bits
High byte des Statuswortes	DAT.1H	<p>0x04 = HW_ENABLE, Hardware Freigabe ist aktiv 0x08 = ERROR, ein Fehler steht an 0x80 = BUS_DISABLE ist aktiv In diesem Modus nicht verfügbar, reserviert:</p> <p>0x01 = STAT_1 0x02 = STAT_2 0x10 = STAT_3 0x20 = STAT_4 0x40 = STAT_5</p>
Low byte des Statuswortes	DAT.1L	<p>Im Falle eines Fehlers, d.h. wenn das "ERROR" Bit gesetzt ist: Fehler Nummer, sonst: immer "0" In diesem Modus nicht verfügbar, reserviert:</p> <p>0x01 = Din_1 active 0x02 = Dout_1 active 0x04 = DKOMP_1 active 0x08 = DKOMP_11 active 0x10 = Din_2 active 0x20 = Dout_2 active 0x40 = DKOMP_2 active 0x80 = DKOMP_22 active</p>

4.5.4 Version: V32.xxy, geregelt, Modus 3, 4

**Telegram Struktur Beispiel mit einem Slave (Modul, SNUM = 1),
Master schreibt mehrere Parameter (3 + 5·Modul Bytes)**

Zuerst schreiben...

Anmerkung	Byte	Struktur	Abkürzung	Beschreibung
	0	Telegramm Kopf	TADR	Telegrammadresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
	1		SNUM	Anzahl der angeschlossenen Slaves (Module) (Maximal 5)
	2		CMD	Befehl: 15 = schreiben mehrerer Parameter durch den Master
	3	Modul Daten	DAT1.1	Steuer-Byte Byte für spezielle Funktionen
	4		DAT1.2H	VALUE1 = A1.01
	5		DAT1.2L	Sollwert Eingang (Profibus) für (Zweig 1)
	6		DAT1.3H	VALUE2 = A1.02
	7		DAT1.3L	Istwert Eingang (Profibus) für (Zweig 1)
....Zuletzt schreiben	8			Gesamte Telegrammlänge in Bytes

Erklärung des Byte für spezielle Funktionen (**Steuer-Byte**)

Beschreibung	Abkürzung	Funktion des einzelnen Bits
Byte für spezielle Funktionen	DATx.1	<p>Kann nur gesetzt werden wenn HW_Enable aktiv. 0x01 = Din_1; Setzt das Signal Din_1 aktiv. 0x04 = TST_CMP; Fehler/Komparatormodus definiert setzen. 0x10 = Din_2; Setzt das Signal Din_2 aktiv. 0x80 = BUS_DISABLE (Freigabe via Profibus sperren) In diesem Modus nicht verfügbar, reserviert: 0x02 = CNTRL_1 0x08 = CNTRL_2 0x20 = CNTRL_3 0x40 = CNTRL_4</p> <p>Das Steuer-Byte wird auf intern 0x00 (reset) gesetzt, wenn die Hardware-Freigabe inaktiv ist.</p>

Antwort Telegram:

Zuerst schreiben...

Anmerkung	Byte	Struktur	Abkürzung	Beschreibung
	0	Telegramm Kopf	TADR	Telegrammadresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
	1		SADR	Anzahl der angeschlossenen Slaves (Module) (Maximal 5)
	2		CMD	Befehl: 15 = schreiben mehrerer Parameter durch den Master
	3	Modul Daten	DAT.1H	Statuswort High Byte des Slave (Modul) Status
	4		DAT.1L	Low Byte des Slave (Modul) Status
	5		DAT.2H	VALUE1 = d1.11 von dem Modul mit der entsprechenden SADR
	6		DAT.2L	Aktueller Istwert (Zweig 1)
	7		DAT.3H	VALUE2 = d1.10 von dem Modul mit der entsprechenden SADR
	8		DAT.3L	Sollwert (interer Wert) (Zweig 1)
....Zuletzt schreiben	9			Gesamte Telegrammlänge in Bytes

Erklärung des High and Low Byte des Slave-Modules (**Statuswort**):

Beschreibung	Abkürzung	Funktion des einzelnen Bits
High byte des Statuswortes	DAT.1H	<p>0x04 = HW_ENABLE, Hardware Freigabe ist aktiv 0x08 = ERROR, ein Fehler steht an 0x80 = BUS_DISABLE ist aktiv In diesem Modus nicht verfügbar, reserviert: 0x01 = STAT_1 0x02 = STAT_2 0x10 = STAT_3 0x20 = STAT_4 0x40 = STAT_5</p>
Low byte des Statuswortes	DAT.1L	<p>Im Falle eines Fehlers, d.h. wenn das "ERROR" Bit gesetzt ist: Fehler Nummer. Ansonsten: 0x01 = Din_1 ist aktiv 0x02 = Dout_1 ist aktiv 0x04 = DKOMP_1 ist aktiv 0x08 = DKOMP_11 ist aktiv 0x10 = Din_2 ist aktiv 0x20 = Dout_2 ist aktiv 0x40 = DKOMP_2 ist aktiv 0x80 = DKOMP_22 ist aktiv</p>

4.5.5 Version: V32.xxy, geregelt, Modus 6

**Telegram Struktur Beispiel mit einem Slave (Modul, SNUM = 1),
Master schreibt mehrere Parameter (3 + 5·Modul Bytes)**

Zuerst schreiben...

Anmerkung	Byte	Struktur	Abkürzung	Beschreibung
	0	Telegramm Kopf	TADR	Telegrammadresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
	1		SNUM	Anzahl der angeschlossenen Slaves (Module) (Maximal 5)
	2		CMD	Befehl: 15 = schreiben mehrerer Parameter durch den Master
	3	Modul Daten	DAT1.1	Steuer-Byte Byte für spezielle Funktionen
	4		DAT1.2H	VALUE1 = A1.01
	5		DAT1.2L	Sollwert Eingang (Profibus) für (Zweig 1)
	6		DAT1.3H	VALUE2 = A2.02
	7		DAT1.3L	Istwert Eingang (Profibus) für (Zweig 2)
....Zuletzt schreiben	8			Gesamte Telegrammlänge in Bytes

Erklärung des Byte für spezielle Funktionen (**Steuer-Byte**)

Beschreibung	Abkürzung	Funktion des einzelnen Bits
Byte für spezielle Funktionen	DATx.1	<p>Kann nur gesetzt werden wenn HW_Enable aktiv. 0x01 = Din_1; Setzt das Signal Din_1 aktiv. 0x04 = TST_CMP; Fehler/Komparatormodus definiert setzen. 0x10 = Din_2; Setzt das Signal Din_2 aktiv. 0x80 = BUS_DISABLE (Freigabe via Profibus sperren) In diesem Modus nicht verfügbar, reserviert: 0x02 = CNTRL_1 0x08 = CNTRL_2 0x20 = CNTRL_3 0x40 = CNTRL_4</p> <p>Das Steuer-Byte wird auf intern 0x00 (reset) gesetzt, wenn die Hardware-Freigabe inaktiv ist.</p>

Antwort Telegram:

Zuerst schreiben...

Anmerkung	Byte	Struktur	Abkürzung	Beschreibung
	0	Telegramm Kopf	TADR	Telegrammadresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
	1		SADR	Anzahl der angeschlossenen Slaves (Module) (Maximal 5)
	2		CMD	Befehl: 15 = schreiben mehrerer Parameter durch den Master
	3	Modul Daten	DAT.1H	Statuswort High Byte des Slave (Modul) Status
	4		DAT.1L	Low Byte des Slave (Modul) Status
	5		DAT.2H	VALUE1 = d1.11 von dem Modul mit der entsprechenden SADR
	6		DAT.2L	Aktueller Istwert (Zweig 1)
	7		DAT.3H	VALUE2 = d2.11 von dem Modul mit der entsprechenden SADR
	8		DAT.3L	Aktueller Istwert (Zweig 2)
....Zuletzt schreiben	9			Gesamte Telegrammlänge in Bytes

Erklärung des High and Low Byte des Slave-Modules (**Statuswort**):

Beschreibung	Abkürzung	Funktion des einzelnen Bits
High byte des Statuswortes	DAT.1H	<p>0x04 = HW_ENABLE, Hardware Freigabe ist aktiv 0x08 = ERROR, ein Fehler steht an 0x80 = BUS_DISABLE ist aktiv In diesem Modus nicht verfügbar, reserviert: 0x01 = STAT_1 0x02 = STAT_2 0x10 = STAT_3 0x20 = STAT_4 0x40 = STAT_5</p>
Low byte des Statuswortes	DAT.1L	<p>Im Falle eines Fehlers, d.h. wenn das "ERROR" Bit gesetzt ist: Fehler Nummer. Ansonsten: 0x01 = Din_1 ist aktiv 0x02 = Dout_1 ist aktiv 0x04 = DKOMP_1 ist aktiv 0x08 = DKOMP_11 ist aktiv 0x10 = Din_2 ist aktiv 0x20 = Dout_2 ist aktiv 0x40 = DKOMP_2 ist aktiv 0x80 = DKOMP_22 ist aktiv</p>

4.5.6 Version: V32.xxy, geregelt, Modus 8

**Telegram Struktur Beispiel mit einem Slave (Modul, SNUM = 1),
Master schreibt mehrere Parameter (3 + 5·Modul Bytes)**

Zuerst schreiben...

Anmerkung	Byte	Struktur	Abkürzung	Beschreibung
	0	Telegramm Kopf	TADR	Telegrammadresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
	1		SNUM	Anzahl der angeschlossenen Slaves (Module) (Maximal 5)
	2		CMD	Befehl: 15 = schreiben mehrerer Parameter durch den Master
	3	Modul Daten	DAT1.1	Steuer-Byte Byte für spezielle Funktionen
	4		DAT1.2H	VALUE1 = A1.01
	5		DAT1.2L	Sollwert Eingang (Profibus) für (Zweig 1)
	6		DAT1.3H	VALUE2 = A2.01
	7		DAT1.3L	Sollwert Eingang (Profibus) für (Zweig2)
....Zuletzt schreiben	8			Gesamte Telegrammlänge in Bytes

Erklärung des Byte für spezielle Funktionen (**Steuer-Byte**)

Beschreibung	Abkürzung	Funktion des einzelnen Bits
Byte für spezielle Funktionen	DATx.1	<p>Kann nur gesetzt werden wenn HW_Enable aktiv. 0x01 = Din_1; Setzt das Signal Din_1 aktiv. 0x04 = TST_CMP; Fehler/Komparatormodus definiert setzen. 0x10 = Din_2; Setzt das Signal Din_2 aktiv. 0x80 = BUS_DISABLE (Freigabe via Profibus sperren) In diesem Modus nicht verfügbar, reserviert: 0x02 = CNTRL_1 0x08 = CNTRL_2 0x20 = CNTRL_3 0x40 = CNTRL_4</p> <p>Das Steuer-Byte wird auf intern 0x00 (reset) gesetzt, wenn die Hardware-Freigabe inaktiv ist.</p>

Antwort Telegram:

Zuerst schreiben...

Anmerkung	Byte	Struktur	Abkürzung	Beschreibung
	0	Telegramm Kopf	TADR	Telegrammadresse, durch den Anwender vorzugeben, Bereich 0..255, 0..0xFF
	1		SADR	Anzahl der angeschlossenen Slaves (Module) (Maximal 5)
	2		CMD	Befehl: 15 = schreiben mehrerer Parameter durch den Master
	3	Modul Daten	DAT.1H	Statuswort High Byte des Slave (Modul) Status
	4		DAT.1L	Low Byte des Slave (Modul) Status
	5		DAT.2H	VALUE1 = d1.11 von dem Modul mit der entsprechenden SADR
	6		DAT.2L	Aktueller Istwert (Zweig 1)
	7		DAT.3H	VALUE2 = d2.11 von dem Modul mit der entsprechenden SADR
	8		DAT.3L	Aktueller Istwert (Zweig 2)
....Zuletzt schreiben	9			Gesamte Telegrammlänge in Bytes

Erklärung des High and Low Byte des Slave-Modules (**Statuswort**):

Beschreibung	Abkürzung	Funktion des einzelnen Bits
High byte des Statuswortes	DAT.1H	<p>0x04 = HW_ENABLE, Hardware Freigabe ist aktiv 0x08 = ERROR, ein Fehler steht an 0x80 = BUS_DISABLE ist aktiv In diesem Modus nicht verfügbar, reserviert: 0x01 = STAT_1 0x02 = STAT_2 0x10 = STAT_3 0x20 = STAT_4 0x40 = STAT_5</p>
Low byte des Statuswortes	DAT.1L	<p>Im Falle eines Fehlers, d.h. wenn das "ERROR" Bit gesetzt ist: Fehler Nummer. Ansonsten: 0x01 = Din_1 ist aktiv 0x02 = Dout_1 ist aktiv 0x04 = DKOMP_1 ist aktiv 0x08 = DKOMP_11 ist aktiv 0x10 = Din_2 ist aktiv 0x20 = Dout_2 ist aktiv 0x40 = DKOMP_2 ist aktiv 0x80 = DKOMP_22 ist aktiv</p>

