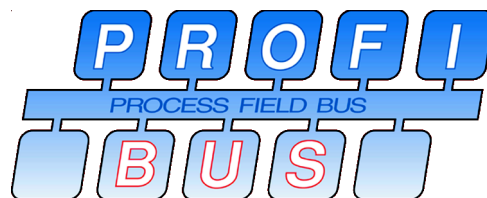


Industrie-Automatisierung **System *HIMatrix***

PROFIBUS-DP ***Master/Slave*** **Handbuch**



Alle in diesem Handbuch genannten HIMA-Produkte sind mit dem HIMA-Warenzeichen geschützt. Dies gilt ebenfalls, soweit nicht anders vermerkt, für andere genannte Hersteller und deren Produkte.

Alle technischen Angaben und Hinweise in diesem Handbuch wurden mit größter Sorgfalt erarbeitet und unter Einschaltung wirksamer Kontrollmaßnahmen zusammengestellt. Trotzdem sind Fehler nicht ganz auszuschließen.

HIMA sieht sich deshalb veranlasst darauf hinzuweisen, dass weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen übernommen werden kann, die auf fehlerhafte Angaben zurückgehen. Für die Mitteilung eventueller Fehler ist HIMA jederzeit dankbar.

Technische Änderungen vorbehalten.

Weitere Informationen sind in der Dokumentation auf der CD-ROM und auf unserer Website unter <http://www.hima.de> zu finden.

HIMA Paul Hildebrandt GmbH
Postfach 12 61
D-68777 Brühl
Tel: +49 (6202) 709 0
Fax: +49 (6202) 709 107
E-Mail: info@hima.com

Inhaltsverzeichnis	Seite
1	HIMatrix PROFIBUS-DP Master9
1.1	Benötigte Ausstattung und Systemanforderungen.....9
1.2	PROFIBUS-DP Master Eigenschaften9
1.3	Anlegen eines HIMatrix PROFIBUS-DP Master10
1.4	HIMatrix PROFIBUS-DP Master Kontextmenü10
1.4.1	Menüfunktion <i>Signale verbinden</i>11
1.4.2	Standard Menüfunktionen12
1.4.2.1	Menüfunktion <i>Validieren</i>12
1.4.2.2	Menüfunktion <i>Neu</i>12
1.4.2.3	Menüfunktionen Import, Export.....12
1.4.2.4	Menüfunktionen <i>Kopieren, Einfügen, Löschen, Drucken</i>12
1.4.3	Menüfunktion <i>Eigenschaften</i>13
1.4.3.1	Register <i>Allgemein</i>13
1.4.3.2	Register <i>Zeiten</i>13
1.4.3.3	Register CPU/COM.....15
1.4.3.4	Register <i>Sonstige</i>16
1.4.4	Isochron Mode (ab DP-V2)17
1.4.4.1	Isochron Mode Sync (ab DP-V2)17
1.4.4.2	Isochrone Mode Freeze (ab DP-V2)17
1.4.5	Richtwerte für verschiedene Übertragungsraten17
1.4.5.1	Berechnung der Token-Umlaufzeit <i>Ttr</i>18
1.4.5.2	Beispiel zur Berechnung der Token-Umlaufzeit <i>Ttr</i>19
1.5	PROFIBUS-DP Slave Kontextmenü.....20
1.5.1	Menüfunktion <i>Signale verbinden</i>20
1.5.2	Menüfunktion <i>GSD Datei einlesen</i>21
1.5.3	Menüfunktion <i>Module einfügen</i>22
1.5.3.1	Signal Zuordnung in den PROFIBUS-DP Modulen22
1.5.4	Menüfunktion <i>User-Parameter bearbeiten</i>23
1.5.5	Konfiguration der Benutzerdaten in verschiedenen Blöcken24
1.5.6	Menüfunktion <i>Validieren</i>25
1.5.7	Menüfunktion <i>Eigenschaften</i>26
1.5.7.1	Register <i>Parameter</i>26
1.5.7.2	Register <i>Gruppen</i>27
1.5.7.3	Register <i>DP-V1</i>28
1.5.7.4	Register <i>Alarmer</i>29
1.5.7.5	Register <i>Daten</i>29
1.5.7.6	Register <i>Modell</i>30
1.5.7.7	Register <i>Features</i>30
1.5.7.8	Register <i>Übertragungsraten</i>31
1.5.7.9	Register <i>Azyklisch</i>31
1.6	Diagnose und Protokollzustände des HIMatrix PROFIBUS-DP Master32
1.6.1	Statistikwerte zurücksetzen:.....32
1.6.2	<i>PB Master</i>32
1.6.2.1	Schaltflächen.....32
1.6.2.2	Anzeigefeld.....33
1.6.3	<i>PB Slaves</i>34
1.6.3.1	Schaltflächen.....34
1.6.3.2	Anzeigefeld.....35
1.6.4	Protokollzustände des HIMatrix PROFIBUS-DP Master.....35
1.6.5	Verhalten des HIMatrix PROFIBUS-DP Master.....36
1.6.6	Funktion der FBx LED beim PROFIBUS Master36
1.7	Beispiel: Konfiguration des HIMatrix PROFIBUS-DP Master.....37
2	HIMatrix PROFIBUS-DP Funktionsbausteine48
2.1	Konfiguration der Funktionsbausteine48
2.1.1	PROFIBUS-DP Funktionsbausteinbibliothek49

2.1.2	Konfiguration der Funktionsbaustein im Anwenderprogramm.....	49
2.1.3	Konfiguration der Funktionsbausteine im Hardware Management.....	50
2.2	Funktionsbaustein <i>MSTAT</i>.....	51
2.2.1	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix A</i>	51
2.2.2	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix F</i>	52
2.2.3	Funktionsbaustein <i>MSTAT</i> im Hardware-Management erstellen.....	53
2.2.4	Funktionsablauf.....	53
2.3	Funktionsbaustein <i>RALRM</i>.....	54
2.3.1	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix A</i>	54
2.3.2	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix F</i>	55
2.3.3	Funktionsbaustein <i>RALRM</i> im Hardware-Management erstellen.....	56
2.3.4	Funktionsablauf.....	57
2.4	Funktionsbaustein <i>RDIAG</i>.....	58
2.4.1	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix A</i>	58
2.4.2	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix F</i>	59
2.4.3	Funktionsbaustein <i>RDIAG</i> im Hardware-Management erstellen.....	60
2.4.4	Funktionsablauf.....	61
2.5	Funktionsbaustein <i>RDREC</i>.....	62
2.5.1	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix A</i>	62
2.5.2	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix F</i>	63
2.5.3	Funktionsbaustein <i>RDREC</i> im Hardware-Management erstellen.....	64
2.5.4	Funktionsablauf.....	65
2.6	Funktionsbaustein <i>SLACT</i>.....	66
2.6.1	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix A</i>	66
2.6.2	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix F</i>	67
2.6.3	Funktionsbaustein <i>SLACT</i> im Hardware-Management erstellen.....	68
2.6.4	Funktionsablauf.....	68
2.7	Funktionsbaustein <i>WRREC</i>.....	69
2.7.1	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix A</i>	69
2.7.2	Ein- und Ausgänge des Funktionsbausteins mit dem <i>Präfix F</i>	70
2.7.3	Funktionsbaustein <i>WRREC</i> im Hardware-Management erstellen.....	71
2.7.4	Funktionsablauf.....	72
2.8	Fehlercodes der Funktionsbausteine.....	73
2.9	Hilfsfunktionsbausteine.....	74
2.9.1	Hilfsfunktionsbausteine die mit dem Identifier arbeiten.....	75
2.9.1.1	Hilfsfunktionsbaustein <i>ID</i>	75
2.9.1.2	Hilfsfunktionsbaustein <i>SLOT</i>	76
2.9.1.3	Hilfsfunktionsbaustein <i>NSLOT</i>	77
2.9.1.4	Hilfsfunktionsbaustein <i>DEID</i>	78
2.9.2	Hilfsfunktionsbausteine die mit der Standarddiagnose arbeiten.....	79
2.9.2.1	Hilfsfunktionsbaustein <i>ACTIVE</i>	79
2.9.2.2	Hilfsfunktionsbaustein <i>STDDIAG</i>	80
2.9.2.3	Hilfsfunktionsbaustein <i>Alarm</i>	82
2.10	Beispiel: Konfiguration des Funktionsbausteins <i>RDIAG</i>.....	84
3	HIMatrix PROFIBUS-DP Slave.....	89
3.1	Benötigte Ausstattung und Systemanforderungen.....	89
3.2	HIMatrix PROFIBUS-DP Slave Eigenschaften.....	89
3.3	Funktion der FBx LED beim PROFIBUS-DP Slave.....	90
3.4	HIMatrix PROFIBUS-DP Slave Kontextmenü.....	91
3.4.1	Menüfunktion <i>Signale verbinden</i>	91
3.4.1.1	Register <i>Eingänge</i>	91
3.4.1.2	Register <i>Ausgänge</i>	93
3.4.2	Menüfunktion <i>Eigenschaften</i>	93
3.4.2.1	Register <i>Allgemein</i>	93
3.4.2.2	Register <i>CPU/COM</i>	94
3.5	Beispiel: Konfiguration eines HIMatrix PROFIBUS-DP Slave.....	95
3.5.1	Signale im HIMatrix PROFIBUS-DP Slave zuordnen.....	95

3.5.1.1	PROFIBUS-DP Slave Konfiguration Prüfen.....	95
3.5.2	Konfiguration des PROFIBUS-DP Slave im PROFIBUS-DP Master	96
3.5.2.1	Anlegen der <i>HIMatrix</i> PROFIBUS-DP-Module.....	96
3.5.2.2	Signal Zuordnung in den PROFIBUS-DP Modulen	97
3.5.2.2.1	Signal Zuordnung in den Eingangsmodulen	97
3.5.2.2.2	Signal Zuordnung in den Ausgangsmodule	98
3.5.2.3	Anlegen der Benutzerdaten im PROFIBUS-DP Master	99
3.5.2.4	PROFIBUS-DP Master Konfiguration Prüfen.....	99
4	HIMatrix PROFIBUS-DP Grundlagen	100
4.1	DP-Leistungsstufen	100
4.2	PROFIBUS-DP Gerätetypen	100
4.3	Hardwaregrundlagen der seriellen Datenübertragung.....	101
4.3.1	Grundlegende technische Eigenschaften der RS-485-Übertragung	101
4.3.2	Reichweite in Abhängigkeit von der Baudrate	101
4.3.3	Busanschluss und Busabschluss.....	102
4.3.4	PROFIBUS-DP Buskabel.....	103
4.3.5	Bus-Topologie	104
4.4	Die PROFIBUS-DP Telegramme.....	105
4.4.1	Funktionsbytes für die PROFIBUS-DP-Telegramme	105
4.4.2	Verwendete PROFIBUS-DP Telegramme der HIMatrix Steuerungen.....	106
4.4.3	Mögliche Stationsadressen in den Telegrammfeldern DA und SA.....	106
4.4.4	Schutzmechanismen der PROFIBUS-DP Telegramme zur Datensicherung .	107
4.4.5	Die PROFIBUS-DP Buszugriffsverfahren	107
4.4.5.1	Master/Slave-Protokoll	107
4.4.5.2	Token-Protokoll	107
4.5	Isochroner PROFIBUS-DP Zyklus (ab DP-V2)	109
4.6	Zyklischer PROFIBUS-DP Zyklus (ab DP-V0).....	110
4.6.1	Pollzyklus	111
4.6.2	Zeiten des Pollzyklus.....	112
4.6.2.1	Idle Time (Tid)	112
4.6.2.2	Slot Time (Tsl).....	112
4.6.2.3	Synchronization Time (Tsyn)	113
4.6.2.4	Station Delay Time (Tsdx).....	113
4.6.2.5	Quiet Time (Tqui)	113
4.6.2.6	Safety Margin (Tsm).....	114
4.6.2.7	Time-out Time (Tto).....	114
4.6.2.8	Weitere Zeiten des Pollzyklus	114
	Literaturverzeichnis	115

Zu diesem Handbuch

Dieses Handbuch beschreibt die Konfiguration und den Betrieb des HIMatrix PROFIBUS-DP Masters und des HIMatrix PROFIBUS-DP Slaves. Es richtet sich an den qualifizierten Anwender, der den Umgang mit dem Programmiertool *ELOP II Factory* und den HIMA HIMatrix Steuerungen beherrscht.