5 Siemens S7 Implementierung von “Multiple-Slave” (5 DMA Module)

5.1 Hardware Konfiguration

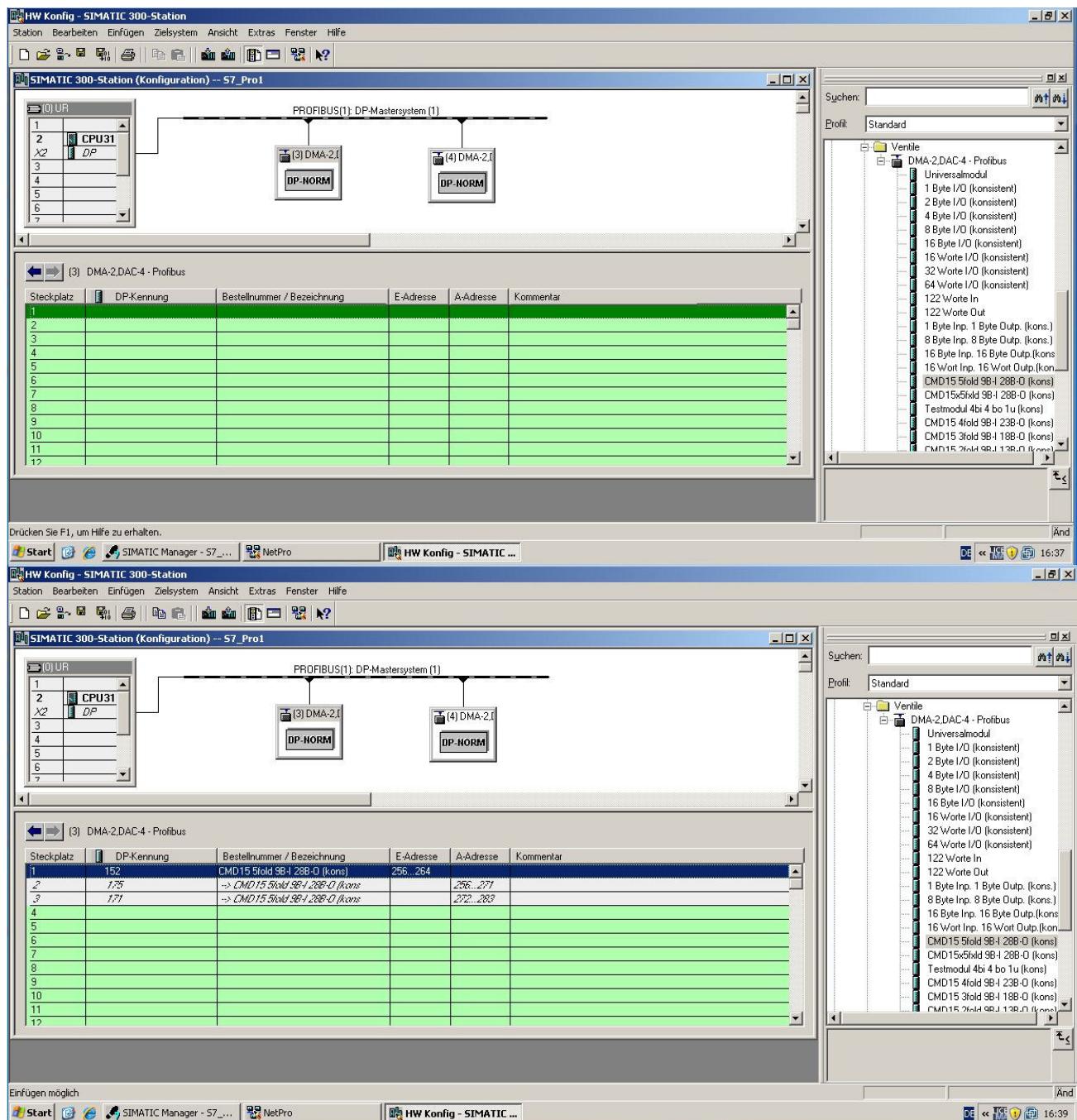
Hardware Konfiguration wird mit dem Programm “HW KONFIG” durchgeführt. Dort findet man den DMA Profibus-Node.

Die richtige Anzahl der Bytes für den Aus- und Eingangsbuffer festlegen, abhängig von der Anzahl der verwendeten Slaves (Module). In diesem Fall mit 5 Slaves (Module) ist der Ausgangs-Buffer auf mindestens 28 Byte und der Eingangs-Buffer auf mindestens 9 Bytes festgelegt.

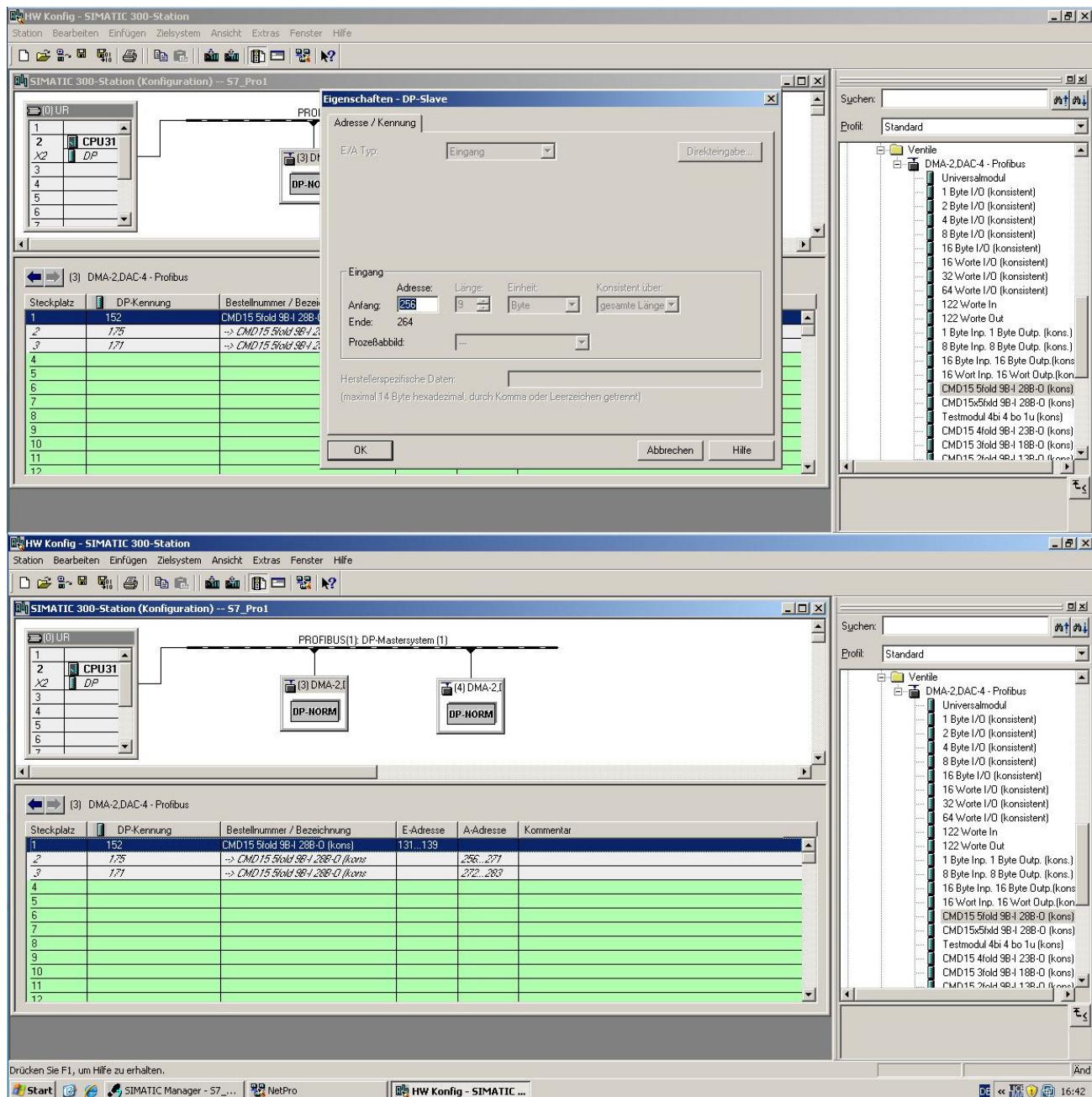
Ventile → “DMA-2 – Profibus”. Im Beispiel “CMD15 5fold 9B-I 28B-O” Profil auswählen.

Anwahl der richtigen Profibus Adresse. In diesem Beispiel: #03 = 0x03.

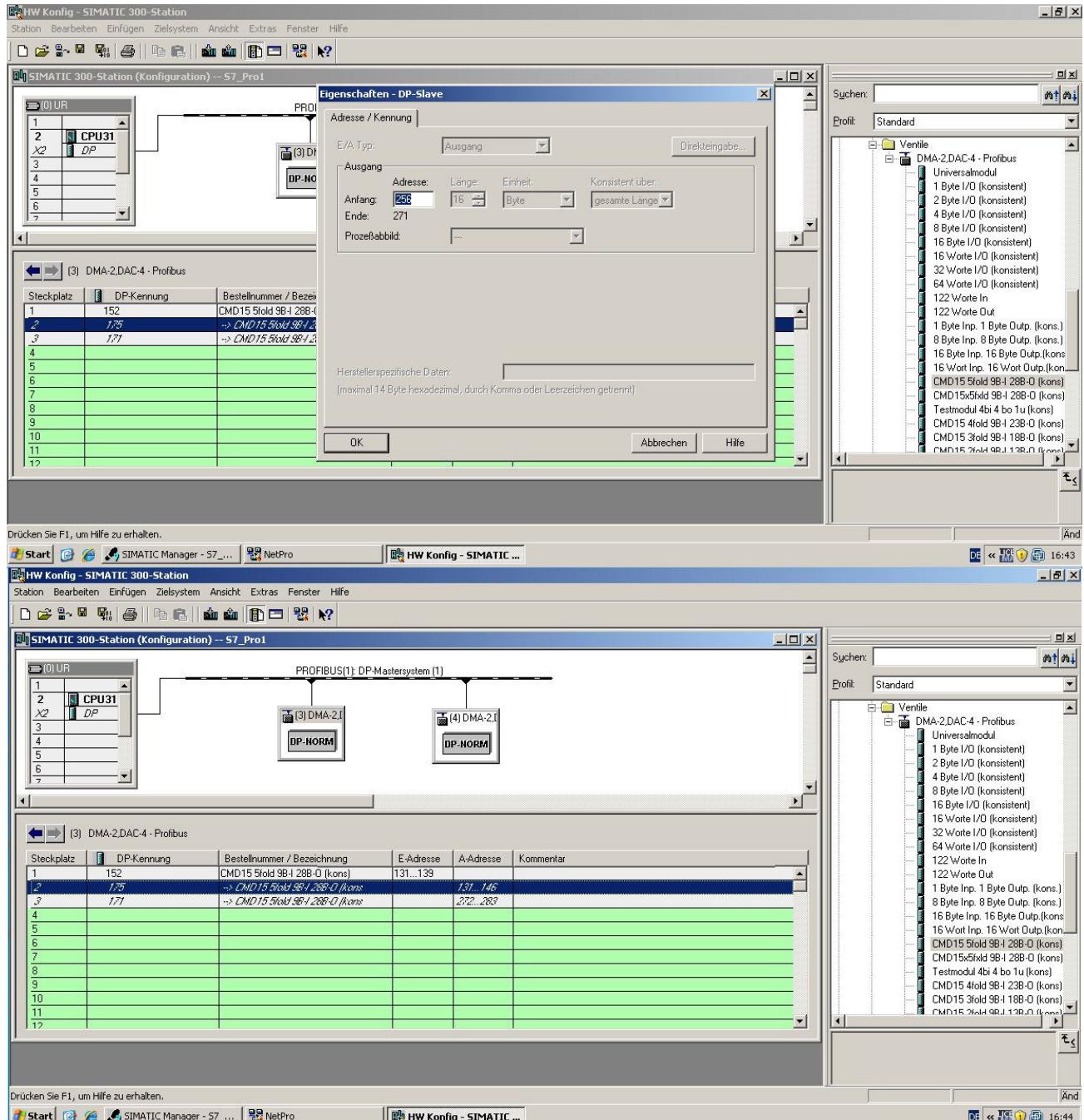
Einstellen der Profibusadresse an der Vorderseite des Profibus-Node (DIP-Schalter)



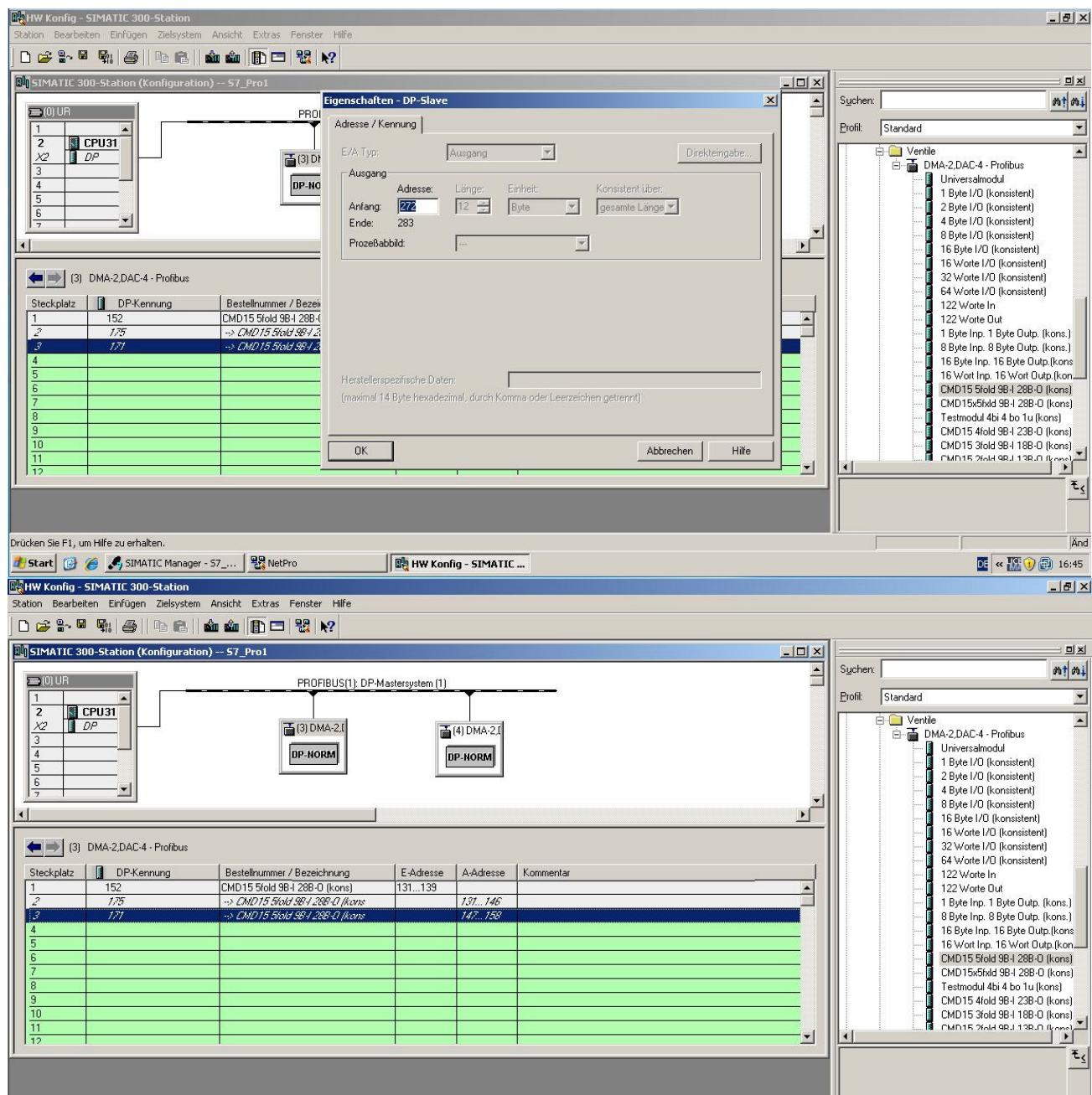
9 Byte Eingangslänge anpassen:



die ersten 16 Byte von insgesamt 28 Byte Ausgangslänge anpassen:



die restlichen 12 byte von 28 Byte Ausgangslänge anpassen



5.1.1 Hardware Konfiguration bei Geräten ohne CMD15-Module in der GSD-Datei

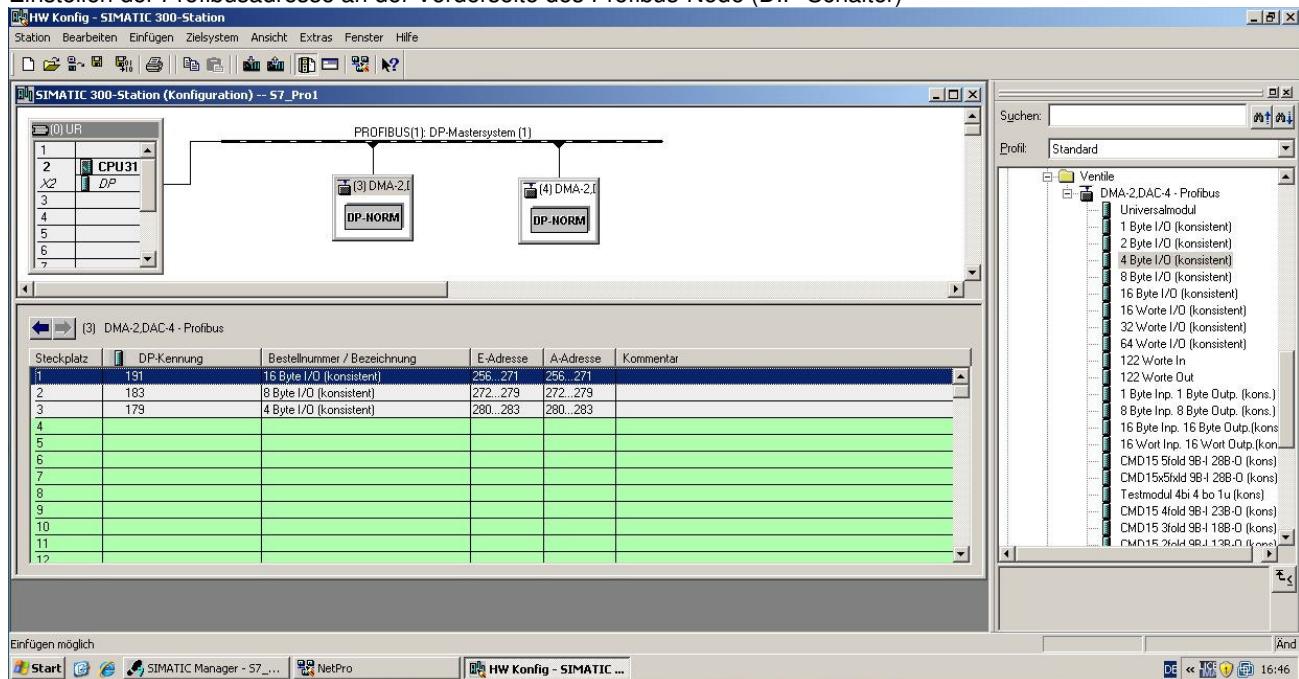
Hardware Konfiguration wird mit dem Programm "HW KONFIG" durchgeführt, Dort findet man den DMA Profibus-Node.

Die richtige Anzahl der Bytes für den Aus- und Eingangsbuffer festlegen, abhängig von der Anzahl der verwendeten Slaves (Module). In diesem Fall mit 5 Slaves (Module) ist der Ausgangs-Buffer auf mindestens 28 Byte und der Eingangs-Buffer auf mindestens 9 Bytes festgelegt.

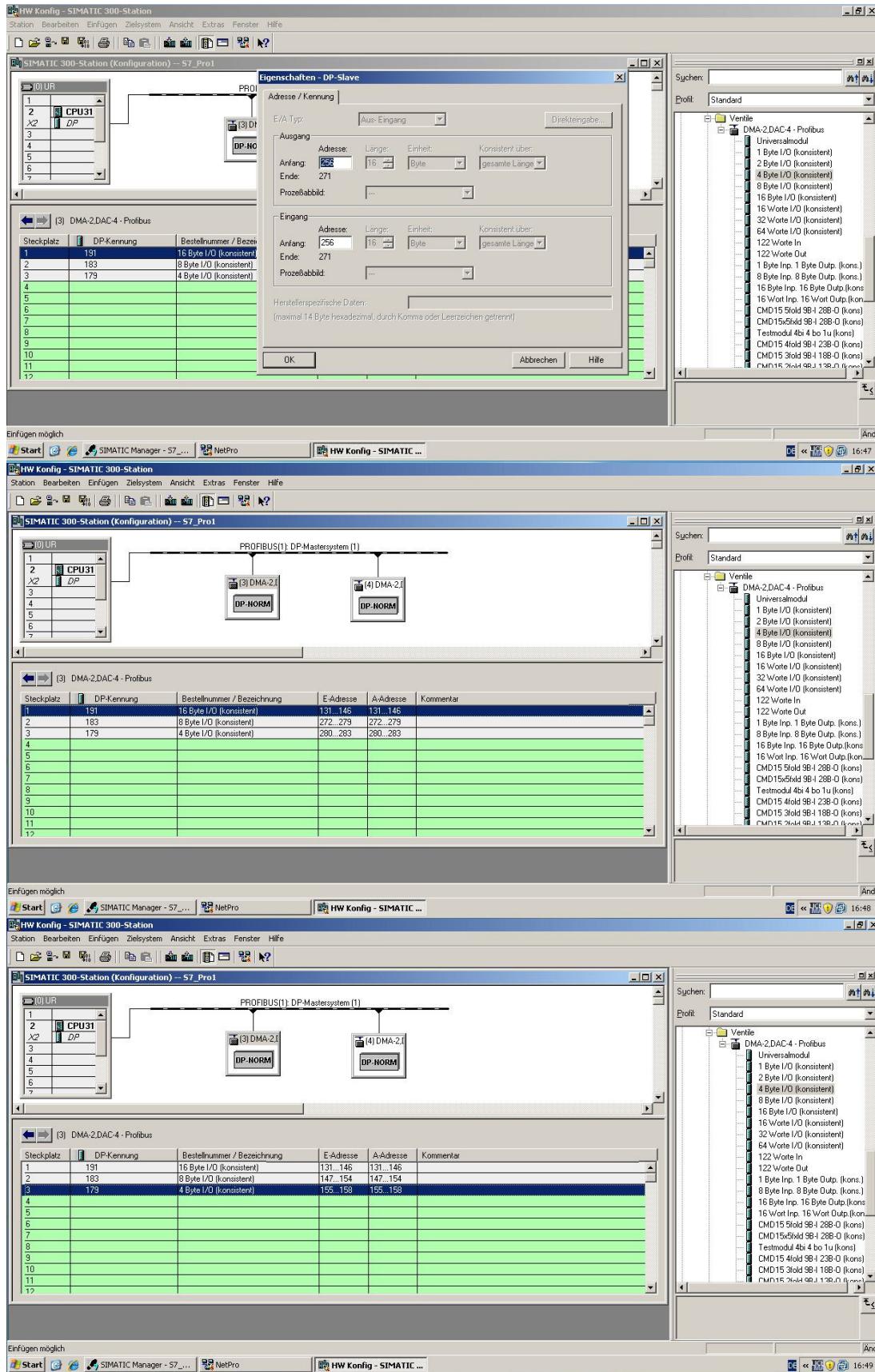
Ventile → "DMA-2 – Profibus". Zum Beispiel folgende Profile auswählen:
"16 Byte I/O (konsistent)" UND "8 Byte I/O (konsistent) UND "4 Byte I/O (konsistent)

Anwahl der richtigen Profibus Adresse. In diesem Beispiel: #03 = 0x03.

Einstellen der Profibusadresse an der Vorderseite des Profibus-Node (DIP-Schalter)

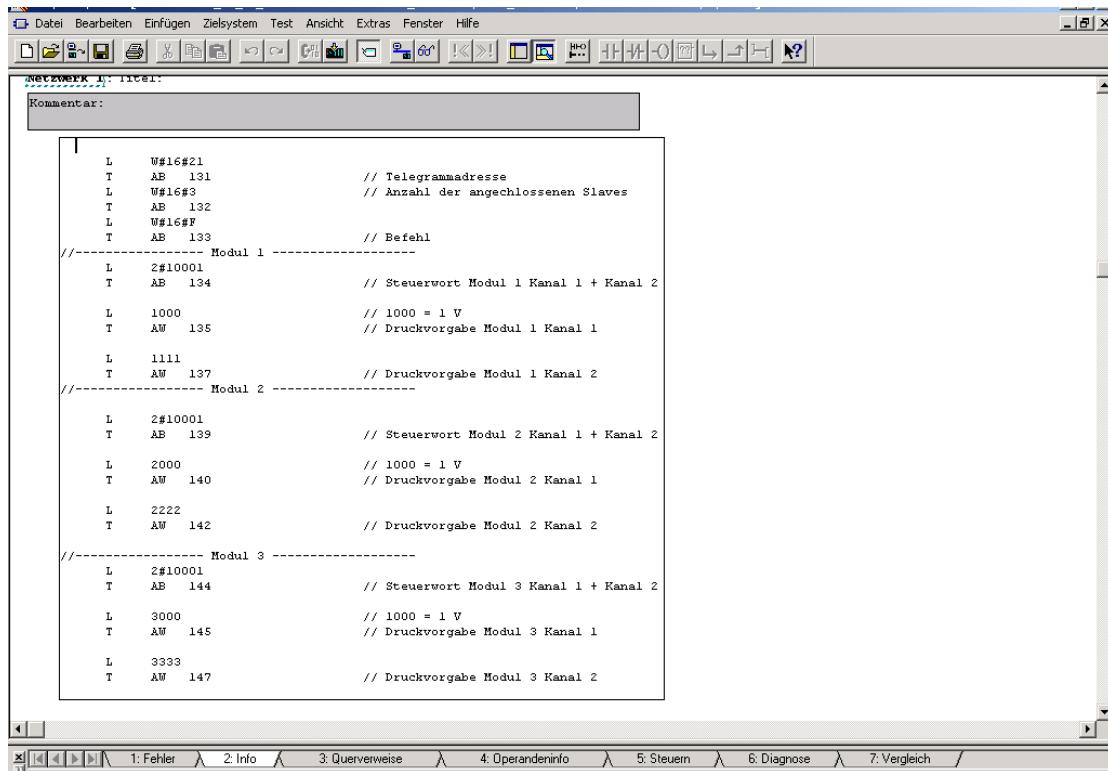


E/A Adressen anpassen:



5.2 CMD = 15. Protokoll Konfiguration

Ausgangs-Buffer definieren:



```

L  U#16#21
T  AB 131          // Telegrammadresse
L  U#16#3          // Anzahl der angeschlossenen Slaves
T  AB 132
L  U#16#F
T  AB 133          // Befehl
//----- Modul 1 -----
L  2#10001
T  AB 134          // Steuerwort Modul 1 Kanal 1 + Kanal 2

L  1000            // 1000 = 1 V
T  AW 135          // Druckvorgabe Modul 1 Kanal 1

L  1111
T  AW 137          // Druckvorgabe Modul 1 Kanal 2
//----- Modul 2 -----
L  2#10001
T  AB 139          // Steuerwort Modul 2 Kanal 1 + Kanal 2

L  2000            // 1000 = 1 V
T  AW 140          // Druckvorgabe Modul 2 Kanal 1

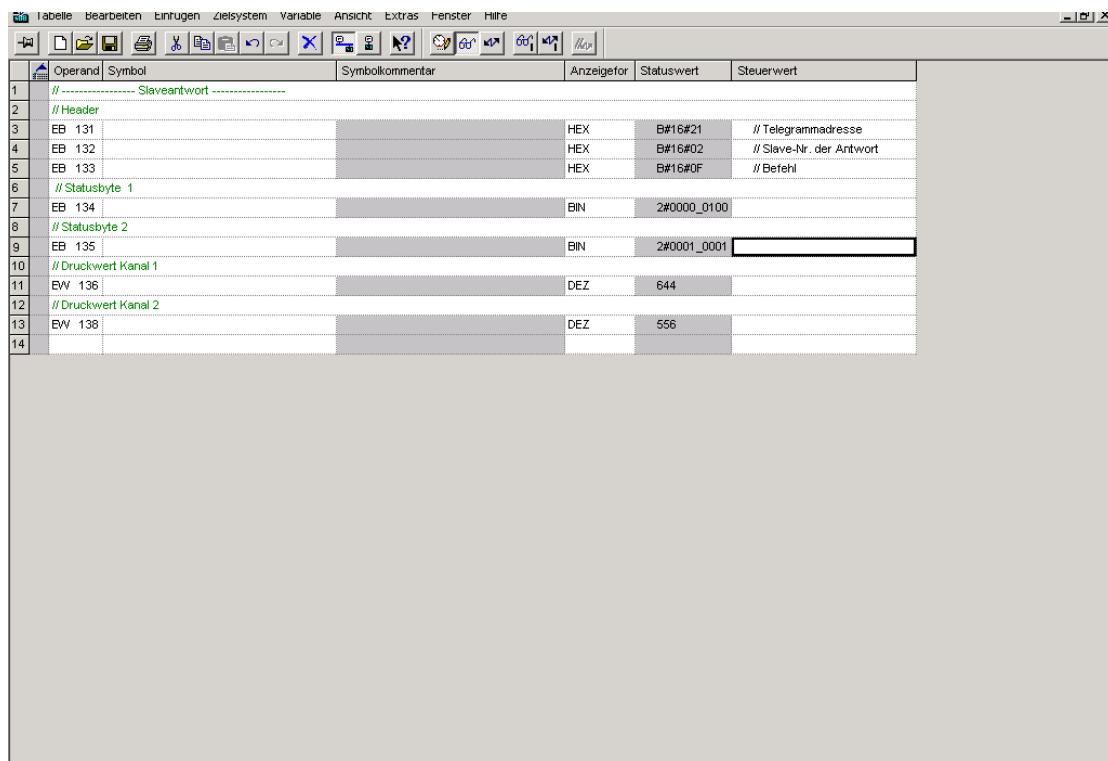
L  2222
T  AW 142          // Druckvorgabe Modul 2 Kanal 2
//----- Modul 3 -----
L  2#10001
T  AB 144          // Steuerwort Modul 3 Kanal 1 + Kanal 2

L  3000            // 1000 = 1 V
T  AW 145          // Druckvorgabe Modul 3 Kanal 1

L  3333
T  AW 147          // Druckvorgabe Modul 3 Kanal 2

```

Nun kann man den Eingangs-Buffer prüfen. In diesem Fall hat Slave (Modul) „2“ die Antwort geschickt. Man kann die empfangene Slave (Modul) Adresse am Eingangs-byte „EB 132“ sehen.



	Operand	Symbol	Symbolkommentar	Anzeigefor	Statuswert	Steuerwert
----- Slaveantwort -----						
2	// Header					
3	EB 131			HEX	B#16#21	// Telegrammadresse
4	EB 132			HEX	B#16#02	// Slave-Nr. der Antwort
5	EB 133			HEX	B#16#0F	// Befehl
6	// Statusbyte 1					
7	EB 134			BIN	2#0000_0100	
8	// Statusbyte 2					
9	EB 135			BIN	2#0001_0001	
10	// Druckwert Kanal 1					
11	EW 136			DEZ	644	
12	// Druckwert Kanal 2					
13	EW 138			DEZ	556	
14						

6 Konfiguration für das SIEMENS TIA portal

Zwei Funktionsabusteine für das Siemens TIA Portal sind verfügbar und können von der HCS Webseite heruntergeladen werden. Welches der Anwender benötigt hängt von der verwendeten CPU ab.

- Siemens CPU Serie S7-300,400 PB_DMA_2_CLASSIC_DB(Vxx).scl
- Siemens CPU Serie S7-1200,1500 PB_DMA_2_DB.scl(Vxx).scl

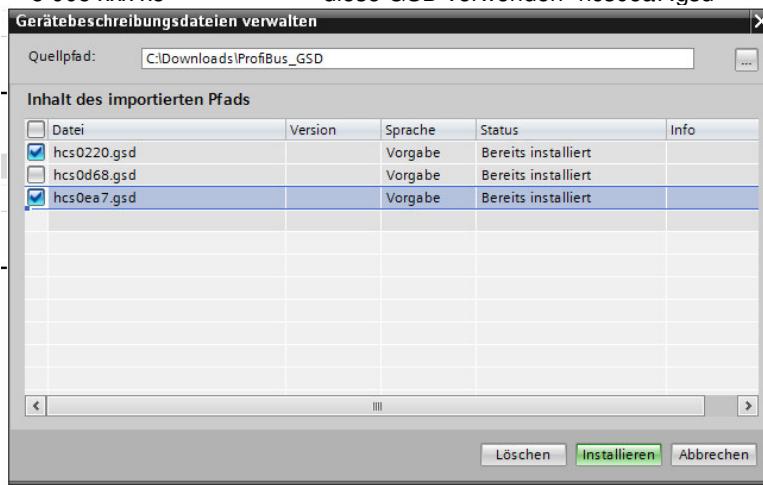
Wenn der DMA Knoten aus DMA's besteht, welche in unterschiedlichen Moden arbeiten, so sind nur die Bit's im Steuer-Byte und Statuswort entsprechend dem angesprochenen DMA, bzw. dessen Betriebsmode, zu beachten).

6.1 Einrichten der Netzwerkkonfiguration für das DMA-Modul

6.1.1 Laden Sie die GSD-Datei herunter und installieren Sie diese

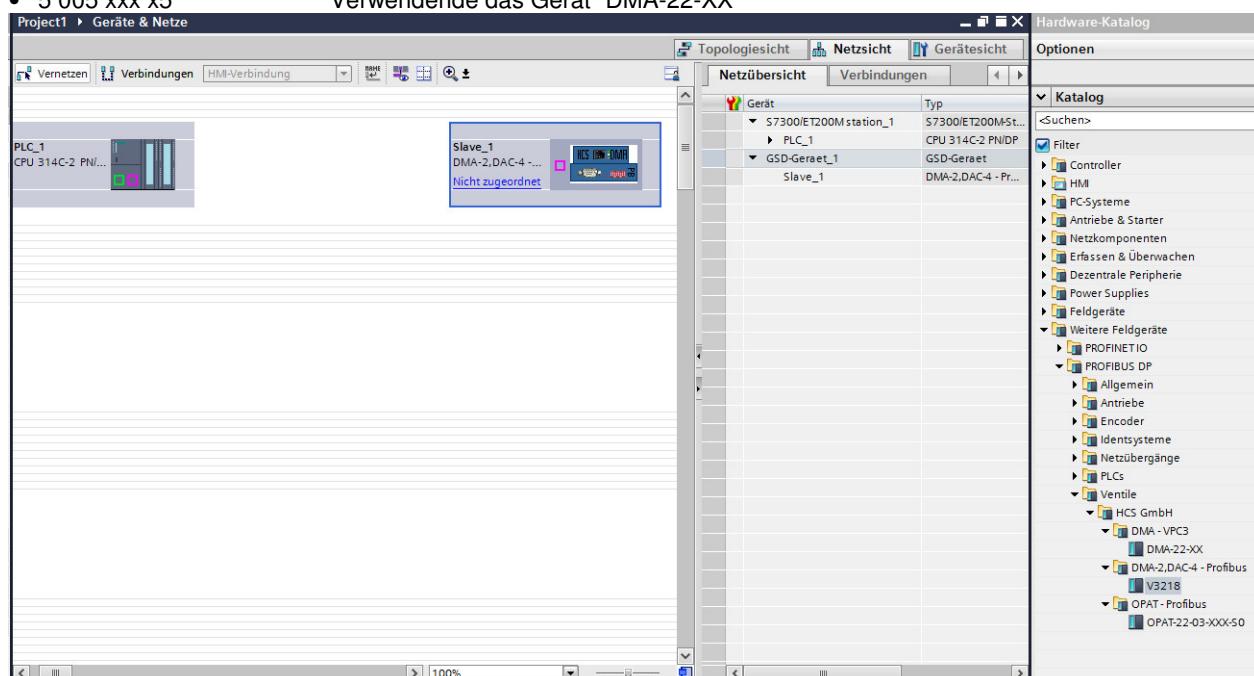
Bitte wählen Sie die richtige GSD-Datei für das DMA-Modul aus. Die letzten beiden Ziffern der Produktnummer sind relevant.

- 5 xxx xxx x1 to x4 diese GSD verwenden "hcs0220.gsd"
- 5 005 xxx x5 diese GSD verwenden "hcs0ea7.gsd"

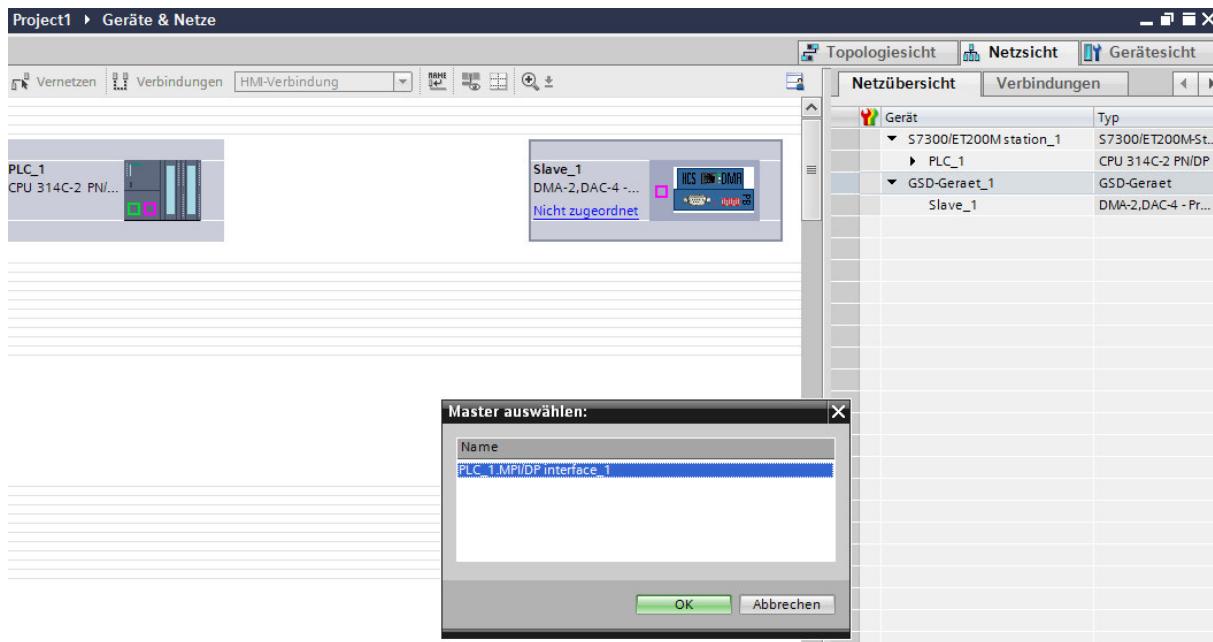


6.1.2 Füge ein neues Gerät dem Netzwerk hinzu

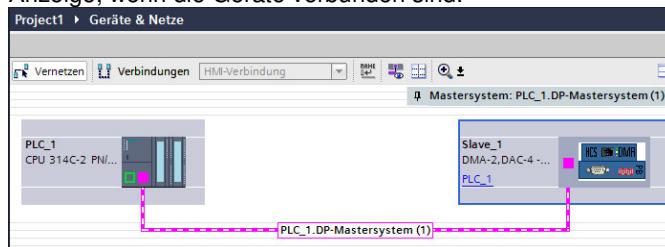
- 5 xxx xxx x1 to x4 Verwendende das Gerät "V3218"
- 5 005 xxx x5 Verwendende das Gerät "DMA-22-XX"