Das Handbuch ist in vier Kapitel gegliedert:

1. *HIMatrix PROFIBUS-DP Master* beschreibt die Menüfunktionen und die Dialoge in *ELOP II Factory* zur Konfiguration des HIMatrix PROFIBUS-DP Master.
2. *HIMatrix PROFIBUS-DP Funktionsbausteine* beschreibt die Funktion und Konfiguration der PROFIBUS-DP Master Funktionsbausteine.
3. *HIMatrix PROFIBUS-DP Slave* beschreibt die Menüfunktionen und Dialoge in *ELOP II Factory* zur Konfiguration des HIMatrix PROFIBUS-DP Slave.
4. *HIMatrix PROFIBUS-DP Grundlagen* gibt dem interessierten Leser Hintergrundinformationen zum HIMatrix PROFIBUS-DP Protokoll.

Für weitere Informationen zu PROFIBUS-DP verweist HIMA auf die folgenden Spezifikationen der PROFIBUS Nutzerorganisation e.V. (siehe www.profibus.de):

- PROFIBUS Technologie und Anwendung, Oktober 2002
- PROFIBUS Guideline-Ordner No. 2.182 Version 1.2, Juli 2001

Sollten Sie weitere Fragen haben, wenden Sie sich bitte direkt an HIMA.

Alle Rechte und technische Änderungen vorbehalten.

© HIMA Paul Hildebrandt GmbH
Postfach 1261
D - 68777 Brühl bei Mannheim

Terminologie

Begriff	Definition
ASIC	Anwenderspezifischer Schaltkreis
Cfg	Konfigurationskontrolle
CRC	Cyclic Redundancy Check
COM	Kommunikationsmodul
CPU	Zentralbaugruppe
DDLM	DP User Interface (Direct Data Link Mapper)
DP	Dezentrale Peripherie
DP-V0	DP Version 0: Stellt die Grundfunktionalität von DP zur Verfügung
DP-V1	DP Version 1: Ergänzungen, wie z.B. azyklischer Datenverkehr
DP-V2	DP Version 2: Weitere Ergänzungen, wie z.B. isochroner Slavebetrieb
EMV	Elektromagnetische Verträglichkeit
EN	Europäische Normen
FB	Feldbus
FBS	Funktionsbausteinsprache
FDL	PROFIBUS Buszugriffsprotokoll (Fieldbus Data Link)
GSD-Datei	Die GSD-Datei enthält die Geräte-Stammdaten des Produkts und wird vom Gerätehersteller bereitgestellt.
HSA	Highest Station Address
HWM	ELOP II Factory Hardwaremanagement
Identifizier	16-Bit-Nummer, die von der PROFIBUS Nutzerorganisation e.V. einem Gerät zugeteilt wird und eindeutig kennzeichnet.
LAS	Liste der aktiven Busteilnehmer
Master Klasse 1	Master für den Nutzdatenverkehr
Master Klasse 2	Master für die Projektierung und Inbetriebnahme (Programmiergerät)
NIL	Nichts oder Null
ISO	International Standard Organisation
PADT (PC)	Programming and Debugging Tool (nach IEC 61131-3)
PES	Programmierbares Elektronisches System
PI	PROFIBUS International
PM	ELOP II Factory Projektmanagement
PNO	PROFIBUS Nutzerorganisation e.V.
SAP	Service Access Point (Dienstzugangspunkt)
Slave	Ein Slave ist eine passive Station und tauscht Nutzdaten mit dem Master aus.
Tbit	Zeiteinheit für die Übertragung eines Bits, Kehrwert der Übertragungsrate (Bsp. 1Tbit bei 12 Mbit/s = 83ns).
TMO	Timeout
UART	Universal Asynchronous Receiver/ Transmitter
Validieren	Auf Gültigkeit prüfen

Einführung

PROFIBUS-DP ist ein internationaler, offener Feldbusstandard und wird überall dort eingesetzt, wo eine schnelle Reaktionszeit bei vornehmlich kleinen Datenmengen gefordert wird.

Der HIMatrix PROFIBUS-DP Master und der HIMatrix PROFIBUS-DP Slave erfüllen die Kriterien der europäischen Norm EN 50170 [7] und der weltweit verbindlichen Norm IEC 61158 für PROFIBUS-DP.

Der HIMatrix PROFIBUS-DP Master kann mit den verbundenen PROFIBUS-DP Slaves zyklisch und/oder azyklisch Daten austauschen.

Zum azyklischen Datenaustausch stehen dem Anwender in ELOP II Factory verschiedene Funktionsbausteine zur Verfügung. Mit diesen Funktionsbausteinen kann der Anwender den HIMatrix PROFIBUS-DP Master optimal den Erfordernissen der PROFIBUS-DP Slaves und des Projekts anpassen.

1 HIMatrix PROFIBUS-DP Master

Dieses Kapitel beschreibt die Eigenschaften des HIMatrix PROFIBUS-DP Master sowie die Menüfunktionen und Dialoge in *ELOP II Factory*, die zur Konfiguration des HIMatrix PROFIBUS-DP Master benötigt werden.

Hinweis: Für jeden HIMatrix-Steuerungs Typ steht die jeweilige Systemdokumentationen mit den elektrischen und mechanischen Daten zur Verfügung. (Siehe Projektierungshandbuch HI 800 100 und Datenblätter der jeweiligen HIMatrix Steuerung).

1.1 Benötigte Ausstattung und Systemanforderungen

HIMA ELOP II Factory	Ab Version 5.2.0
HIMatrix Steuerungen	F20 F30, F35 und F60 ab Hardware Revision: 02
Betriebssystemversionen der HIMatrix Steuerungen	COM BS ab Version 6.22 CPU BS ab Version 4.50
HIMatrix PROFIBUS-DP-Master-Modul	Die HIMatrix Steuerung muss an der verwendeten seriellen Feldbusschnittstelle (FB1 oder FB2) mit einem optionalen HIMatrix PROFIBUS-DP Modul ausgerüstet sein.
Lizenznummer	Wird nicht benötigt, Freischaltung durch Modul

1.2 PROFIBUS-DP Master Eigenschaften

Typ des HIMatrix PROFIBUS-DP Master	DP-V1 Master Klasse 1 mit zusätzlichen Funktionen aus DP-V2
Übertragungsrate	9,6 kbit/s bis 12 Mbit/s
Busadresse	0 bis 125
Max. Anzahl PROFIBUS-DP Master	Es kann nur ein PROFIBUS-DP Master pro Ressource konfiguriert werden. F30, F35, F60: wahlweise auch zwei Master
Max. Anzahl PROFIBUS-DP Slaves	Es können bis zu 125 ¹ Slaves pro Ressource (in allen Master Protokollinstanzen) konfiguriert werden. Hierbei besteht die Restriktion, daß maximal 32 Busteilnehmer an ein Bussegment ohne Repeater angeschlossen werden können.
Max. Prozessdatenlänge zu einem Slave	DP-Output: max. 244 Bytes DP-Input: max. 244 Bytes
Verbindungsüberwachung	



Ist der PROFIBUS-DP Master im Zustand OPERATE und die Verbindung zu einem PROFIBUS-DP Slave geht verloren, dann wird dies vom PROFIBUS-DP Master nach wenigen PROFIBUS-DP Zyklen erkannt.

In diesem Fall wird der Verbindungszustand auf OFFLINE gesetzt. Die Eingangssignale von diesem PROFIBUS-DP Slave werden ignoriert und stattdessen die Initialwerte verwendet.

¹ Nach der Norm sind insgesamt drei Repeater zulässig, so dass maximal 122 Slaves pro serielle Schnittstelle eines Masters angeschlossen werden können.

Hinweis: Neben dem PROFIBUS-DP Protokoll können gleichzeitig noch weitere Protokolle (z.B. Modbus, TCP S/R ...) auf der *HIMatrix*-Steuerung betrieben werden.

Insgesamt können pro *HIMatrix*-Steuerung 16284 Byte Daten gesendet und 16284 Byte Daten empfangen werden.

Die 16284 Byte Daten, können beliebig zwischen den Protokollen aufgeteilt werden, jedoch nicht mehr als 8192 Byte für ein Protokoll und Richtung.

1.3 Anlegen eines HIMatrix PROFIBUS-DP Master

Starten Sie *ELOP II Factory* und erstellen Sie ein neues Projekt, oder laden Sie ein vorhandenes Projekt.

Wechseln Sie danach ins Hardware-Management und wählen Sie aus dem Kontextmenü für Protokolle **Neu, Profibus Master**, um einen neuen PROFIBUS-DP Master in der Resource anzulegen.

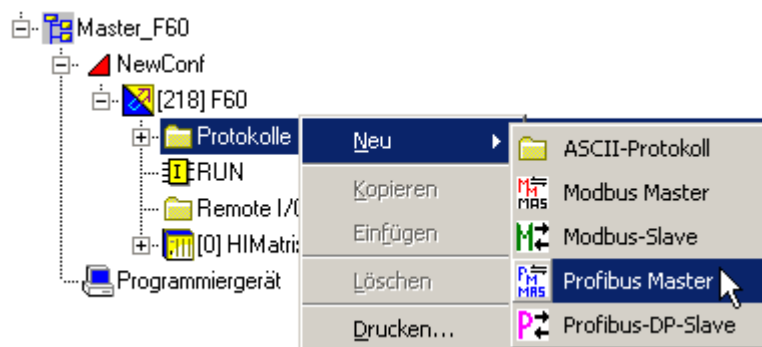


Bild 1: Neuer *PROFIBUS-DP Master*

1.4 HIMatrix PROFIBUS-DP Master Kontextmenü

Das Kontextmenü des HIMatrix PROFIBUS-DP Master enthält die folgenden Funktionen.

PROFIBUS-DP Master
Signale verbinden
Validieren
Neu
Import
Export
Kopieren
Einfügen
Löschen
Drucken
Eigenschaften

1.4.1 Menüfunktion *Signale verbinden*

Die Menüfunktion *Signale verbinden* aus dem Kontextmenü des PROFIBUS-DP Master öffnet den Dialog *Signal-Zuordnungen*.

Der Dialog *Signal-Zuordnungen* stellt die beiden Statussignale

- Bus-Fehler und
- Master-Status

bereit, die es erlauben, den Zustand des PROFIBUS-DP Master im Anwenderprogramm auszuwerten.