6.1.3 Weisen Sie das Slave_1-Gerät dem PLC_1 Profibus-Master zu

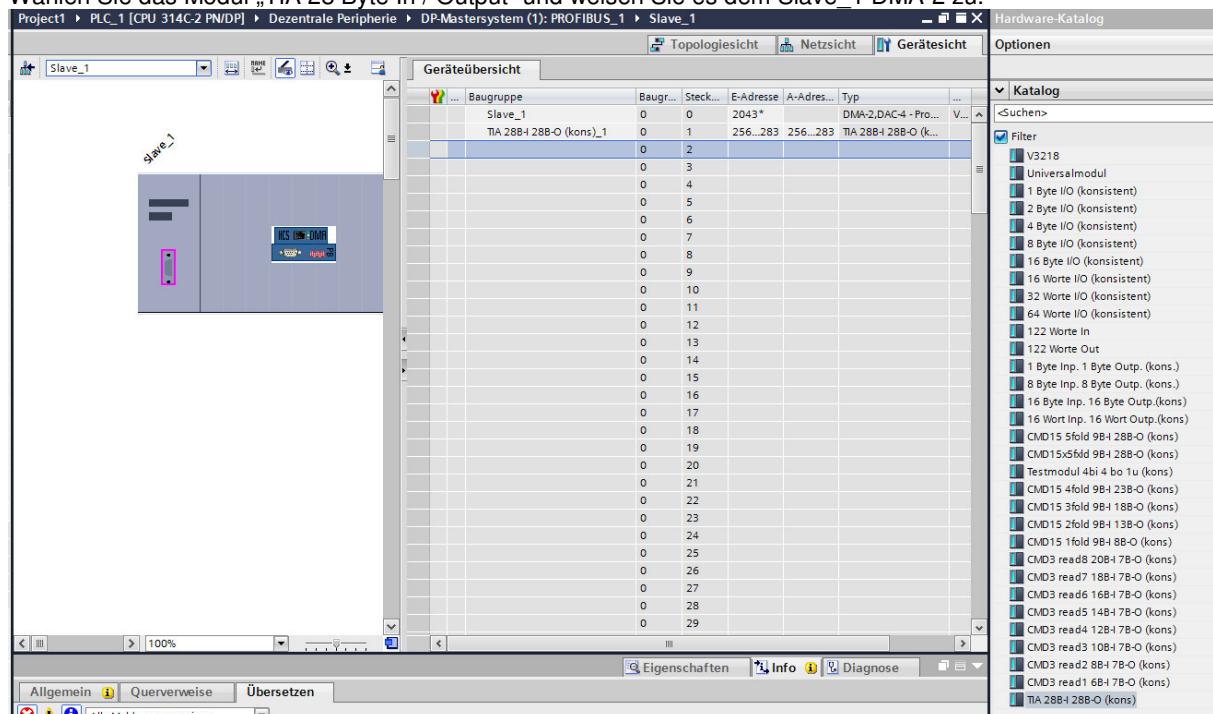


Anzeige, wenn die Geräte verbunden sind.



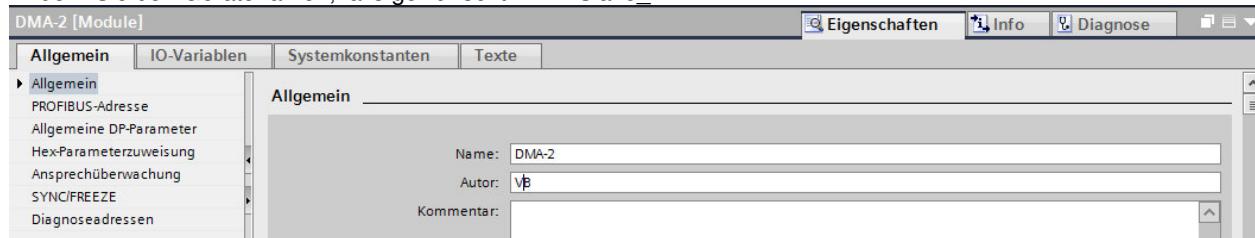
6.1.4 Definieren Sie das Kommunikationsmodul aus der GSD

Wählen Sie das Modul „TIA 28 Byte In / Output“ und weisen Sie es dem Slave_1 DMA-2 zu.



6.1.5 Legen Sie die Profibus Adresse des DMA Modules fest.

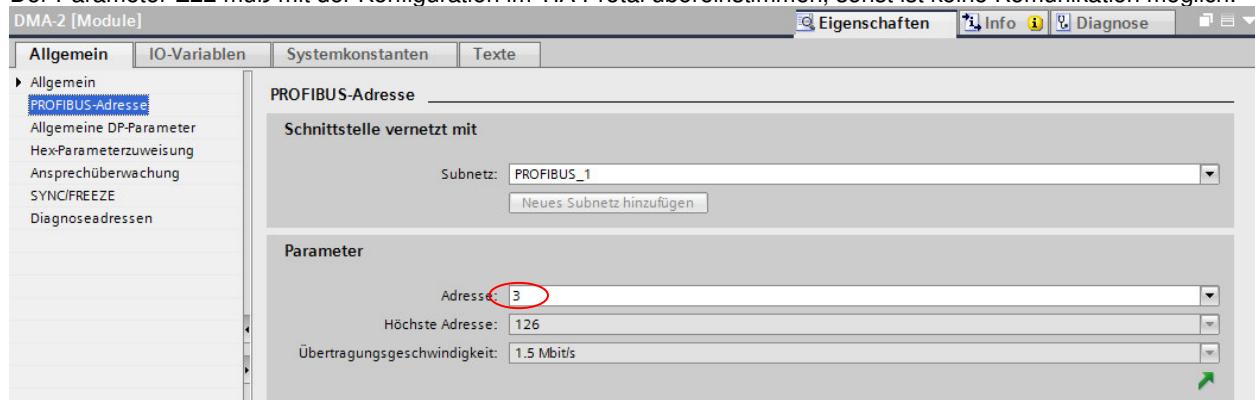
Ändern Sie den Gerätenamen, falls gewünscht. Z. B. Slave_1 → DMA-2



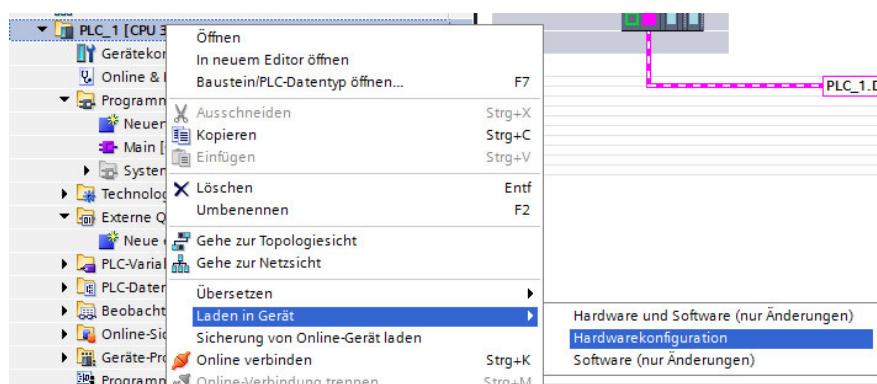
Mit dem HCSTool kann der Parameter E 22, die Profibus Slave Adresse, festgelegt werden. Dazu muss das DMA-2 Modul, welches direkt neben dem Profibus Konten sitzt, entsprechend parametriert werden.

Parameter	Name	Wert	Einheit	Beschreibung
... Sollwert/Rampe1	E 00	1		Betriebsmode
... Controller1	E 02	off		Push-Pull Betrieb
... Erweitert	E 03	0,800	A	Magnetanwahl
	E 04	500		P-Anteil Stromregler erregen
	E 05	500		I-Anteil Stromregler erregen
	E 06	1300		P-Anteil Stromregler entregen
	E 07	500		I-Anteil Stromregler entregen
	E 08	0		Rampenart
	E 09	0,000	s	Zeitverzögerung Freigabe
	E 10	1,00		Magnetstromanpassung
	E 11	0,000	V	Vorstrom A
	E 12	0,000	V	Vorstrom B
	E 13	0,000	V	Dither Amplitude
	E 14	0	Hz	Dither Frequenz
	E 15	0		Signaldefinition Sollwert S1.06
	E 17	2		Sollwert Aktivierungsmodus
	E 22	3		BUS Slave Adresse
	E 23	0,000	s	BUS timeout, bei 0s ist die Funktion ausgeschaltet

Der Parameter E22 muß mit der Konfiguration im TIA Protal übereinstimmen, sonst ist keine Kommunikation möglich.



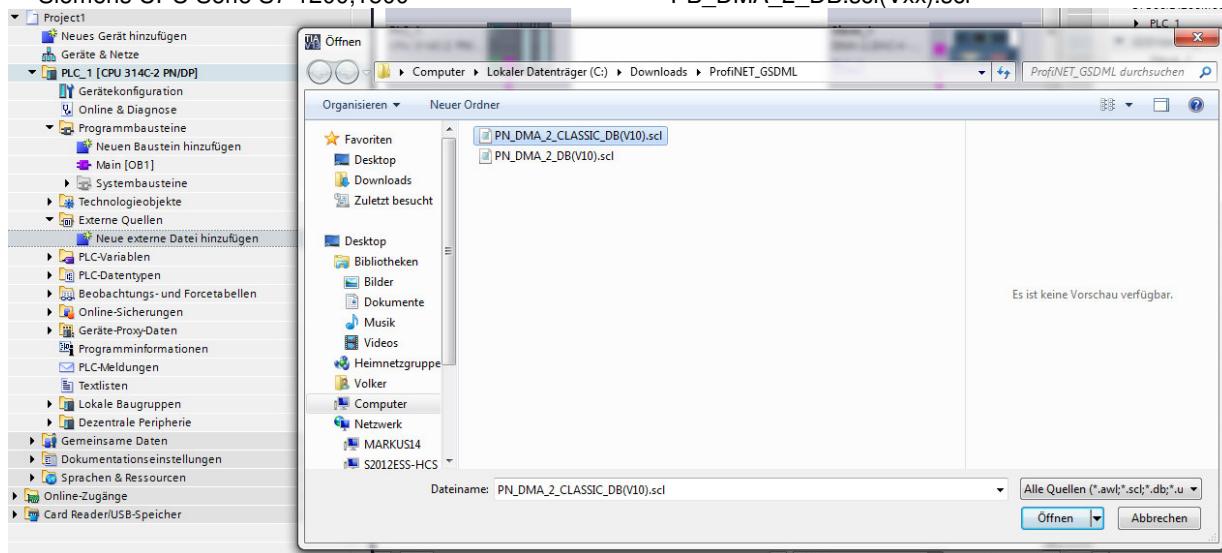
6.1.6 Laden Sie die neue Hardwarekonfiguration in die SPS



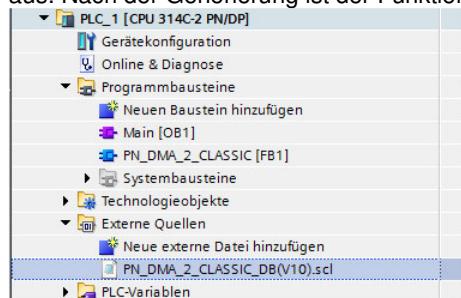
6.1.7 Installieren Sie den Funktionsbaustein

Neue externe Datei hinzufügen, wählen Sie den Funktionsbaustein entsprechend Ihrer verwendeten CPU-Serie aus:

- Siemens CPU Serie S7-300,400 PB_DMA_2_CLASSIC_DB(Vxx).scl
- Siemens CPU Serie S7-1200,1500 PB_DMA_2_DB.scl(Vxx).scl

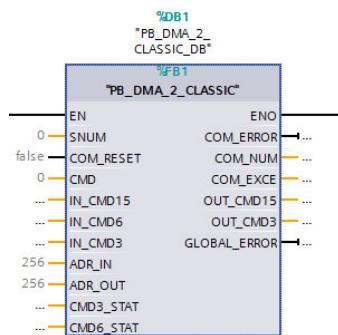


Klicken Sie mit der rechten Maustaste auf die ausgewählte Datei und wählen Sie "Bausteine aus der Quelle generieren" aus. Nach der Generierung ist der Funktionsbaustein verfügbar.



6.2 Erläuterung der Funktionen des Funkfionsblocks

In diesem Kapitel wird beschrieben, wie der Funktionsbaustein in den verschiedenen Befehlsmodi verwendet wird. Aufbau und Funktion für CMD 3, CMD 6 sind in jedem Arbeitsmodus des DMA-2 gleich. Nur der CMD 15 unterscheidet sich je nach DMA-2-Betriebsmodus.



Die Bus-Slave-Adresse SNUM jedes angeschlossenen DMA-2 wird durch den Parameter E 22 festgelegt.



COM_ERROR wird aus dem Telegrammkopf abgerufen, der vom Profibus-Knoten an die SPS übertragen wird. Wenn dieser Fehler ansteht, können die Slave-Adressquelle und die Ursache durch Analysieren des COM_EXCE-Codes ermittelt werden.

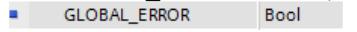
(Siehe [3.3 CMD = 3, Fehler vom Slave \(4 Bytes\)](#) or [3.6 CMD = 6, Fehler vom Slave \(4 Bytes\)](#))



Ein Kommunikationsfehler kann durch den Eingang #COM_RESET zurückgesetzt werden.



Ein GLOBAL_ERROR tritt auf, wenn ein Kommunikationsfehler und / oder ein interner DMA-2-Fehler ansteht.



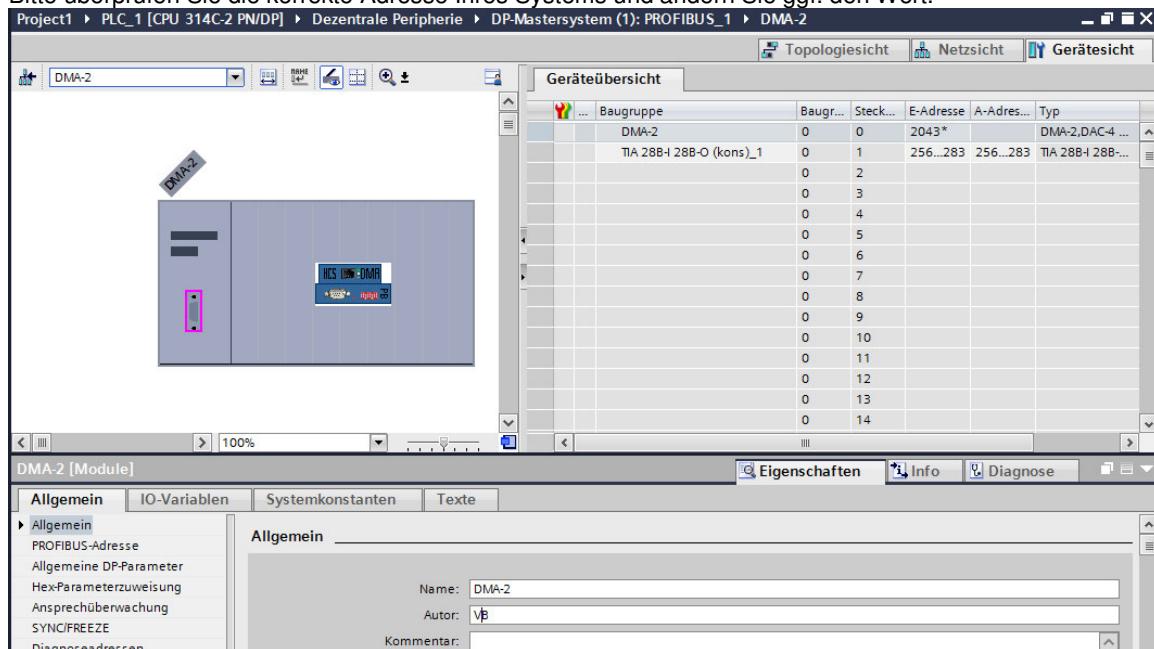
6.2.1 Klassik CPU 300,400: Festlegung der Datenpuffer Adresse

Im Unterschied zur CPU 1200, 1500 muss hier die E-Adresse und A-Adresse der Variablen #ADR_IN und #ADR_OUT zugewiesen werden.

Die Datenpufferadresse #ADR_IN, #ADR_OUT ist auf 256 vordefiniert.

256 — **ADR_IN**
256 — **ADR_OUT**

Bitte überprüfen Sie die korrekte Adresse Ihres Systems und ändern Sie ggf. den Wert.



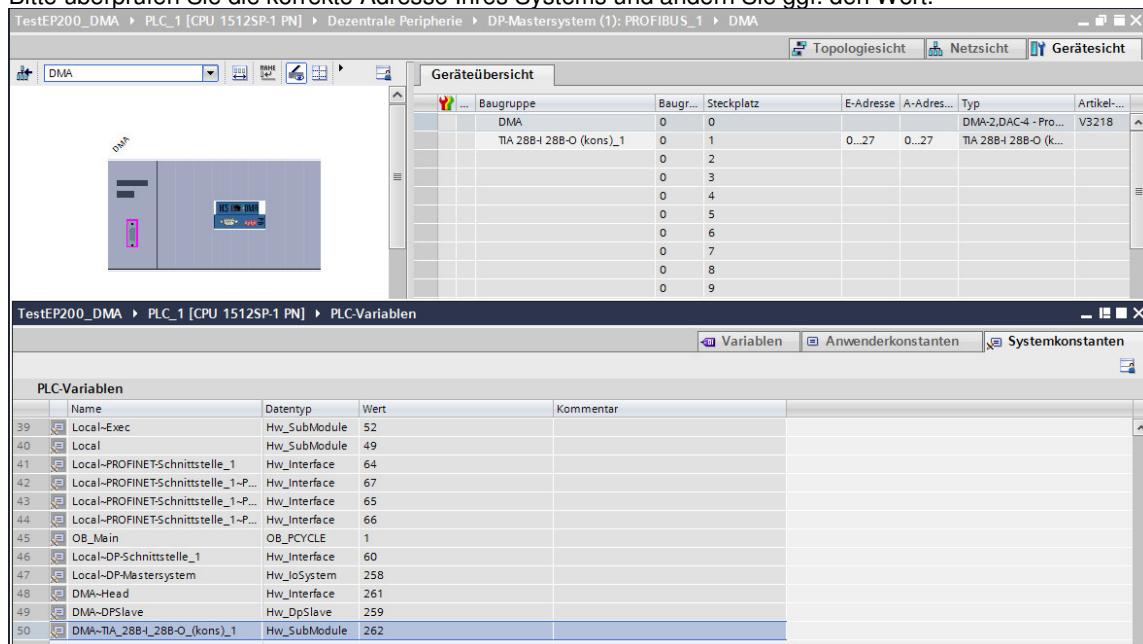
6.2.2 CPU 1200,1500: Festlegung der Datenpuffer Adresse anhand der Hardwarekennung

Im Unterschied zur CPU 300, 400 muss hier die Hardwarekennung für das GSD I/O Module der Variablen #DEV_ID zugewiesen werden.

Die Datenpufferadresse #ADR_IN, #ADR_OUT ist auf 261 vordefiniert.

DEV_ID HW_IO 44.0 261

Bitte überprüfen Sie die korrekte Adresse Ihres Systems und ändern Sie ggf. den Wert.



6.2.3 CMD 3, Lesen von Parametern aus einem DMA Modul

Jedes an den Profibus-Knoten angeschlossene DMA-2-Modul verfügt über eine eigene SNUM (Slave-Nummer). Zusammen mit dieser Nummer ist es möglich, einen Parametersatz dieses spezifischen DMA-2 zu lesen. Die Parameter-IDs werden in der Tabelle angezeigt, siehe [7 Parameterliste](#).

Ein Lesezugriff erfolgt wie folgt:

- Definieren Sie die Slave-Nummer SNUM des DMA-2, von dem die Parameter gelesen werden sollen.

■ SNUM	Int
--------	-----

Z.B. für Slave 1 #SNUM := 1;

- Den Befehl CMD für Lesen: # CMD := 3;

■ CMD	Int
-------	-----

- Schreiben Sie die Startadresse der ersten Parameter-ID, die von der Struktur gelesen werden soll.

■ IN_CMD3	Struct
■ PAR_ID	Int

Z.B. für Parameter 22, S1.01 #IN_CMD3.PAR_ID := 22;

- Legen Sie die Anzahl der zu lesenden Parameter fest (maximal 8)

■ IN_CMD3	Struct
■ PAR_ID	Int
■ PAR_N	Int

Z.B. für 4 Parameter S1.01..S1.04 #IN_CMD3.PAR_N := 4;

- Definition der Zustände in #CMD3_STAT:

■ CMD_UNSED	Int	0
■ CMD_START	Int	1
■ CMD_PROGRESS	Int	2
■ CMD_FINISH	Int	4
■ CMD_FAIL	Int	8

0: keine Aktion

1: Beginn des Lesens

2: Lesevorgang

4: Lesen beendet

8: Lesefehler aufgetreten.

- Starten Sie den Prozess „Lesen“ #CMD3_STAT := 1; (CMD_START)

■ CMD3_STAT	Int
-------------	-----

- Wenn #CMD3_STAT den Wert 4 (CMD_FINISH) erreicht, ist das Lesen beendet und die Werte können in der Struktur #OUT_CMD.VALUE [1..8] gelesen werden. Zusätzlich wird die Anzahl der empfangenen Bytes in der Struktur #OUT_CMD.CNT. angezeigt. Jeder gelesene Parameter benötigt 2 Bytes.

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

- Wenn ein Lesefehler aufgetreten ist, kann die Ursache des Fehlers durch Überprüfen der Ausgänge #COM_xxxxx des Funktionsbausteins analysiert werden.

■ COM_ERROR	Bool
■ COM_NUM	Int
■ COM_EXCE	Int

6.2.4 CMD 6, Schreiben eines Parameters zu einem DMA Modul

Die geschriebenen Werte werden nichtflüchtig gespeichert.

Jedes an den Profibus-Knoten angeschlossene DMA-2-Modul verfügt über eine eigene SNUM (Slave-Nummer). Zusammen mit dieser Nummer ist es möglich, einen neuen Parameterwert in die Parameter-ID dieses bestimmten DMA-2 zu schreiben.

Die Parameter-IDs werden in der Tabelle angezeigt, siehe [7 Parameterliste](#).

Ein Parameterschreibzugriff wird wie folgt durchgeführt:

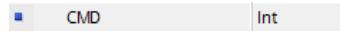
- Definieren Sie die Slave-Nummer SNUM des DMA-2, dessen Parameter geändert werden sollen.

 SNUM Int

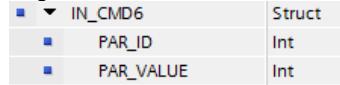
Z.B. für Slave 1

#SNUM := 1;;

- Den Befehl CMD für Schreiben: # CMD := 6;

 CMD Int

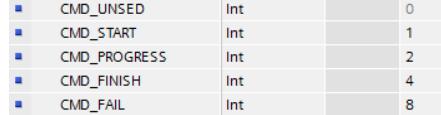
- Legen Sie die Parameter-ID und den Wert fest, der geändert werden soll.

 IN_CMD6 Struct

- PAR_ID Int
- PAR_VALUE Int

Z.B. Parameter 22, S1.01 =1000 (1.0 V) #IN_CMD3.PAR_ID := 22;
#IN_CMD3.PAR_VALUE := 1000;

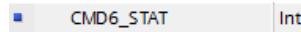
- Definition der Zustände in #CMD6_STAT:



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

0: keine Aktion
1: Beginn des Schreibens
2: In Arbeit schreiben
4: Das Schreiben ist beendet
8: Es ist ein Schreibfehler aufgetreten.

- Starten Sie den Prozess „Schreiben“ #CMD6_STAT := 1; (CMD_START)

 CMD6_STAT Int

- Wenn # CMD6_STAT den Wert 4 (CMD_FINISH) erreicht, ist das Schreiben beendet.

- Wenn ein Schreibfehler aufgetreten ist, kann die Ursache des Fehlers durch Überprüfen der Ausgänge #COM_xxxxxx des Funktionsbausteins analysiert werden.



COM_ERROR	Bool
COM_NUM	Int
COM_EXCE	Int

6.2.5 CMD 15, Schreiben und Lesen wichtiger Prozessparameter zu allen DMA

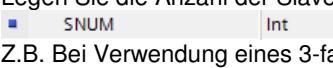
Mit dem CMD 15 ist es möglich, Prozessdaten, von allen DMA's die an den Profibus-Knoten angeschlossen sind, zu senden und zu empfangen. In diesem Fall enthält die Variable SNUM (Slave-Nummer) die maximale Anzahl der angeschlossener DMA-2. (abweichend von CMD 3 oder CMAD6).

Die Bedeutung der Prozesswerte hängt von der Betriebsart des DMA-2 ab.

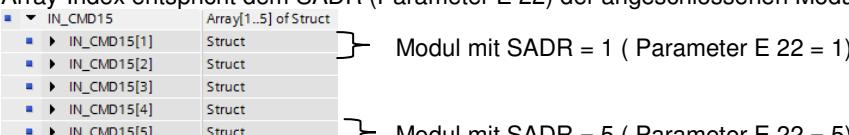
Siehe [4.5 CMD = 15, Struktur des Profibus Telegrams in Abhängigkeit der SW Version](#)

Dies kann aufgrund der DMA-2-Knoten-Zusammensetzung auch für jedes Modul unterschiedlich sein. Auch das Control-Byte ist abhängig vom Arbeitsmodus. Erläuterung der Bits siehe [Fehler! Verweisquelle konnte nicht gefunden werden. Fehler! Verweisquelle konnte nicht gefunden werden.](#)

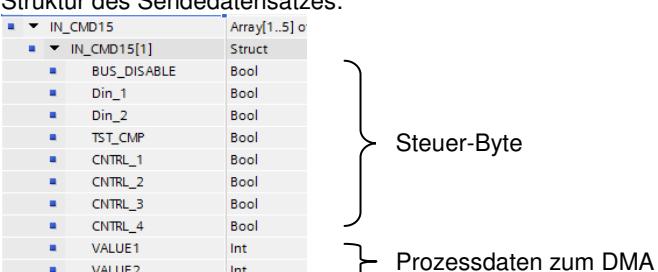
Ein CMD 15-Zugriff wird wie folgt durchgeführt:

- Legen Sie die Anzahl der Slaves SNUM der am Profibus-Knoten angeschlossenen Module fest.