Signal	Beschreibung	Typ
Bus-Fehler	<p>Tritt ein Busfehler auf, wird im System-Signal <i>Bus-Fehler</i> ein Fehlercode gesetzt. Ein Fehlercode bleibt solange anliegen, bis der Busfehler behoben ist.</p> <p>Fehlercodes:</p> <p>0: OK, kein Busfehler</p> <p>1: Adressfehler Die Adresse des Masters ist auf dem Bus bereits vorhanden.</p> <p>2: Busstörung Es wurde eine Störung auf dem Bus registriert, (z.B. Bus nicht richtig abgeschlossen, mehrere Teilnehmer senden gleichzeitig).</p> <p>3: Protokollfehler Ein fehlerhaft codiertes Paket wurde empfangen.</p> <p>4: Hardwarefehler Die Hardware hat einen Fehler gemeldet, z.B. bei zu kurz eingestellten Zeiten.</p> <p>5: Unbekannter Fehler Master hat Zustand aus unbekanntem Grund gewechselt.</p> <p>6: Controller Reset Controller-Chip wird bei schweren Busstörungen zurückgesetzt.</p> <p>Das Statussignal <i>M1BusErr</i> kann im Anwenderprogramm ausgewertet werden.</p>	BYTE
Master-Status	<p>Der Master-Status zeigt den momentanen Protokollzustand an.</p> <p>0: OFFLINE</p> <p>1: STOP</p> <p>2: CLEAR</p> <p>3: OPERATE</p> <p>(Siehe Kapitel 1.6.4)</p> <p>Das Statussignal <i>Master-Status</i> kann im Anwenderprogramm ausgewertet werden.</p>	BYTE

Tabelle 1: Statussignale des PROFIBUS-DP Master

1.4.2 Standard Menüfunktionen

1.4.2.1 Menüfunktion *Validieren*

Vor der Codegenerierung kann die Parametrierung des Masters und der Slaves getestet werden. In der Strukturansicht wird das entsprechende Objekt selektiert und im Kontextmenü wird *Validieren* gewählt. In der Fehler-Status-Anzeige werden dann eventuelle Fehler und Warnungen angezeigt.

Die Validation wird zudem automatisch vor jeder Codegeneration durchgeführt. Wird bei der Validation ein Fehler festgestellt, dann wird die Codegeneration abgebrochen.

1.4.2.2 Menüfunktion *Neu*

Mit **Neu** aus dem Kontextmenü können neue Objekte hinzugefügt werden.

1.4.2.3 Menüfunktionen *Import, Export*

Unter den Menüfunktionen **Import** und **Export** aus dem Kontextmenü werden die Elemente angezeigt, in welche die über csv-Dateien importiert und exportiert werden können.

1.4.2.4 Menüfunktionen *Kopieren, Einfügen, Löschen, Drucken*

Kopieren: Kopiert das Objekt in die Zwischenablage

Einfügen: Fügt das Objekt aus der Zwischenablage hinzu.

Löschen: Löscht das gewählte Objekt aus dem Projekt.

Drucken: Sendet die Konfiguration des Objekts an den gewählten Drucker.

1.4.3 Menüfunktion *Eigenschaften*

Mit *Eigenschaften* im Kontextmenü des HIMatrix PROFIBUS-DP Master wird der Dialog *Eigenschaften* geöffnet.

Das Dialogfenster enthält die Register *Allgemein*, *Zeiten*, *CPU/COM* und *Sonstige*.

1.4.3.1 Register *Allgemein*

Element	Beschreibung	Wert
Typ	PROFIBUS-DP Master	Nur Anzeige
Name	Beliebiger, eindeutiger Name, für einen PROFIBUS-DP Master	
Adresse	Stationsadresse des Master Die Stationsadresse des Masters darf auf dem Bus nur einmal vorhanden sein.	Min: 0 Max: 125 Standard: 0
Schnittstelle	COM-Schnittstelle, die für den Master benutzt werden soll.	FB1, FB2
Baudrate	Baudrate, mit welcher der Bus betrieben wird. Mögliche Werte: 9600 (9,6 kBaud) 19200 (19,2 kBaud) 45450 (45,45 kBaud) 93750 (93,75 kBaud) 187500 (187,5 kBaud) 500000 (500 kBaud) 1500000 (1,5 MBaud) 3000000 (3 MBaud) 6000000 (6 MBaud) 12000000 (12 MBaud)	

Tabelle 2: Allgemeine Einstellungen des PROFIBUS-DP Master im Dialog *Eigenschaften*

1.4.3.2 Register *Zeiten*

Element	Beschreibung	Wert
MinTsdr [bit time]	Min. Station Delay Time Zeit, die ein PROFIBUS-DP Slave mindestens warten muss, bevor er antworten darf.	Min: 11 Max: 1023 Standard: 11
MaxTsdr [bit time]	Max. Station Delay Time: Zeit, die ein PROFIBUS-DP Slave maximal benötigen darf, um zu antworten. $\text{MaxTsdr} \geq \max(\text{MaxTsdr aller Slaves})$ Die MaxTsdr-Werte der Slaves werden aus den GSD-Dateien gelesen und befinden sich im <i>Eigenschaften</i> -Dialog der Slaves im Register <i>Übertragungsraten</i>	Min: 37 Max: 65525 Standard: 37
Tsl [bit time]	Slot Time Maximale Zeitspanne, in welcher der Master auf eine Antwort des Slaves wartet. $\text{Tsl} > \text{MaxTsdr} + 2 \cdot \text{Tset} + \text{Tqui} + 13$	Min: 37 Max: 16383 Standard: 37

Tqui [bit time]	Quiet Time for Modulator (Modulatorausklingzeit) Zeit, die ein Teilnehmer für das Umschalten von Senden auf Empfangen benötigen darf.	Min: 0 Max: 493 Standard: 0
Tset [bit time]	Setup Time Reaktionszeit auf ein Ereignis	Min: 1 Max: 494 Standard: 1
Ttr [bit time]	Projektierte Token-Umlaufzeit Die maximale zur Verfügung gestellte Zeit für einen Token-Umlauf. Eine untere Abschätzung der Ttr erhält man durch folgende Formel: $Ttr = n \cdot (198 + T1 + T2) + b \cdot 11 + 242 + T1 + T2 + Ts$ Siehe Kapitel 1.4.5.	Min: 256 Max: 16777215 Standard: 999
Ttr [ms]	Projektierte Token-Umlaufzeit in ms	Nur Anzeige
Min. Slave Intervall [ms]	Mindestzeit für einen PROFIBUS-DP-Zyklus Die Zeit <i>Min. Slave Intervall</i> wird vom Master eingehalten und auf keinen Fall unterschritten. Der PROFIBUS-DP-Zyklus kann sich jedoch verlängern, wenn der Isochron Mode inaktiv ist und der Anteil der azyklischen Telegramme in einem Zyklus ansteigt. Der Wert für <i>Min. Slave Intervall</i> des Slaves wird aus der GSD-Datei gelesen und befindet sich im Dialog <i>Eigenschaften</i> des Slaves im Register <i>Features</i> . Im Isochron Mode gibt <i>Min. Slave Intervall</i> die Zeit des isochronen Zyklus vor. Der Isochron Mode wird aktiviert wenn Isochron Mode Sync oder Freeze aktiviert sind. Siehe auch Aktualisierungszeit zwischen CPU und COM in Kapitel 1.4.3.3.	Min: 0 Max: 6553.5 Standard: 1
Nutzdatenüberwachungszeit [ms]	Zeitspanne, innerhalb welcher der Master seinen momentanen Zustand auf dem Bus mitteilen muss. Richtwert: Nutzdatenüberwachung = WDZ des Slaves	Wertebereich 0-65535 [10ms] Standard: 1000

Tabelle 3: Einstellung der Zeiten des PROFIBUS-DP Master im Dialog *Eigenschaften*

1.4.3.3 Register CPU/COM

Die Vorgabewerte für die Parameter sorgen für den schnellstmöglichen Datenaustausch der PROFIBUS DP Daten zwischen dem COM-Prozessor (COM) und dem CPU-Prozessor (CPU) in der HIMatrix Steuerung.

Diese Parameter sollten nur dann geändert werden, wenn eine Reduzierung der COM- und/oder CPU-Auslastung für eine Anwendung erforderlich ist und der Prozess dies zulässt.



Die Änderung der Parameter wird nur für den erfahrenen Programmierer empfohlen.

Eine Erhöhung der COM und CPU Aktualisierungszeit bedeutet auch, dass die tatsächliche Aktualisierungszeit der PROFIBUS-DP Daten erhöht wird. Die Zeitanforderungen der Anlage sind zu prüfen.

Beachten Sie auch den Parameter *Min. Slave Intervall [ms]* (siehe 1.4.3.2), der die Aktualisierungszeit der PROFIBUS DP Daten vom/zum PROFIBUS DP Slave festlegt.

Dieser kann entsprechend der COM/CPU-Aktualisierungszeit erhöht werden.

Element	Beschreibung
Refresh Rate [ms]	<p>Aktualisierungszeit in Millisekunden, mit der die Daten des Protokolls zwischen COM und CPU ausgetauscht werden.</p> <p>Ist die <i>Refresh Rate</i> Null oder kleiner als die Zykluszeit der Steuerung, dann erfolgt der Datenaustausch so schnell wie möglich.</p> <p>Wertebereich: 0 bis $(2^{31}-1)$</p> <p>Vorgabewert: 0</p>
In einem Zyklus	<p>Aktiviert Transfer der gesamten Daten des Protokolls von der CPU zur COM innerhalb eines Zyklus der CPU.</p> <p>Deaktiviert Transfer der gesamten Daten des Protokolls von der CPU zur COM, verteilt über mehrere CPU Zyklen zu je 900 Byte pro Datenrichtung.</p> <p>Damit kann eventuell auch die Zykluszeit der Steuerung reduziert werden.</p> <p>Vorgabewert: Aktiviert</p>

Tabelle 4: Eigenschaften des PROFIBUS-DP Master

1.4.3.4 Register *Sonstige*

Element	Beschreibung	Wert
Max. Anz. Sendewdh.	Maximale Anzahl an Sendewiederholungen eines Masters, wenn ein Slave nicht antwortet.	Min: 0 Max: 7 Standard: 1
Highest Active Address	Highest Station Address (HSA) Höchste zu erwartende Stationsadresse eines Masters. Master, mit Stationsadressen jenseits von HSA, werden nicht in den Token-Ring aufgenommen.	Min: 0 Max: 125 Standard: 125
Isochron Mode Sync	Der Isochron Mode Sync ermöglicht eine taktsynchrone Regelung in Master und Slave und ein zeitgleiches Aktivieren der physikalischen Ausgänge mehrerer Slaves. Ist der Isochron Mode Sync aktiv, dann sendet der Master den Steuerbefehl <i>Sync</i> als Broadcast-Telegramm an alle Slaves. Sobald Slaves, die den Isochron Mode unterstützen, den Steuerbefehl <i>Sync</i> erhalten, schalten sie die Daten aus dem Anwenderprogramm zeitgleich auf die physikalischen Ausgänge. Die Werte der physikalischen Ausgänge bleiben bis zum nächsten Sync-Befehl eingefroren. Die Zykluszeit wird durch das <i>Min. Slave Intervall</i> vorgegeben. Bedingung: $T_{tr} < \text{Min. Slave Intervall}$	Standard: FALSE
Isochron Mode Freeze	Der Isochron Mode Freeze ermöglicht eine zeitgleiche Übernahme der Eingangsdaten mehrerer Slaves. Ist der Isochron Mode Freeze aktiv, sendet der Master den Steuerbefehl <i>Freeze</i> als Broadcast-Telegramm an alle Slaves. Sobald die Slaves, die den Isochron Mode unterstützen, den Steuerbefehl <i>Freeze</i> erhalten, werden die Signale der physikalischen Eingänge auf dem momentanen Wert eingefroren. Die Werte können dann vom Master gelesen werden. Erst nach einem weiteren Steuerbefehl <i>Freeze</i> werden die Eingangsdaten aktualisiert. Die Zykluszeit wird durch das <i>Min. Slave Intervall</i> vorgegeben. Bedingung: $T_{tr} < \text{Min. Slave Intervall}$	Standard: FALSE
Auto-Clear bei Fehler	Der Master geht in den Zustand CLEAR, wenn ein Slave ausfällt bei dem <i>Auto-Clear bei Ausfall</i> gesetzt ist.	Standard: FALSE
Zeitmaster	Der Master ist auch Zeitmaster und versendet die Systemzeit periodisch über den Bus.	Standard: FALSE
Clock Sync Interval [ms]	Uhr-Synchronisations-Intervall. Zeitspanne, innerhalb welcher der Zeitmaster die Systemzeit auf dem Bus versendet.	Min: 0 Max: 65535 Standard: 0

Tabelle 5: Sonstige Einstellungen des PROFIBUS-DP Master im Dialog *Eigenschaften*

1.4.4 Isochron Mode (ab DP-V2)

Diese Funktion ermöglicht eine taktsynchrone Regelung in Master und Slave, unabhängig von der Belastung des Busses. Der Buszyklus wird mit einer Taktabweichungen von $< 10 \mu\text{s}$ synchronisiert. Damit können hochgenaue Positionierungsvorgänge realisiert werden (siehe auch Seite 109).