- Den Befehl für CMD 15 eintragen: # CMD := 15;

- Schreiben Sie die Prozessdaten, die Sie an die DMA-2-Module senden wollen, in die vorgesehene Struktur. Der Array-Index entspricht dem SADR (Parameter E 22) der angeschlossenen Module

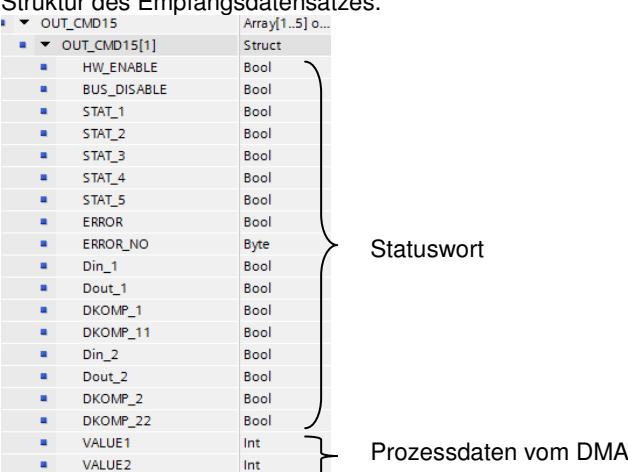


Struktur des Sendedatensatzes:

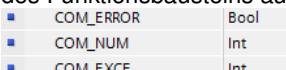


Die interne Telegrammadresse wird automatisch geändert, so dass bei jedem Funktionsblockaufruf ein Rücktelegramm ausgelöst wird. Der Array-Index entspricht dem SADR (Parameter E 22) der angeschlossenen Module.

Struktur des Empfangsdatensatzes.



- Wenn ein Fehler aufgetreten ist, kann die Ursache des Fehlers analysiert werden, indem die Ausgänge #COM_xxxxx des Funktionsbausteins ausgewertet werden.



7 Parameterliste

Parameter dargestellt in "grau" sind noch nicht in der Slave (Modul) Software implementiert

Alle Werte sind in Dezimal. Um Werte als Daten zu verwenden, müssen diese in eine Hex-Zahl gewandelt werden.

R = Parameter kann nur gelesen werden.

X = Implementiert, aber noch nicht Aktiv (kein Zugang)

W/R = schreibbar und lesbar.

ID _{Hex}	ID _{dec}	W/R	Name	Funktion	Einh.	Step	Min	Max	Beschreibung
0x00	0	R	Vers	Software Version	---	---	xx.xx	xx.xx	Abhängig von Hardware
0x01	1	R	d1.01	Summe analoger Sollwerte	V	1	-9999	9999	
0x02	2	R	d1.02	Summe Sollwert nach Rampe	V	1	-9999	9999	
0x03	3	R	d1.03	Sollwerte nach Linearisierung	V	1	-9999	9999	
0x04	4	R	d1.04	Wert nach Verstärkungsanpassung	V	1	-9999	9999	
0x05	5	R	d1.05	Signal A	V	1	-9999	9999	
0x06	6	R	d1.06	Signal B	V	1	-9999	9999	
0x07	7	R	d1.07	Strom A	A	1	0	5000	
0x08	8	R	d1.08	Strom B	A	1	0	5000	1000 == 1.000 A
0x09	9	R	d1.09	Gesamter Strom	A	1	0	5000	
0x0A	10	R	d1.10	Führungsgröße (w)	V	1	-9999	9999	
0x0B	11	R	d1.11	Istwert, Regelgröße (x)	V	1	-9999	9999	
0x0C	12	R	d1.12	Regeldifferenz (e)	V	1	-9999	9999	
0x0D	13	R	d1.13	Ausgang Regler, Stellgröße	V	1	-9999	9999	
0x0E	14	R	d2.01	Summe analoger Sollwerte	V	1	-9999	9999	
0x0F	15	R	d2.02	Summe Sollwert nach Rampe	V	1	-9999	9999	
0x10	16	R	d2.03	Sollwerte nach Linearisierung	V	1	-9999	9999	
0x11	17	R	d2.04	Wert nach Verstärkungsanpassung	V	1	-9999	9999	
0x12	18	R	d2.10	Führungsgröße (w)	V	1	-9999	9999	
0x13	19	R	d2.11	Istwert, Regelgröße (x)	V	1	-9999	9999	
0x14	20	R	d2.12	Regeldifferenz (e)	V	1	-9999	9999	
0x15	21	R	d2.13	Ausgang Regler, Stellgröße	V	1	-9999	9999	
0x16	22	R/W	S1.01	Interner Sollwert 1	V	1	-9999	9999	
0x17	23	R/W	S1.02	Interner Sollwert 2	V	1	-9999	9999	
0x18	24	R/W	S1.03	Interner Sollwert 3	V	1	-9999	9999	
0x19	25	R/W	S1.04	Interner Sollwert 4	V	1	-9999	9999	
0x1A	26	X	S1.08	Reserviert					---
0x1B	27	R/W	r1.01	Rampe von 0 \Rightarrow -	S	1	0	3950	
0x1C	28	R/W	r1.02	Rampe von - \Rightarrow 0	S	1	0	3950	1 == 10ms 0 == Rampe aus
0x1D	29	R/W	r1.03	Rampe von 0 \Rightarrow +	S	1	0	3950	
0x1E	30	R/W	r1.04	Rampe von + \Rightarrow 0	S	1	0	3950	
0x1F	31	R/W	A1.01	Profibus Sollwert (Zweig 1)	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V Wert ist, wenn geschrieben, immer Aktiv. Löschen entweder durch „Reset“ oder durch Schreiben von „0“. Zum Abschalten des analogen Sollwert E17 = „2“ setzen.
0x20	32	R/W	A1.02	Profibus Istwert (Zweig 1)	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V C1.09 muss abgeschaltet werden, um Störungen auszuschließen
0x21	33	R/W	S2.01	Sollwert 1 (Zweig 2)	V	1	-9999	9999	
0x22	34	R/W	S2.02	Sollwert 2 (Zweig 2)	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V
0x23	35	R/W	r2.01	Rampe von 0 \Rightarrow -	S	1	0	3950	
0x24	36	R/W	r2.02	Rampe von - \Rightarrow 0	S	1	0	3950	1 == 10 ms 0 == Rampe aus
0x25	37	R/W	r2.03	Rampe von 0 \Rightarrow + (Zweig 2)	S	1	0	3950	
0x26	38	R/W	r2.04	Rampe von + \Rightarrow 0 (Zweig 2)	S	1	0	3950	

ID _{Hex}	ID _{dec}	W/R	Name	Funktion	Einh.	Step	Min	Max	Beschreibung
0x27	39	R/W	A2.01	Profibus Sollwert (Zweig 2) Ausnahme: in Modus1 auch für Zweig 1	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V Wert ist, wenn geschrieben, immer Aktiv. Löschen entweder durch „Reset“ oder durch Schreiben von „0“. Zum Abschalten des analogen Sollwert E17 = „2“ setzen.
0x28	40	R/W	A2.02	Profibus Istwert (Zweig 2)	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V C2.09 muss abgeschaltet werden, um Störungen auszuschließen
0x29	41	R/W	C1.00	Regleranwahl	---	1	0	4	0 = aus 1 = P-PT ₁ -I-DT ₁ 2 = schaltender Regler 3 = dff 4 = schaltender Regler + dff
0x2A	42	R/W	C1.01	Sicherheitsfunktion	---	1	0	1	0 = aus; 1 = on
0x2B	43	R/W	C1.02	Linearisierung	---	1	0	5	0 = linear; 1 ... 5 = curve
0x2C	44	R/W	C1.03	Verstärkung A	V/V	1	0	200	100 == Faktor 1.00
0x2D	45	R/W	C1.04	Verstärkung B	V/V	1	0	200	100 == Faktor 1.00
0x2E	46	R/W	C1.05	Sollwert Vorzeichen und Verstärkung	V/V	1	-400	400	100 == Faktor 1.00 Vorzeichen und Verstärkung!
0x2F	47	R/W	C1.06	Sollwert Offset	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V
0x30	48	R/W	C1.07	Totbereichskompensation A	V	1	0	9999	1000 == 1.000 V
0x31	49	R/W	C1.08	Totbereichskompensation B	V	1	0	9999	9.999 V = max. Strom je n. Magnet
0x32	50	R/W	C1.09	Sensortyp	---	1	0	12	0 = aus (verwendet für Profibus Anwendung) 1 = 0 ... 20 mA 2 = 4 ... 20 mA 3 = 12 mA ± 8 mA 4 = 0 ... 10 V 5 = 0 ... ± 10 V 6 = 6 V ± 2,5 V 7 = 7,5 V ± 2,5 V 8 = 6 V ± 5 V 9 = 7,5 V ± 5 V 10 = 0 ... 20 mA 11 = 4 ... 20 mA 12 = 0 ... 10 V
0x33	51	R/W	C1.10	Istwert Verstärkungsgain	V/V	1	0	400	100 == Faktor 1.00
0x34	52	R/W	C1.11	Istwert Offset	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V
0x35	53	R/W	C1.12	Istwert Vorzeichen	---	---	-1	+ 1	- 1 = negativ 0 = aus + 1 = positiv
0x36	54	R/W	C1.13	P-Anteil K _{P1}	V/V	1	0	400	100 == Faktor 1.00
0x37	55	R/W	C1.14	T-Anteil für PT1 (to C1.16)	S	1	0	1000	100 == 1.00
0x38	56	R/W	C1.15	Threshold (C1.13, C1.16)	V	1	0	9999	1000 == 1.000 V
0x39	57	R/W	C1.16	P-Anteil K _{P2}	V/V	1	0	400	100 == Faktor 1.00
0x3A	58	R/W	C1.17	I-Anteil	V/s	1	0	4000	1000 == 1.000
0x3B	59	R/W	C1.18	D-Anteil	Vs	1	0	400	100 == 1.00
0x3C	60	R/W	C1.19	T-Anteil für DT1	S	1	0	1000	100 == 1.00
0x3D	61	R/W	C1.20	Verstärkung (C1.13 und C1.16)	V/V	1	1	32	2 = Faktor 2.00
0x3E	62	R/W	C1.21	Komparator obere Schwelle	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V
0x3F	63	R/W	C1.22	Komparator untere Schwelle	V	1	-9999	9999	
0x40	64	R/W	C1.23	Komparator Verzögerung ein	S	1	0	9999	1 == 10 ms
0x41	65	R/W	C1.24	Komparator Verzögerung aus	S	1	0	9999	0 == keine Verzögerung
0x42	66	R/W	C1.25	Komparatorauswahl KOMP_1	---	1	0	3	0 = aus 1 = Sollwert 2 = Istwert 3 = Regeldifferenz
0x43	67	R/W	C1.26	Drahtbruchüberwachung Istwert	---	---	0	1	0 = aus; 1 = Aktiv

ID _{Hex}	ID _{dec}	W/R	Name	Funktion	Einh.	Step	Min	Max	Beschreibung
0x44	68	R/W	C2.00	Regleranwahl	---	1	0	4	0 = aus 1 = P-PT ₁ -I-DT ₁ 2 = Remote 3 = dff 4 = Remote + dff
0x45	69	R/W	C2.01	Sicherheitsfunktion	---	1	0	1	0 = aus; 1 = on
0x46	70	R/W	C2.02	Linearisierung	---	1	0	5	0 = linear; 1 ... 5 = curve
0x47	71	R/W	C2.03	Verstärkung A	V/V	1	0	200	100 == Faktor 1.00
0x48	72	R/W	C2.04	Verstärkung B	V/V	1	0	200	100 == Faktor 1.00
0x49	73	R/W	C2.05	Sollwert Vorzeichen und Verstärkung	V/V	1	-400	400	100 == Faktor 1.00 Vorzeichen und Verstärkung!
0x4A	74	R/W	C2.06	Sollwert Offset	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V
0x4B	75	R/W	C2.07	Totbereichkompensation A	V	1	0	9999	1000 == 1.000 V
0x4C	76	R/W	C2.08	Totbereichkompensation B	V	1	0	9999	9.999 V = max. Strom abhängig von der Magnetauswahl
0x4D	77	R/W	C2.09	Sensortyp	---	1	0	12	0 = aus (used für Profibus application) 1 = 0 ... 20 mA 2 = 4 ... 20 mA 3 = 12 mA ± 8 mA 4 = 0 ... 10 V 5 = 0 ... ± 10 V 6 = 6 V ± 2,5 V 7 = 7,5 V ± 2,5 V 8 = 6 V ± 5 V 9 = 7,5 V ± 5 V 10 = 0 ... 20 mA 11 = 4 ... 20 mA 12 = 0 ... 10 V
0x4E	78	R/W	C2.10	Istwert Verstärkungsgain	V/V	1	0	400	100 == Faktor 1.00
0x4F	79	R/W	C2.11	Istwert Offset	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V
0x50	80	R/W	C2.12	Istwert Vorzeichen	---	---	-1	+ 1	- 1 = negativ 0 = aus + 1 = positiv
0x51	81	R/W	C2.13	P-Anteil K _{P1}	V/V	1	0	400	100 == Faktor 1.00
0x52	82	R/W	C2.14	T-Anteil für PT1 (to C2.16)	S	1	0	1000	100 == 1.00
0x53	83	R/W	C2.15	Threshold (C2.13, C2.16)	V	1	0	9999	1000 == 1.000 V
0x54	84	R/W	C2.16	P-Anteil K _{P2}	V/V	1	0	400	100 == Faktor 1.00
0x55	85	R/W	C2.17	I-Anteil	V/s	1	0	4000	1000 == 1.000
0x56	86	R/W	C2.18	D-Anteil	Vs	1	0	400	100 == 1.00
0x57	87	R/W	C2.19	T-Anteil für DT1	S	1	0	1000	100 == 1.00
0x58	88	R/W	C2.20	Verstärkung (C2.13 und C2.16)	V/V	1	1	32	2 = Faktor 2.00
0x59	89	R/W	C2.21	Komparator obere Schwelle	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V
0x5A	90	R/W	C2.22	Komparator untere Schwelle	V	1	-9999	9999	-
0x5B	91	R/W	C2.23	Komparator Verzögerung ein	S	1	0	9999	1 == 10 ms
0x5C	92	R/W	C2.24	Komparator Verzögerung aus	S	1	0	9999	0 == keine Verzögerung
0x5D	93	R/W	C2.25	Komparatorauswahl KOMP_2	---	1	0	3	0 = aus 1 = Sollwert 2 = Istwert 3 = Regeldifferenz
0x5E	94	R/W	C2.26	Drahtbruchüberwachung Istwert	---	---	0	1	0 = aus; 1 = Aktiv