Hinweis Die Vorteile des isochron-Mode können auch eingeschränkt von Slaves (DP-V0-Slaves) genutzt werden, die den isochron-Mode nicht unterstützen. Man aktiviert dazu bei den Slaves *Sync* und/oder *Freeze* und ordnet sie der Gruppe 8 zu.
Typischerweise verwendet man den Sync- und Freeze -Mode gleichzeitig.

1.4.4.1 Isochron Mode Sync (ab DP-V2)

Der Isochron Sync Mode ermöglicht eine taktsynchrone Regelung in Master und Slave, und ein zeitgleiches aktivieren der Ausgänge mehrerer Slaves.

1.4.4.2 Isochrone Mode Freeze (ab DP-V2)

Der isochron Mode Freeze ermöglicht eine zeitgleiche Übernahme der Eingangsdaten mehrerer Slaves.

1.4.5 Richtwerte für verschiedene Übertragungsraten

Bei der Konfiguration des PROFIBUS-DP Master ist zu beachten, dass ein Teil der Parameter im Register *Zeiten* von der im Register *Allgemein* eingestellten Baudrate abhängt. Verwenden Sie für die erste (initial) Konfiguration die in Tabelle 6 angegebenen Richtwerte. In einem späteren Schritt werden die Werte optimiert.

	9,6k	19,2k	45,45k	93,75k	187,5k	500k	1,5M	3M	6M	12M
MinTsdr	11	11	11	11	11	11	11	11	11	11
MaxTsdr	60	60	400	60	60	100	150	250	450	800
Tsl bit time	100	100	640	100	100	200	300	400	600	1000
Tqui bit time	0	0	0	0	0	0	0	3	6	9
Tset bit time	1	1	95	1	1	1	1	4	8	16

Tabelle 6: Richtwerte für verschiedene Übertragungsraten

Alle Zeitangaben in Tabelle 6 sind in T_{bit} angegeben ($1T_{\text{bit}} = 1/[\text{bit/s}]$).

MinTsdr ist mindestens 11 T_{bit} lang, da ein Zeichen aus 11 Bits (1 Startbit, 1 Stoppbit, 1 Paritätsbit, 8 Datenbits) besteht.

Übertragungszeit für ein Zeichen		
Baudrate	$T_{\text{bit}} = 1/\text{Baudrate}$	Zeit
9600 bit/s	$1 / 9600 = 104,166 \mu\text{s}$	$11 \cdot 104,166 \mu\text{s} = 114,583 \text{ ms}$
6 Mbit/s	$1 / 6 \cdot 10^6 = 166,667 \text{ ns}$	$11 \cdot 166,667 \text{ ns} = 1,833 \mu\text{s}$

Tabelle 7: Übertragungszeit für ein Zeichen

1.4.5.1 Berechnung der Token-Umlaufzeit T_{tr}

Die minimale Token-Umlaufzeit T_{tr} kann abgeschätzt werden:

$$T_{tr_{min}} = n * (198 + T_1 + T_2) + b * 11 + 242 + T_1 + T_2 + T_{sl}$$

n	:	Anzahl aktiver Slaves
b	:	Anzahl E/A-Datenbytes der aktiven Slaves (Input plus Output)
T0		$35 + 2 * T_{set} + T_{qui}$
T1	:	Wenn $T_0 < MinT_{sdr}$: $T_1 = MinT_{sdr}$
	:	Wenn $T_0 > MinT_{sdr}$: $T_1 = T_0$
T2	:	Wenn $T_0 < MaxT_{sdr}$: $T_2 = MaxT_{sdr}$
	:	Wenn $T_0 > MaxT_{sdr}$: $T_2 = T_0$
Tsl	:	Slot Time Maximale Zeitspanne, in welcher der Master auf eine Antwort des Slave wartet
198	:	Zweimal Telegrammkopf des Telegramms mit variabler Länge (für Request und Response)
242	:	Global_Control, FDL_Status_Req und Token-Weitergabe

Hinweis Die Abschätzung der minimalen Token-Umlaufzeit $T_{tr_{min}}$ gilt nur, wenn folgende Bedingungen eingehalten sind:

- Es wird nur ein Master am Bus betrieben
- Es werden keine Sendungen wiederholt
- Es findet kein azyklischer Datenverkehr statt

Stellen Sie auf keinen Fall ein kleineres $T_{tr_{min}}$ ein als mit obiger Formel berechnet, da sonst eine fehlerfreie Kommunikation nicht mehr garantiert werden kann. Es wird empfohlen das Doppelte oder Dreifache des berechneten Wertes einzustellen.

1.4.5.2 Beispiel zur Berechnung der Token-Umlaufzeit T_{tr}

Folgende Konfiguration ist gegeben:

5 aktive Slaves

(n = 5)

20 E/A-Datenbytes pro Slave

(b = 100)

Die folgenden Zeitkonstanten für eine Übertragungsrate von 6 Mbit/s wurden aus der Tabelle 6 entnommen

MinTsdr	=	11	T_{bit}
MaxTsdr	=	450	T_{bit}
Tsl bit time	=	600	T_{bit}
Tqui bit time	=	6	T_{bit}
Tset bit time	=	8	T_{bit}

$$T_0 = 35 + 2 * T_{set} + T_{qui}$$

$$T_0 = 35 + 2 * 8 + 6$$

$$T_0 = 57 T_{bit}$$

Da $T_0 > \text{MinTsdr}$: $T_1 = 57 T_{bit}$

Da $T_0 < \text{MaxTsdr}$: $T_2 = 450 T_{bit}$

Die ermittelten Werte werden in die Formel für die minimale Token-Umlaufzeit eingesetzt:

$$T_{tr_{min}} = n * (198 + T_1 + T_2) + b * 11 + 242 + T_1 + T_2 + T_{sl}$$

$$T_{tr_{min}} = 5 (198 + 57 + 450) + 100 * 11 + 242 + 57 + 450 + 600$$

$$T_{tr_{min}} [T_{bit}] = 5974 T_{bit}$$

$$T_{tr_{min}} [\mu s] = 5974 T_{bit} * 166,67 \text{ ns} = 995,68 \mu s$$

Hinweis T_{tr} wird bei der Eingabe im Dialogfenster geprüft.
 Ist der vom Anwender eingetragene Wert T_{tr} kleiner als der vom Programm errechnete Wert, erfolgt eine Fehlermeldung in der Fehler-Status-Anzeige. Zusätzlich wird ein Mindestwert für T_{tr} vorgeschlagen.
 Sind *Isochron Mode Sync* oder *Isochron Mode Freeze* ausgewählt, wird die Zykluszeit vom Parameter *Min slave Intervall* vorgegeben. Die T_{tr} muss dann auf jeden Fall kleiner als das *Min slave Intervall* sein.
 Das Nicht einhalten dieser Bedingung im Iso-Mode führt zu einer Fehlermeldung.

1.5 PROFIBUS-DP Slave Kontextmenü

Das Kontextmenü PROFIBUS-DP Slave enthält die folgenden Menüfunktionen:

PROFIBUS-DP Slave
Signale verbinden
GSD Datei einlesen
Module einfügen
User Parameter bearbeiten
Validieren
Import
Export
Neu
Kopieren
Einfügen
Löschen
Drucken
Eigenschaften

Zum Verhalten der Standard Menüfunktionen Validieren, Neu, Import, Export, Kopieren, Einfügen und Drucken siehe Kapitel 1.4.2.

1.5.1 Menüfunktion *Signale verbinden*

Mit *Signale verbinden* wird das Dialogfenster *Signal-Zuordnungen* geöffnet, in welchem die Systemsignale *PNO Ident Nummer* und *Standard Diagnose* mit der Anwenderlogik verbunden werden können.

Name	Beschreibung	Typ
PNO Ident Nummer	Die von der PROFIBUS-DP Nutzerorganisation e.V. zugeteilte 16-Bit-Nummer, die ein Produkt (Feldgerät) eindeutig kennzeichnet.	WORD
Standard Diagnose	Über die Standard Diagnose teilt der Slave dem Master seinen aktuellen Zustand mit. Dieses Signal enthält immer die zuletzt empfangene Standard Diagnose. Die Parameter entsprechen dem Diagnosetelegramm gemäß IEC 61158.	DWORD

Tabelle 8: Dialog *Signal-Zuordnungen*

1.5.2 Menüfunktion *GSD Datei einlesen*

GSD Datei einlesen öffnet ein Standard-Dialogfenster zum Laden von Dateien. Diese Datei enthält wichtige Daten für die Parametrierung des PROFIBUS-DP Slave.

Hinweis Nicht alle GSD-Parameter sind für *ELOP II Factory* notwendig. Daher werden nicht alle GSD-Parameter in *ELOP II Factory* angezeigt.

Die GSD-Datei des *HIMatrix* PROFIBUS-DP Slave stellt die folgenden Module bereit:

PROFIBUS-DP Master Eingangs Module	Anzahl	Typ
DP-Input/ELOP-Export	1	Byte
DP-Input/ELOP-Export	2	Bytes
DP-Input/ELOP-Export	4	Bytes
DP-Input/ELOP-Export	8	Bytes
DP-Input/ELOP-Export	16	Bytes
DP-Input/ELOP-Export	1	Word
DP-Input/ELOP-Export	2	Words
DP-Input/ELOP-Export	4	Words
DP-Input/ELOP-Export	8	Words
DP-Input/ELOP-Export	16	Words
PROFIBUS-DP Master Ausgangs Module	Anzahl	Typ
DP-Output/ELOP-Import	1	Byte
DP-Output/ELOP-Import	2	Bytes
DP-Output/ELOP-Import	4	Bytes
DP-Output/ELOP-Import	8	Bytes
DP-Output/ELOP-Import	16	Bytes
DP-Output/ELOP-Import	1	Word
DP-Output/ELOP-Import	2	Words
DP-Output/ELOP-Import	4	Words
DP-Output/ELOP-Import	8	Words
DP-Output/ELOP-Import	16	Words

Tabelle 9: Module der HIMatrix GSD-Datei `hix100ea.gsd`

Siehe auch www.hima.com und www.PROFIBUS.com.

Hinweis Für die Richtigkeit der GSD-Datei ist der Hersteller des Feldgerätes verantwortlich.
GSD-Dateien sind ASCII-Dateien und können mit einem ASCII-Editor betrachtet werden.

1.5.3 Menüfunktion *Module einfügen*

Module einfügen öffnet das gleichnamige Dialogfenster, in welchem die verwendeten PROFIBUS-DP Module ausgewählt werden.

Im PROFIBUS-DP Master muss die Anzahl der tatsächlich zu übertragenden Bytes konfiguriert werden. Dies geschieht durch Auswahl von Modulen, die in der GSD-Datei des PROFIBUS-DP Slave definiert sind.

Um die Anzahl der Bytes für die Eingangs- und Ausgangssignale des PROFIBUS-DP Masters zu konfigurieren, wählt man mehrere Module, bis die physikalische Konfiguration des Slaves erreicht ist.

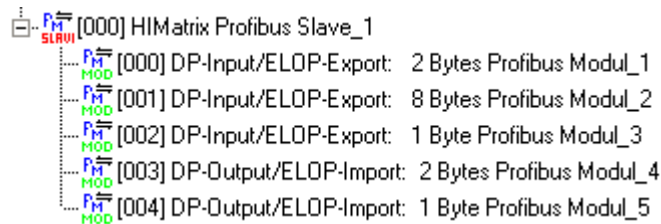


Bild 2: Passende Module aus der *HIMatrix* GSD-Datei für dieses Beispiel

Der Index der PROFIBUS-DP-Module muss in aufsteigender Reihenfolge und ohne Lücken nummeriert werden.

Die Reihenfolge der PROFIBUS-DP Module ist für die Funktion nicht von Bedeutung. Zur besseren Übersicht sollten die DP-Input Module und der DP-Output Module jedoch geordnet angelegt werden.

Hinweis	<p>Es ist nicht von Bedeutung, wie viele Module verwendet werden, um auf die erforderliche Anzahl an Bytes zu kommen, so lange die Anzahl von maximal 32 Modulen nicht überschritten wird.</p> <p>Um die Konfiguration des PROFIBUS-DP Master nicht unnötig zu erschweren, sollte die Zahl der gewählten Module möglichst klein gehalten werden. (Gilt für HIMA Slaves. Für Slaves andere Hersteller beachten Sie das Handbuch des Slaves.)</p>
----------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1.5.3.1 Signal Zuordnung in den PROFIBUS-DP Modulen

Wählen Sie im Kontextmenü des jeweiligen PROFIBUS-DP Moduls *Signale verbinden* um den Dialog *Signal-Zuordnungen* zu öffnen. Die Summe der Signale in Byte muss mit der Größe des jeweiligen Moduls in Byte übereinstimmen.

Signal Zuordnung in den Eingangsmodulen

In das Register *Eingänge* der Eingangsmodule werden die Signale eingetragen, die der Master vom Slave empfängt.

Signal Zuordnung in den Ausgangsmodulen

In das Register *Ausgänge* der Ausgangsmodule werden die Signale eingetragen, die der Master zum Slave sendet.

1.5.4 Menüfunktion *User-Parameter bearbeiten*

Die Menüfunktion *User-Parameter bearbeiten* öffnet das gleichnamige Dialogfenster.

In dem Benutzerdatenfeld werden die **Startadresse** und die **Anzahl der Signale** definiert. Die Anzahl der tatsächlich zu übertragenden Bytes wird durch die Auswahl von **Modulen** aus der GSD-Datei (siehe Kapitel 1.5.3) konfiguriert.

Aufbau des 32 Byte Benutzerdatenfelds

Das 32 Byte Benutzerdatenfeld ist wie folgt aufgebaut:

Die 32 Byte sind in **acht Blöcke** gruppiert, mit jeweils **vier Bytes per Block**.

Die Blöcke 1 ... 4 definieren welche und wie viele Signale der PROFIBUS-DP Master vom PROFIBUS-DP Slave empfängt.

Die Blöcke 5 ... 8 definieren welche und wie viele Signale der PROFIBUS-DP Master an den PROFIBUS-DP Slave sendet.

Die ersten beiden Bytes eines jeden Blocks spezifizieren die Startadresse für das erste zu lesende oder zu schreibende Signal.

Die letzten beiden Bytes eines jeden Blocks spezifizieren die Anzahl der Signale, die empfangen oder gesendet werden sollen.

32 Byte Benutzerdaten aufgeteilt in acht Blöcke		
Master Import/Slave Export	Startadresse	Anz. Signale
1. Block (Byte 0 bis 3)	0,0	0,0
2. Block (Byte 4 bis 7)	0,0	0,0
3. Block (Byte 8 bis 11)	0,0	0,0
4. Block (Byte 12 bis 15)	0,0	0,0
Master Export/Slave Import	Startadresse	Anz. Signale
5. Block (Byte 16 bis 19)	0,0	0,0
6. Block (Byte 20 bis 23)	0,0	0,0
7. Block (Byte 24 bis 27)	0,0	0,0
8. Block (Byte 28 bis 31)	0,0	0,0

Tabelle 10: 32 Bytes Benutzerdatenfeld im PROFIBUS-DP Master

1.5.5 Konfiguration der Benutzerdaten in verschiedenen Blöcken

Normalerweise ist es nicht notwendig, die Signale (Benutzerdaten) auf verschiedene Blöcke zu verteilen. Es ist vollkommen ausreichend nur den jeweils ersten Signalblock der Eingangs und Ausgangssignale zu definieren und die Daten 'en bloc' zu lesen oder zu schreiben.

In Anwendungen, in denen es jedoch erforderlich ist, nur ausgewählte Signale zu lesen oder zu schreiben, können bis zu je vier Signalblöcke für die Ausgangs- und Eingangssignale definiert werden.

Beispiel

Der PROFIBUS-DP Master sendet und empfängt die folgenden Signale vom PROFIBUS-DP Slave:

1. Block: **4** Eingangssignale ab der Startadresse **0**.
2. Block: **6** Eingangssignale ab der Startadresse **50**.
4. Block: **9** Eingangssignale ab der Startadresse **100**.
5. Block: **2** Ausgangssignale ab der Startadresse **10**.

Konfiguration der Benutzerdaten im PROFIBUS-DP Master

32 Byte Benutzerdaten aufgeteilt in acht Blöcke		
Master Import/Slave Export	Startadresse	Anz. Signale
1. Block (Byte 0 bis 3)	0,0	0,4
2. Block (Byte 4 bis 7)	0,50	0,6
3. Block (Byte 8 bis 11)	0,0	0,0
4. Block (Byte 12 bis 15)	0,100	0,9
Master Export/Slave Import	Startadresse	Anz. Signale
5. Block (Byte 16 bis 19)	0,10	0,2
6. Block (Byte 20 bis 23)	0,0	0,0
7. Block (Byte 24 bis 27)	0,0	0,0
8. Block (Byte 28 bis 31)	0,0	0,0

Tabelle 11: 32 Bytes Benutzerdaten eines PROFIBUS-DP Slave

Die Ansicht des Dialogs *User Parameter bearbeiten* ist von der geladenen GSD-Datei abhängig. Mehr Komfort bei der Eingabe der Benutzerdaten bietet z.B. der HIMatrix PROFIBUS-DP Slave (siehe Kapitel 3.5.2.3).

Im HIMatrix PROFIBUS-DP Master sind dann die folgenden Benutzerdaten im Dialog *User Parameter bearbeiten* einzutragen.

	Name	Wert	Standard
1	Unref.Datenbyte[0]	0	Standard
2	Unref.Datenbyte[1]	0	Standard
3	Unref.Datenbyte[2]	0	Standard
4	Unref.Datenbyte[3]	4	Standard
5	Unref.Datenbyte[4]	0	Standard
6	Unref.Datenbyte[5]	50	Standard
7	Unref.Datenbyte[6]	0	Standard
8	Unref.Datenbyte[7]	6	Standard
9	Unref.Datenbyte[8]	0	Standard
10	Unref.Datenbyte[9]	0	Standard
11	Unref.Datenbyte[10]	0	Standard
12	Unref.Datenbyte[11]	0	Standard
13	Unref.Datenbyte[12]	0	Standard
14	Unref.Datenbyte[13]	100	Standard
15	Unref.Datenbyte[14]	0	Standard
16	Unref.Datenbyte[15]	9	Standard
17	Unref.Datenbyte[16]	0	Standard
18	Unref.Datenbyte[17]	10	Standard
19	Unref.Datenbyte[18]	0	Standard
20	Unref.Datenbyte[19]	2	Standard

Buttons: OK, Abbrechen, Übernehmen, Standard, Hilfe

Bild 3: 32 Bytes Benutzerdaten eines PROFIBUS-DP Slave

1.5.6 Menüfunktion Validieren

Vor der Codegenerierung kann die Parametrierung des Masters und der Slaves getestet werden. In der Strukturbaum wird der PROFIBUS-DP Slave selektiert und im Kontextmenü wird *Validieren* gewählt. In der Fehler-Status-Anzeige werden dann eventuelle Fehler und Warnungen angezeigt.

Die Validation wird zudem automatisch vor jeder Codegeneration durchgeführt. Wird bei der Validation ein Fehler festgestellt, dann wird die Codegeneration abgebrochen.

1.5.7 Menüfunktion *Eigenschaften*

Mit *Eigenschaften* wird das Dialogfenster *Eigenschaften* geöffnet.

Das Dialogfenster enthält die Register:

- Parameter
- Gruppen
- DP-V1
- Alarme
- Daten
- Modell
- Features
- Übertragungsraten
- Azyklisch

1.5.7.1 Register *Parameter*

Element	Beschreibung	Wert
Name	Name des Slave	Eindeutig pro Master
Adresse	Adresse des Slave	Min: 0 Max: 125 Standard: 125
Aktiv	Zustand des Slave Nur ein aktiver Slave kann mit einem PROFIBUS-DP Master kommunizieren.	Standard: TRUE
DP-V0 Sync aktiv	Der Sync Mode ermöglicht ein zeitgleiches aktivieren der Ausgänge mehrerer DP-V0-Slaves. Achtung Bei DP-V2-Slaves, die im Isochron Mode Sync arbeiten, muss dieses Feld deaktiviert (FALSE) sein.	Standard: FALSE
DP-V0 Freeze aktiv	Der Freeze Mode ermöglicht eine Zeitgleiche Übernahme der Eingangsdaten mehrerer DP-V0-Slaves. Achtung Bei DP-V2-Slaves, die im Isochron Mode Freeze arbeiten, muss dieses Feld deaktiviert (FALSE) sein.	Standard: FALSE
Watchdog aktiv	Ist <i>Watchdog aktiv</i> ausgewählt, dann können der Master und der HIMatrix Slave eine Unterbrechung der PROFIBUS-DP Kommunikation erkennen und darauf reagieren (siehe auch Kapitel 1.2 und 3.2). Um auch eine Watchdogzeit eingeben zu können, muss die Schaltfläche Übernehmen betätigt werden.	Standard: FALSE
Watchdog-Zeit [ms]	Richtwert: Watchdog-Zeit des Slave > 6 * T _{tr}	Min: 0 Max: 65535 Standard: 0

Bei Ausfall letzte Daten senden	FALSE: Verbindung wird im Fehlerfall abgebaut und neu aufgebaut. TRUE: Sendet im Fehlerfall weiter Daten auch ohne Bestätigung des Slaves.	Standard: FALSE
Auto-Clear bei Ausfall	Ist Auto-Clear bei Ausfall im Master und bei diesem Slave auf TRUE gesetzt, dann schaltet der Master bei einem Ausfall dieses Slaves den kompletten PROFIBUS-DP in den sicheren Zustand.	Standard: FALSE

Tabelle 12: Parameter des Slave

1.5.7.2 Register Gruppen

In diesem Register können die Slaves in verschiedenen Gruppen organisiert werden. Die Global Control-Kommandos Sync und Freeze können dann gezielt eine oder mehrere Gruppen ansprechen. Praktische Bedeutung hat das aber nur noch im Isochron Mode, der immer die Gruppe 8 adressiert.