ID _{Hex}	ID _{dec}	W/R	Name	Funktion	Einh.	Step	Min	Max	Beschreibung
0x5F	95	R	E00	Betriebsmode (Abhängig von Hard- und Software Version)	---	1	1	8	1 = Gesteuert ein Ventil 2 = Gesteuert zwei Ventile 3 = Geregelt ein Ventil 4 = Geregelt ein Prozess 6 = Geregelt Ventil/Prozess 8 = Geregelt Prozess/Proz.
0x60	96	R/W	E01	Analogausgang	---	---	1 und 14	13 und 21	1 = d1.01 to 13 = d1.13 und 14 = d2.01 to 21 = d2.13
0x61	97	R/W	E02	Push-Pull Funktion	---	---	0	1	0 = aus 1 = Aktiv 2 = Common „+“ für Magnete (Anmerkung: nur für max 0,8 A Strom) 3 = Voll H-Brücke 4 = aus + Magnetüberwachung
0x62	98	R/W	E1.03	Magnet Auswahl A	---	1	1	7	1 = 0,8 A 2 = 1,1 A 3 = 1,3 A 4 = 1,6 A 5 = 2,4 A 6 = 2,7 A 7 = 3,5 A
0x63	99	R	E1.04	P-Anteil Stromregler erregen	---	1	0	9999	Default für 2,700 A Magnet
0x64	100	R	E1.05	I-Anteil Stromregler erregen	---	1	0	9999	
0x65	101	R	E1.06	P-Anteil Stromregler entregen	---	1	0	9999	
0x66	102	R	E1.07	I-Anteil Stromregler entregen	---	1	0	9999	
0x67	103	R/W	E08	Rampenart	---	1	0	2	0 = digitale Sollwerte (Zeitkonstant) 1 = alle Sollwerte (Steigungskonstant.)
0x68	104	R/W	E09	Zeitverzögerung Freigabe	s	1	0	9999	1 = 1.00s
0x69	105	R/W	E1.10	Magnetstromanpassung A	---	1	50	110	Variable Anapssung des Nennstroms 100 == Faktor 1.00
0x6A	106	R/W	E11	Ruhestrom Magnet A	V	1	0	3000	3.000 V = 30 % des Nennstroms
0x6B	107	R/W	E12	Ruhestrom Magnet B	V	1	0	3000	
0x6C	108	R/W	E1.13	Dither Amplitude A	V	1	0	3000	
0x6D	109	R/W	E1.14	Dither Frequenz A	Hz	1	1	300	---
0x6E	110	R/W	E15	Signaldefinition Sollwert S1.06 (U/I)	---	1	0	6	0 = S1.06 Spannungseingang Aktiv 1 = 0 ... 20 mA ohne Drahtb. 2 = 10 mA ± 10 mA ohne Drahtb. 3 = 4 ... 20 mA ohne Drahtb. 4 = 4 20 mA mit Drahtbr. 5 = 12 mA +- 8 mA ohne Drahtb. 6 = 12 mA +- 8 mA mit Drahtb.
0x6F	111	R/W	E16	Signaldefinition Sollwert S1.05 (U/I)	---	1	0	6	0 = S1.05 Spannungseingang Aktiv 1 = 0 ... 20 mA ohne Drahtb. 2 = 10 mA ± 10 mA ohne Drahtb. 3 = 4 ... 20 mA ohne Drahtb. 4 = 4 20 mA mit Drahtbr. 5 = 12 mA +- 8 mA ohne Drahtb. 6 = 12 mA +- 8 mA mit Drahtb.
0x70	112	R/W	E17	Sollwert Aktivierungsmoduse	---	1	0	2	0 = 4 digital, 1 analog Aktiv 2 = only 4 digital Aktiv

0x71	113	R/W	E18	Fehler/Komparator Ausgang (from V32.09*)	---	1	0	8	0 = Fehler 1 = Comp1. positive Logik 2 = Comp1. negative Logik 3 = Comp2. positive Logik 4 = Comp2. negative Logik 5 = Dout_1 positive Logik 6 = Dout_1 negative Logik 7 = Dout_2 positive Logik 8 = Dout_2 negative logic
0x72	114	R/W	E19	Reserved					
0x73	115	R/W	E20	Reserved					
0x74	116	R/W	E21	Reserved					
0x75	117	X	FB1M	Reserved					
0x76	118	X	FB1o	Reserved					
0x77	119	X	FB1f	Reserved					
0x78	120	X	DOUT	Reserviert					
0x79	121	X	D_IO	Reserviert					
0x7A	122	X	D_IN	Reserviert					
0x7B	123	X	Uout	Reserviert					
0x7C	124	X	Umon	Reserviert					
0x7D	125	X	1_E4	Reserviert					
0x7E	126	X	1_E5	Reserviert					
0x7F	127	X	1_E6	Reserviert					
0x80	128	X	1_E7	Reserviert					
0x81	129	X	2_E4	Reserviert					
0x82	130	X	2_E5	Reserviert					
0x83	131	X	2_E6	Reserviert					
0x84	132	X	2_E7	Reserviert					
0x85	133	X	3_E4	Reserviert					
0x86	134	X	3_E5	Reserviert					
0x87	135	X	3_E6	Reserviert					
0x88	136	X	3_E7	Reserviert					
0x89	137	X	4_E4	Reserviert					
0x8A	138	X	4_E5	Reserviert					
0x8B	139	X	4_E6	Reserviert					
0x8C	140	X	4_E7	Reserviert					
0x8D	141	X	5_E4	Reserviert					
0x8E	142	X	5_E5	Reserviert					
0x8F	143	X	5_E6	Reserviert					
0x90	144	X	5_E7	Reserviert					
0x91	145	X	6_E4	Reserviert					
0x92	146	X	6_E5	Reserviert					
0x93	147	X	6_E6	Reserviert					
0x94	148	X	6_E7	Reserviert					
0x95	149	X	7_E4	Reserviert					
0x96	150	X	7_E5	Reserviert					
0x97	151	X	7_E6	Reserviert					
0x98	152	X	7_E7	Reserviert					
0x99	153	X	User	Reserviert					

ID _{Hex}	ID _{dec}	W/R	Name	Funktion	Einh.	Step	Min	Max	Beschreibung
0x9A	154	R	So.in	Status der digitalen Eingänge (abhängig von HW Version)	---	---	0	1234	1xxx = S1.01 Aktiv x2xx = S1.02/S2.01 Aktiv xx3x = S1.03 Aktiv xxx4 = S1.04 Aktiv
0x9B	155	R/W	E2.03	Magnet Auswahl A	---	1	1	7	1 = 0,8 A 2 = 1,1 A 3 = 1,3 A 4 = 1,6 A 5 = 2,4 A 6 = 2,7 A 7 = 3,5 A
0x9C	156	R	E2.04	P-Anteil Stromregler erregen	---	1	0	9999	Default für 2,700 A Magnet
0x9D	157	R	E2.05	I-Anteil Stromregler erregen	---	1	0	9999	
0x9E	158	R	E2.06	P-Anteil Stromregler entregen	---	1	0	9999	
0x9F	159	R	E2.07	I-Anteil Stromregler entregen	---	1	0	9999	
0xA0	160	R/W	E2.10	Magnetstromanpassung A	---	1	50	110	Variable Anapssung des Nennstroms 100 == Faktor 1.00
0xA1	161	R/W	E2.13	Dither Amplitude B	V	1	0	3000	3.000 V = 30 % des Nennstroms
0xA2	162	R/W	E2.14	Dither Frequenz B	Hz	1	1	300	---
0xA3	163	R/W	E 22	Modul Slave Adresse	---	1	0	32	0 = Bus Funktion nicht aktiv 1 = Standard (Profibus, CanOpen) 1..5 = Multiple Slave (Profibus)
0xA4	164	R/W	C1.27	Sollwert Hysterese A	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V
0xA5	165	R/W	C2.27	Sollwert Hysterese B	1	1	-9999	9999	
0xA6	166	R/W	C1.28	Komparator obere Schwelle	V	1	-9999	9999	
0xA7	167	R/W	C1.29	Komparator untere Schwelle	V	1	-9999	9999	
0xA8	168	R/W	C1.30	Komparator Verzögerung ein	S	1	0	9999	1 == 10 ms
0xA9	169	R/W	C1.31	Komparator Verzögerung aus	S	1	0	9999	0 == keine Verzögerung
0xAA	170	R/W	C1.32	Komparatorauswahl KOMP_112	---	1	0	3	0 = aus 1 = Sollwert 2 = Istwert 3 = Regeldifferenz
0xAB	171	R/W	C2.28	Komparator obere Schwelle	V	1	-9999	9999	-1000 == -1.000 V; 1000 == 1.000 V
0xAC	172	R/W	C2.29	Komparator untere Schwelle	V	1	-9999	9999	
0xAD	173	R/W	C2.30	Komparator Verzögerung ein	S	1	0	9999	
0xAE	174	R/W	C2.31	Komparator Verzögerung aus	S	1	0	9999	
0xAF	175	R/W	C2.32	Komparatorauswahl KOMP_22	---	1	0	3	0 = aus 1 = Sollwert 2 = Istwert 3 = Regeldifferenz
0xB0	176	R/W	-	Interne digitale Schalter (Read/Set)	---	---	0	32785 0x8011	0x0001 = Din_1 Aktiv (Read/Write) 0x0002 = Dout_1 Aktiv (Read only) 0x0004 = DKOMP_1 Aktiv (Read only) 0x0008 = DKOMP_11 Aktiv (Read only) 0x0010 = Din_2 Aktiv (Read/Write) 0x0020 = Dout_2 Aktiv (Read only) 0x0040 = DKOMP_2 Aktiv (Read only) 0x0080 = DKOMP_22 Aktiv (Read only) 0x0400 = Enable Aktiv 0x0800 = Fehler 0x8000 = Bus Disable Card (Read/Write)
0xB1	177	R/W	E 23	Profibus Telegram Timeout	S	1	0	9999	1 == 10 ms 0 == Funktion nicht aktiv
0xB2	178	X	-	Reserviert					
0xB3	179	R/W	C1.33	I-Anteil Ausgangsbegrenzung	V	1	0	9999	0 == 0.000 V; 9999 == 9.999 V
0xB4	180	R/W	C2.33	I-Anteil Ausgangsbegrenzung	V	1	0	9999	0 == 0.000 V; 9999 == 9.999 V

8 Fehlerbeseitigung

8.1 Allgemein

Die grüne „Power“-LED am Slavemodul leuchtet nicht:
=> Sind alle Module mit Betriebsspannung versorgt? (Supply 24V / Supply 0V)

Die gelbe „Enable“-LED am Slavemodul leuchtet nicht:
=> Sind an allen Modulen die Optoentkoppelten Versorgungen angelegt? (Ext. 24V / Ext. 0V)
=> Ist der Freigabeeingang mit Ext. 24V versorgt?
=> Wurde die Softwarefreigabe ausgeschaltet? (SpezialByte im CMD15-Modus)

Gewünschter Ventilstrom wird nicht erreicht:
=> Liefert die Spannungsversorgung genügend Strom?
=> Richtiger Magnettyp gewählt?

Die rote Fehler LED am Slavemodul leuchtet:
=> Das Slave-Modul befindet sich im Fehlerzustand, dieser muss erst durch wegnahme der Freigabe quittiert werden.

8.2 Keine Kommunikation zum Modul möglich

Es leuchtet die rote LED „BusFehler“:
=> Überprüfen der Profibusverkabelung (Terminierung, Leitungslängen, Steckverbinder)

Es wurde beim projektieren das „Universalmodul“ verwendet:
=> Dies ist nicht zulässig, da nur die in der GSD-Datei verwendeten Module unterstützt werden.
(Das Universalmodul wird von der S7 immer angeboten, auch wenn es nicht in der GSD-Datei vorhanden ist)

8.3 Kommunikation zum Modul ist möglich aber die Ausgänge sind inaktiv

Ist der vorgegebene Sollwert / Istwert im Monitor der Karte zu sehen? (Parameter d1.11 d1.01 d1.02):
Ja => Nachfolgende Parameter Ex.xx Cx.xx sind falsch eingestellt.
Nein => Testweise manuelles vorgeben eines Soll/Istwertes über die Analogen Eingänge

Sind Soll/Istwerte nun sichtbar?
Ja => Die SPS schickt kein stimmiges Telegramm. => Befehle der SPS entsprechend anpassen.
Nein => generelle Fehler wie unter Punkt [8.1 Allgemein](#) beschrieben überprüfen.

ENDE des Dokuments