Element	Beschreibung	Wert
Mitglied in Gruppe 1	Mitglied in Gruppe 1	Standard: FALSE
Mitglied in Gruppe 2	Mitglied in Gruppe 2	
Mitglied in Gruppe 3	Mitglied in Gruppe 3	
Mitglied in Gruppe 4	Mitglied in Gruppe 4	
Mitglied in Gruppe 5	Mitglied in Gruppe 5	
Mitglied in Gruppe 6	Mitglied in Gruppe 6	
Mitglied in Gruppe 7	Mitglied in Gruppe 7	
Mitglied in Gruppe 8	Mitglied in Gruppe 8	

Tabelle 13: Gruppen des Slave

1.5.7.3 Register DP-V1

In diesem Register befinden sich Parameter, die erst ab DP-V1 definiert sind. Bei DP-V0-Slaves kann hier nichts ausgewählt werden. Welche Parameter vom Slave unterstützt werden, erkennt man in der Spalte *Unterstützt*, werden Parameter zwingend gefordert, so ist in der Spalte *Verlangt* ein Häkchen.

Element	Beschreibung	Wert
DP-V1	Wenn der DP-V1-Modus nicht aktiviert ist, können auch die anderen DP-V1-Features nicht genutzt werden. Der Slave verhält sich dann wie ein DP-V0-Slave. Eventuell müssen dann auch die Parametrierdaten geändert werden (siehe Handbuch des Slaves)	Standard: FALSE
Failsafe	Wenn dieser Modus aktiviert ist, sendet der Master im Zustand CLEAR keine Nullen als Ausgangsdaten zum Slave, sondern ein leeres Datenpaket (Failsafe-Paket). Der Slave erkennt daran, dass er jetzt die sicheren Ausgangsdaten (die nicht notwendigerweise alle Null sind) auf die Ausgänge legen soll.	Standard: FALSE
Isochron Modus	Diese Funktion ermöglicht eine taktsynchrone Regelung in Master und Slave unabhängig von der Belastung des Busses. Der Buszyklus wird mit einer Taktabweichungen von $< 1\mu s$ synchronisiert. Damit können hochgenaue Positionierungsvorgänge realisiert werden.	Standard: FALSE
Publisher aktiv	Diese Funktion wird für den Slave-Querverkehr benötigt. Dies ermöglicht die direkte und zeitsparende Kommunikation zwischen den Slaves via Broadcast ohne Umwege über den Master. Dieses Feld muss aktiv sein, wenn der Slave als Publisher Daten an die Subscriber Slaves senden soll. (Siehe auch Register <i>Features Subscriber</i>).	Standard: FALSE
Prm Block Struct. Supp.	Der Slave unterstützt strukturierte Parametrierdaten (Nur Lesen).	Standard: FALSE
Check. Cfg.-Mode	Reduzierte Konfigurationskontrolle, wenn Check Cfg.-Mode aktiviert ist, dann kann der Slave ohne die komplette Konfiguration betrieben werden. Für die Inbetriebnahme sollte dieses Feld deaktiviert werden.	Standard: FALSE

Tabelle 14: Register DP-V1 des Slaves im PROFIBUS-DP Master

1.5.7.4 Register *Alarme*

Auf dieser Seite können Alarme aktiviert werden. Das geht jedoch nur bei DP-V1-Slaves, wenn DP-V1 aktiviert ist und der Slave Alarme unterstützt. Welche Alarme unterstützt werden, erkennt man an dem Häkchen in der Spalte Unterstützt. Wird ein Alarm vorgeschrieben, erkennt man dies in der Spalte Verlangt.

Element	Beschreibung	Wert
Update-Alarm	Alarm, wenn Parameter eines Moduls geändert wurden.	Standard: FALSE
Status-Alarm	Alarm, wenn sich der Zustand eines Moduls geändert hat.	
Vendor-Alarm	Herstellerspezifischer Alarm.	
Diagnose-Alarm	Alarm, wenn bestimmte Ereignisse wie Kurzschluss, Übertemperatur, etc. an einem Modul auftreten.	
Prozess-Alarm	Alarm, wenn wichtige Ereignisse im Prozess auftreten.	
Stecken/Ziehen Alarm	Alarm, wenn eine Baugruppe gezogen oder aufgesteckt wird.	

Tabelle 15: Register *Alarme* des Slave

1.5.7.5 Register *Daten*

In diesem Register befinden sich Informationen über die unterstützten Datenlängen, sowie über die Benutzerdaten (erweiterte Parametrierdaten).

Element	Beschreibung	Wert
Max. Input Len. [Byte]	Maximale Länge der Eingangsdaten.	Nur Anzeige
Max. Output Len. [Byte]	Maximale Länge der Ausgangsdaten.	Nur Anzeige
Max. Data Len. [Byte]	Maximale Gesamtlänge der Ein- und Ausgangsdaten.	Nur Anzeige
Benutzerdatengröße [Byte]	Länge der Benutzerdaten.	Nur Anzeige
Benutzerdaten	Die erweiterten Parametrierdaten, die an den Slave gesendet werden. Das Editieren empfiehlt sich nicht hier durchzuführen. Komfortabler geht es mit dem Befehl <i>User Parameter bearbeiten</i> .	Benutzergröße [Byte]
Max. Diag. Data Len.	Maximale Länge der Diagnosedaten, die der Slave sendet.	Nur Anzeige

Tabelle 16: Register *Daten* des Slaves

1.5.7.6 Register *Modell*

Auf dieser Seite befinden sich verschiedene Informationen, die selbsterklärend sind.

Element	Beschreibung	Wert
Modell	Herstellerbezeichnung des PROFIBUS-DP Slave	Nur Anzeige
Hersteller	Hersteller des Feldgerätes	
Ident. Nummer	Slave-Kennung der PROFIBUS Nutzerorganisation (PNO)	
Revision	Ausgabestand des PROFIBUS-DP Slave	
Hardware-Release	Hardware-Ausgabestand des PROFIBUS-DP Slave	
Software-Release	Software-Ausgabestand des PROFIBUS-DP Slave	
GSD-Dateiname	Dateiname der GSD-Datei	
Infotext	Zusätzliche Info zum PROFIBUS-DP Slave	

Tabelle 17: Register *Modell* des Slave

1.5.7.7 Register *Features*

Element	Beschreibung	Wert
Modularstation	TRUE: Modularstation. FALSE: Kompaktstation	Nur Anzeige
Erste parametrierbare Slot-nummer	Die Nummerierung der Module (Slots) muss mit dieser Nummer beginnen und fortlaufend erfolgen.	
Max. Module	Maximale Anzahl an Modulen, die eine modulare Station aufnehmen kann.	
Unterstützung für ‚Set Slave Add‘	Der Slave unterstützt dynamische Adressvergabe.	
Min. Slave Interval [ms]	Die Mindestzeit, die zwischen zwei zyklischen Aufrufen des Slaves verstreichen muss.	
Diag. Update	Anzahl Pollzyklen, die verstreichen kann, bis die Diagnose des Slaves den aktuellen Zustand widerspiegelt.	
Unterstützung für WDBase1ms	Der Slave unterstützt 1ms als Zeitbasis für die Watchdog.	
Unterstützung für DP-V0 Sync	Der Slave unterstützt DP-V0 Sync	
Unterstützung für DP-V0 Freeze	Der Slave unterstützt DP-V0 Freeze	

DP-V1 Datentypen	Der Slave unterstützt die DP-V1-Datentypen.	
Extra Alarm SAP	Der Slave unterstützt SAP 50 zur Alarmbestätigung.	
Anzahl paralleler, aktiver Alar- me	Gibt an, wie viele aktive Alar- me der Slave gleichzeitig be- arbeiten kann. Null bedeutet <i>ein Alarm von jedem Typ.</i>	

Tabelle 18: Register *Features* des Slaves im HIMatrix Master

1.5.7.8 Register *Übertragungsraten*

In diesem Register befinden sich die *Übertragungsraten*, die der Slave unterstützt, sowie jeweils die zugehörige *MaxTsd*.

MaxTsd ist die Zeit, innerhalb welcher der Slave spätestens auf eine Anforderung vom Master antworten muss. Der Wertebereich ist abhängig vom Slave und der Übertragungs-
geschwindigkeit und liegt zwischen 15 und 800 T_{bit}.

Element	Beschreibung	Wert
9,6k	MaxTsd = 15	Nur Anzeige
19,2k	MaxTsd = 15	
31,25k	Wird nicht unterstützt	
45,45k	Wird nicht unterstützt	
93,75k	MaxTsd = 15	
187,5k	MaxTsd = 15	
500k	MaxTsd = 15	
1,5M	MaxTsd = 25	
3M	MaxTsd = 50	
6M	MaxTsd = 100	
12M	MaxTsd = 200	

Tabelle 19: Register *Übertragungsraten* eines Slaves im HIMA Master

1.5.7.9 Register *Azyklisch*

In diesem Register befinden sich einige Parameter für die azyklische Datenübertragung.

Element	Beschreibung	Wert
C1 Read/Write Unterstützung	Der Slave unterstützt azykli- sche Datenübertragung.	Nur Anzeige
C1 Read/Write notwendig	Der Slave erfordert azyklische Datenübertragung.	
C1 Datengröße [Byte]	Maximale Länge eines azykli- schen Datenpaketes.	
C1 Response Timeout [ms]	Timeout für azyklische Daten- übertragung.	

Tabelle 20: Register *Acyclic* des Slaves im HIMA Master

1.6 Diagnose und Protokollzustände des HIMatrix PROFIBUS-DP Master

Im Register *ProfibusMs.* kann der Anwender die Einstellungen des Masters und der Slaves überprüfen. Zudem werden aktuelle Statusinformationen (z.B. Zykluszeit) des Masters und der Slaves angezeigt.

Öffnen Sie im Hardware-Management das Control Panel und wählen Sie das Register *ProfibusMs.*

Das Register *ProfibusMs.* ist in die Bereiche, *PB Master* und *PB Slaves* unterteilt.

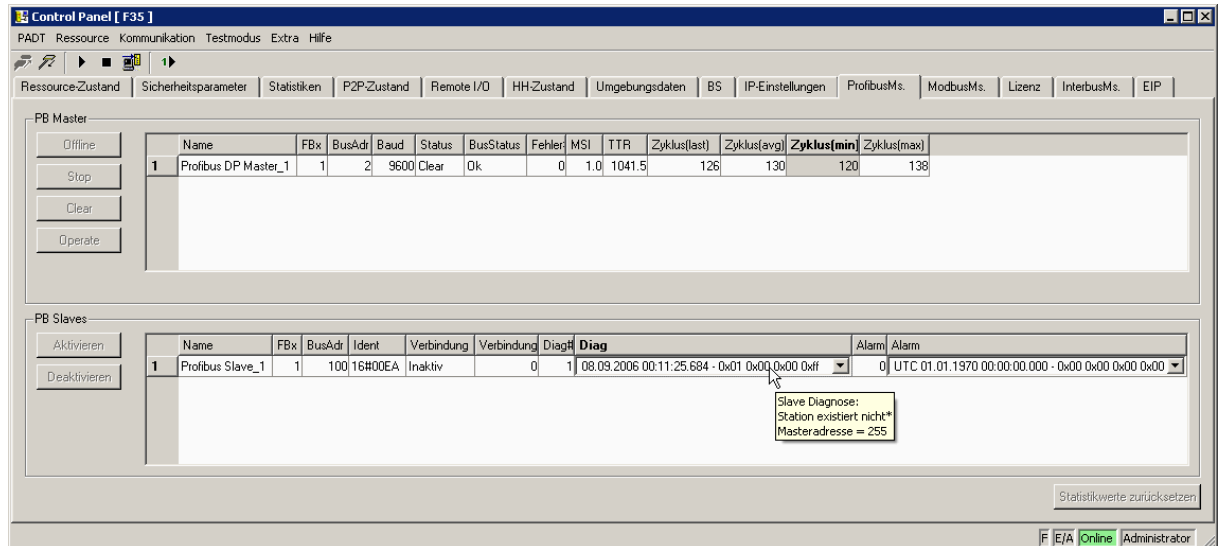


Bild 4: 32 Anzeige des PROFIBUS-DP im Control-Panel

1.6.1 Statistikwerte zurücksetzen:

Die Schaltfläche **Statistikwerte zurücksetzen** setzt die statistischen Daten (Zykluszeit min, max usw.) auf null zurück.

1.6.2 PB Master

Im Bereich *PB Master* kann der Anwender die Einstellungen überprüfen und den Master steuern. Zudem werden aktuelle Statusinformationen (z.B. Zykluszeit) des Masters angezeigt.

Dazu stehen vier Schaltflächen und ein Anzeigefeld zur Verfügung.

1.6.2.1 Schaltflächen

Mit den Schaltflächen können die folgenden Kommandos auf einen oder mehrere selektierte PROFIBUS_DP Master angewendet werden:

Offline:

Schaltet den selektierten PROFIBUS-DP Master aus. Ist der Master ausgeschaltet, erfolgen keine Aktivitäten.

Stop:

Stoppt den selektierten PROFIBUS-DP Master. Der PROFIBUS_DP Master nimmt weiterhin am Token-Protokoll teil, sendet aber keine Daten an die Slaves.

Clear:

Durch Betätigen der Schaltfläche *CLEAR* wird der selektierte PROFIBUS_DP Master in einen sichereren Zustand gesetzt und tauscht nun sichere Daten mit den Slaves aus. Die Ausgangsdaten, die zu den Slaves gesendet werden, enthalten nur Nullen. FailSafe-Slaves erhalten FailSafe-Telegramme, die keine Daten enthalten. Die Eingangsdaten von den Slaves werden vom PROFIBUS_DP Master ignoriert und stattdessen Initialwerte im Anwenderprogramm verwendet.

Operate:

Startet den selektierten PROFIBUS_DP Master. Der PROFIBUS_DP Master tauscht zyklisch E/A-Daten mit den Slaves aus.

1.6.2.2 Anzeigefeld

In dem Anzeigefeld werden die Zustände der PROFIBUS-DP Master angezeigt, die an den Bus angeschlossen sind.

Element	Beschreibung
Name	Name des PROFIBUS-DP Master
FBx	Zugeordnete Feldbusschnittstelle (1, 2)
BusAdr	Busadresse des Master (0 bis 125)
Baud	Baudrate des Master Der Master kann mit allen Baudraten, die im Standard spezifiziert sind kommunizieren. Zykluszeiten sind bis zu einer Untergrenze von 2 ms möglich.
Status	Zeigt den momentanen Protokollzustand an. 0: OFFLINE 1: STOP 2: CLEAR 3: OPERATE 7: UNDEFINED
BusStatus	Fehlercode Busfehler: 0: OK 1: Adressfehler die Adresse des Masters ist auf dem Bus bereits vorhanden. 2: Busstörung es wurde eine Störung auf dem Bus registriert, z.B. Bus nicht richtig abgeschlossen, mehrere Teilnehmer senden gleichzeitig. 3: Protokollfehler ein fehlerhaft codiertes Paket wurde empfangen. 4: Hardwarefehler die Hardware hat einen Fehler gemeldet, z.B. bei zu knapp eingestellten Zeiten. 5: Unbekannter Fehler Master hat Zustand aus unbekanntem Grund gewechselt. 6: Controller Reset Bei schweren Busstörungen bleibt mitunter der Controller-Chip hängen und wird zurückgesetzt. Der Fehlercode bleibt solange anliegen, bis der Busfehler behoben ist.

Tabelle 21: Anzeigefeld

Element	Beschreibung
Fehler#	Anzahl der Busfehler bisher
MSI	Min. Slave Interval in ms, Auflösung 0.1ms
Ttr	Target Rotation Time in ms, Auflösung 0.1ms
Zyklus[last]	Letzte PROFIBUS-DP Zykluszeit [ms]
Zyklus[avg]	Mittlere PROFIBUS-DP Zykluszeit [ms]
Zyklus[min]	Minimale PROFIBUS-DP Zykluszeit [ms]
Zyklus[max]	Maximale PROFIBUS-DP Zykluszeit [ms]

Tabelle 22: Anzeigefeld des PB Master

1.6.3 PB Slaves

Im Bereich *PB Slaves* kann der Anwender die Einstellungen überprüfen und die Slaves steuern. Zudem werden aktuelle Diagnosen und Alarme des Slaves angezeigt.

Dazu stehen zwei Schaltflächen und ein Anzeigefeld zur Verfügung.

1.6.3.1 Schaltflächen

Mit den Schaltflächen können die selektierten Slaves aktiviert oder deaktiviert werden.

Aktivieren:

Aktiviert den selektierten Slave, der mit einem PROFIBUS-DP Master nun Daten austauschen kann.

Deaktivieren:

Deaktiviert den selektierten Slave.

Die Kommunikation wird beendet und zum Anwenderprogramm werden die Initialdaten gesendet.

1.6.3.2 Anzeigefeld

In dem Anzeigefeld werden die Zustände der PROFIBUS-DP Slaves angezeigt, die an den Bus angeschlossen sind.

Element	Beschreibung
Name	Name des PROFIBUS-DP Slave
FBx	Zugeordnete Feldbusschnittstelle des Slave
BusAdr	Busadresse des Slave
Ident	PNO-Identifikationsnummer
Verbindung	Verbindungszustand 0: Deaktiviert, 1: Inaktiv (versucht Verbindung aufzunehmen), 2: Verbunden
Verbindung#	Anzahl bisheriger Verbindungsaufnahmen.
Diag#	Anzahl bisheriger Diagnosemeldungen.
Diag	Erste vier Bytes der letzten Diagnosemeldung mit Zeitstempel. Es öffnet sich ein Popup-Fenster, wenn die Maus über das Feld geführt wird.
Alarm#	Anzahl bisheriger Alarme.
Alarm	Erste vier Bytes der letzten Alarmmeldung mit Zeitstempel. Es öffnet sich ein Popup-Fenster, wenn die Maus über das Feld geführt wird.

Tabelle 23: Anzeigefeld des Slave

1.6.4 Protokollzustände des HIMatrix PROFIBUS-DP Master

Der Protokollzustand wird im Control Panel (siehe Tabelle 22) angezeigt und kann mit dem Statussignal *Master-Status* (siehe Tabelle 1) im Anwenderprogramm ausgewertet werden.

Master Zustand	Beschreibung
OFFLINE	Der Master ist ausgeschaltet und es erfolgen keine Busaktivitäten.
STOPP	Der Master nimmt am Token-Protokoll teil, sendet aber keine Daten an die Slaves.
CLEAR	Der Master ist im sicheren Zustand und tauscht sichere Daten mit den Slaves aus. <ul style="list-style-type: none"> Die Ausgangsdaten, die zu den Slaves gesendet werden, enthalten nur Nullen. Die FailSafe-Slaves erhalten FailSafe-Telegramme (diese enthalten keine Daten). Die Eingangsdaten von den Slaves werden ignoriert und stattdessen werden Initialwerte verwendet.
OPERATE	Der Master ist im Arbeitsmodus und tauscht zyklisch E/A-Daten mit den Slaves aus.

Tabelle 24: Protokollzustände des HIMatrix PROFIBUS-DP Masters

1.6.5 Verhalten des HIMatrix PROFIBUS-DP Master

Verhalten des HIMatrix PROFIBUS-DP Master in Abhängigkeit vom Betriebszustand der Steuerung.

Steuerung Zustand	Verhalten des HIMatrix PROFIBUS-DP Master
STOPP *)	Ist die Steuerung in STOPP, dann ist der Master im Zustand OFFLINE.
RUN	Ist die Steuerung in RUN, versucht der Master, in den Zustand OPERATE zu gelangen.
STOPP	Geht die Steuerung in STOPP, geht der Master in den Zustand CLEAR. Ist der Master bereits in STOPP oder OFFLINE, bleibt er in diesem Zustand.

Tabelle 25: Verhalten des HIMatrix PROFIBUS-DP Master

*)Nach Einschalten der Steuerung oder nach Laden der Konfiguration

1.6.6 Funktion der FBx LED beim PROFIBUS Master

Der Zustand der seriellen PROFIBUS-DP Kommunikation wird mit der FBx LED der jeweiligen konfigurierten seriellen Schnittstellen (fb1, fb2) angezeigt.

FBx LED	Beschreibung
AUS	Der PROFIBUS-DP Master ist nicht oder ungültig konfiguriert.
Blinkt im 2 Sekundentakt	Der PROFIBUS-DP Master befindet sich im Zustand OFFLINE oder STOPP.
AN	Der PROFIBUS-DP Master befindet sich im Zustand OPERATE oder CLEAR und tauscht Daten mit allen aktivierten Slaves aus.
Blinkt im Sekundentakt	Mindestens ein Slave ist ausgefallen.

Tabelle 26: Funktion der FBx LED beim PROFIBUS Master

1.7 Beispiel: Konfiguration des HIMatrix PROFIBUS-DP Master

In diesem Beispiel arbeitet eine HIMatrix Steuerung als PROFIBUS-DP Master und kommuniziert mit einer Siemens ET 200M, die als PROFIBUS-DP Slave arbeitet.

Dabei werden von der HIMatrix Steuerung 2 Bytes gelesen und 1 Byte gesendet.

Die Konfiguration der PROFIBUS-DP Signalverbindungen erfolgt in diesem Beispiel (mit der Siemens GSD-Datei) komplett auf der HIMatrix Steuerung.

Auf der ET 200M ist die folgenden Einstellung (z.B. über DIP-Schalterblock) vorzunehmen: Busadresse = 3.

Hinweis Die beschriebene Konfiguration der Siemens-Steuerung erhebt keinen Anspruch auf Vollständigkeit.
Alle Angaben sind ohne Gewähr, maßgebend zur Projektierung der Siemens ET 200M ist die Dokumentation von Siemens.

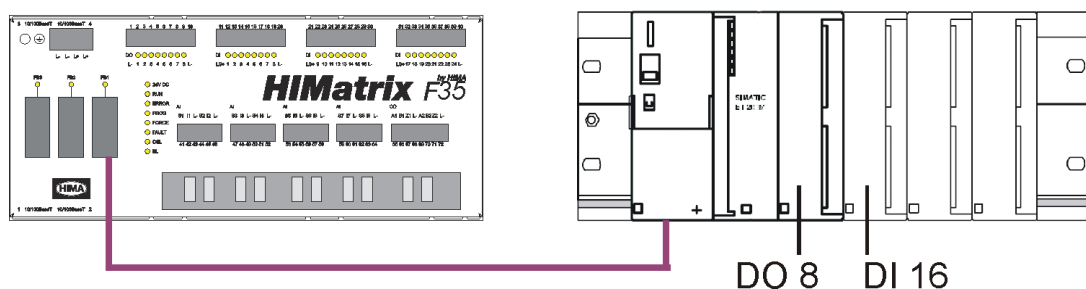


Bild 5: Eine HIMatrix (als Master) mit einer Siemens ET 200M (als Slave)

Starten Sie **ELOP II Factory** und erstellen Sie ein neues Projekt, oder laden Sie ein vorhandenes Projekt.

Schritt 1: Erstellen Sie einen PROFIBUS-DP Master

- ☐ Wählen Sie im Strukturbaum der Ressource **Protokolle**.
- ☐ Wählen Sie aus dem Kontextmenü **Neu, Profibus Master**.

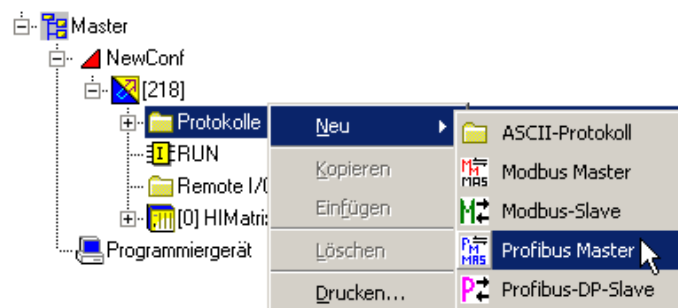


Bild 6: Anlegen des PROFIBUS-DP Master

Schritt 2: Fügen Sie dem PROFIBUS-DP Master einen PROFIBUS-DP Slave hinzu:

- ❑ Öffnen Sie im Strukturbaum das Verzeichnis **Protokolle**.
- ❑ Klicken Sie rechts auf **PROFIBUS-DP Master** und wählen Sie **Neu, PROFIBUS-DP Slave** aus dem Kontextmenü.

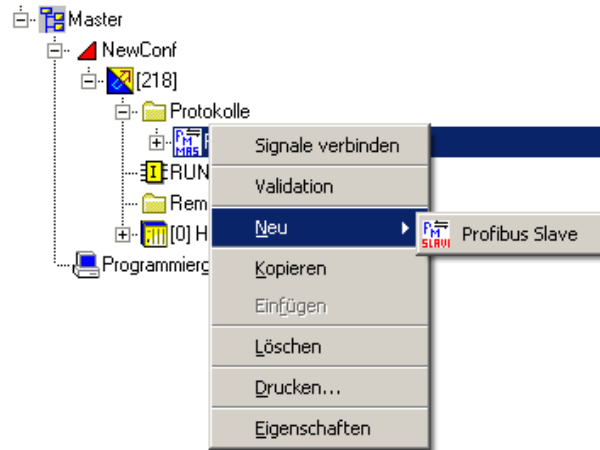


Bild 7: Anlegen des PROFIBUS-DP Slave

Schritt 3: Konfigurieren Sie den PROFIBUS-DP Slave:

- ❑ Öffnen Sie das Kontextmenü *PROFIBUS-DP Slave*.
- ❑ Wählen Sie **GSD Datei einlesen** aus dem Kontextmenü des Slaves.
- ❑ Wählen Sie aus dem Windows Standarddialog zum Öffnen einer Datei die GSD Datei für das Kompaktgerät aus (z.B. für die ET 200M `si02801e.gsd`)

Schritt 4: PROFIBUS-DP Adresse des Slaves eintragen:

- ❑ Öffnen Sie das Kontextmenü des Slaves und wählen Sie **Eigenschaften**.
- ❑ Öffnen Sie das Register **Parameter**.
- ❑ Geben Sie in das Eingabefeld *Adresse* die PROFIBUS Adresse des Slave 1 ein.

Schritt 5: Legen Sie die folgenden Module an:

- ❑ Wählen Sie **Module einfügen** aus dem Kontextmenü des Slaves.
- ❑ Wählen Sie aus dem Dialog *Module einfügen* das Modul **Config for Slot1** (Standard für die ET200M).
- ❑ Wählen Sie aus dem Dialog *Module einfügen* das Modul **Config for Slot2** (Standard für die ET200M).
- ❑ Wählen Sie aus dem Dialog *Module einfügen* das Modul **Config for Slot3** (Standard für die ET200M).
- ❑ Wählen Sie aus dem Dialog *Module einfügen* das Modul **6ES7 322-8BF00-0AB0 8DO**.
- ❑ Wählen Sie aus dem Dialog *Module einfügen* das Modul **6ES7 321-7BH01-0AB0 16DI**.

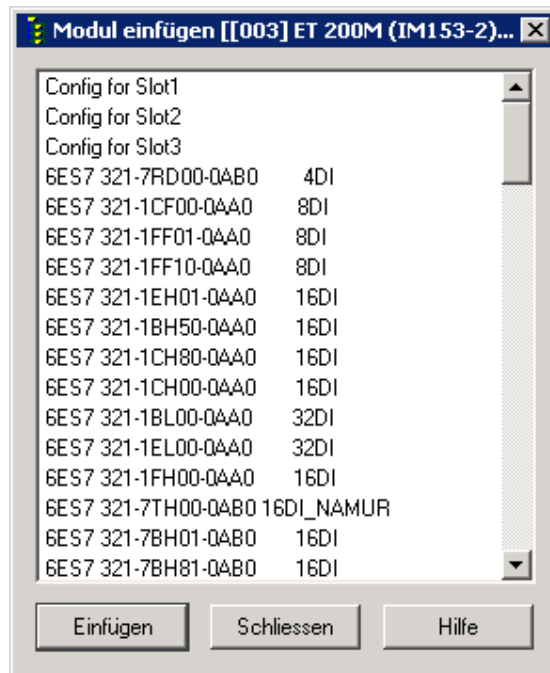


Bild 8: Dialog *Module einfügen*

Schritt 6: Nummerieren Sie die Module:

- ☐ Öffnen Sie den Dialog **Eigenschaften** vom ersten Modul.
- ☐ Tragen Sie **1** im Feld *Steckplatz* ein.
- ☐ Wiederholen Sie das mit den nächsten Modulen und vergeben Sie fortlaufende Steckplatznummern.

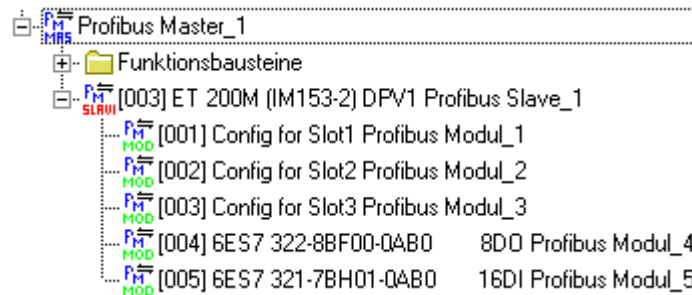


Bild 9: Die PROFIBUS-DP Module des Kompaktgeräts *Siemens ET 200M*

Hinweis Die Steckplatznummern der PROFIBUS-DP Module müssen fortlaufend nummeriert werden.

Schritt 7: Ändern Sie die Steckplatznummer von *8DO PROFIBUS Modul_4*:

- ❑ Öffnen Sie das Kontextmenü von *8DO PROFIBUS Modul_4*.
- ❑ Wählen Sie **User Parameter bearbeiten**.
- ❑ Tragen Sie **4** im Eingabefeld *SlotNumber* ein.

	Name	Wert	Default
1	[SlotNumber]	4	Default
2	Diag:Kurzschluss nach M Kanal0	Nein	Default
3	Diag:Kurzschluss nach M Kanal1	Nein	Default
4	Diag:Kurzschluss nach M Kanal2	Nein	Default
5	Diag:Kurzschluss nach M Kanal3	Nein	Default
6	Diag:Kurzschluss nach M Kanal4	Nein	Default
7	Diag:Kurzschluss nach M Kanal5	Nein	Default
8	Diag:Kurzschluss nach M Kanal6	Nein	Default
9	Diag:Kurzschluss nach M Kanal7	Nein	Default
10	Diag:Kurzschluss nach P Kanal0	Nein	Default
11	Diag:Kurzschluss nach P Kanal1	Nein	Default
12	Diag:Kurzschluss nach P Kanal2	Nein	Default
13	Diag:Kurzschluss nach P Kanal3	Nein	Default
14	Diag:Kurzschluss nach P Kanal4	Nein	Default

Bild 10: User Parameter von 8DO PROFIBUS Modul_4

Schritt 8: Ändern Sie die Slot-Nummer von *16DI PROFIBUS Modul_5*:

- ❑ Öffnen Sie das Kontextmenü von *16DI PROFIBUS Modul_5*.
- ❑ Wählen Sie **User Parameter bearbeiten**.
- ❑ Tragen Sie **5** im Eingabefeld *SlotNumber* ein.

	Name	Wert	Default
1	[SlotNumber]	5	Default
2	Eingangsverzögerung (ms)	3 (DC)	Default
3	Diag:Gebersversorgung 1	Nein	Default
4	Diag:Gebersversorgung 2	Nein	Default
5	Diag:Drahtbruch Kanal 0	Nein	Default
6	Diag:Drahtbruch Kanal 1	Nein	Default
7	Diag:Drahtbruch Kanal 2	Nein	Default
8	Diag:Drahtbruch Kanal 3	Nein	Default
9	Diag:Drahtbruch Kanal 4	Nein	Default
10	Diag:Drahtbruch Kanal 5	Nein	Default
11	Diag:Drahtbruch Kanal 6	Nein	Default
12	Diag:Drahtbruch Kanal 7	Nein	Default
13	Prozessalarmfreigabe	Nein	Default
14	Diagnosealarmfreigabe	Nein	Default

Bild 11: User Parameter von 16DI PROFIBUS Modul_5

Schritt 9: Erstellen Sie die folgenden Signale im Signaleditor

- ❑ Öffnen Sie über das Hauptmenü **Signale, Editor** den *Signaleditor*.
- ❑ Erstellen Sie die Signale **ET200M_F35_1** und **ET200M_F35_2** vom Typ Byte.
- ❑ Erstellen Sie das Signal **F35_ET200M_1** vom Typ Byte.

Schritt 10: Eingangssignale der HIMatrix mit den Signalen des Slaves *ET 200M* verbinden:

- ❑ Öffnen Sie das Kontextmenü des PROFIBUS-DP Moduls *16DI PROFIBUS Modul_5*.
- ❑ Wählen Sie aus dem Kontextmenü **Signale Verbinden**.
- ❑ Öffnen Sie das Register **Eingang** des Dialogfensters *Signal Zuordnungen*.
- ❑ Öffnen Sie über das Hauptmenü **Signale, Editor** den *Signaleditor*.
- ❑ Ziehen Sie die Signale **ET200M_F35_1** und **ET200M_F35_2** per Drag & Drop aus dem *Signaleditor* auf das jeweilige **Eingangssignal** im Register *Eingang* des Dialogfensters *Signal Zuordnungen*.
- ❑ Klicken Sie auf die Schaltfläche **Neue Offsets** im Dialog *Signal Zuordnungen*.
- ❑ Klicken Sie im Dialog *Offsets nummerieren* auf die Schaltfläche **Nummerieren**.
- ❑ Schließen Sie das Dialogfenster mit **OK**.

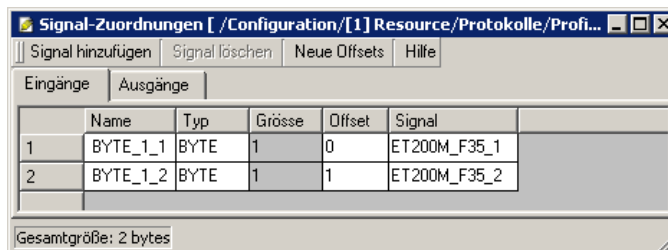


Bild 12: Eingangssignal der HIMatrix Steuerung

Schritt 11: Ausgangssignale der HIMatrix mit den Signalen des Slaves *ET 200M* verbinden:

- ❑ Öffnen Sie das Kontextmenü des PROFIBUS Moduls *8DO PROFIBUS Modul_4*.
- ❑ Wählen Sie aus dem Kontextmenü **Signale Verbinden**.
- ❑ Öffnen Sie das Register **Ausgang** des Dialogfensters *Signal Zuordnungen*.
- ❑ Öffnen Sie über das Hauptmenü **Signale, Editor** den *Signaleditor*.
- ❑ Ziehen Sie das Signal **F35_ET200M_1** per Drag & Drop aus dem *Signaleditor* auf das **Ausgangssignal** im Register *Ausgang* des Dialogfensters *Signale Zuordnungen*.
- ❑ Klicken Sie auf die Schaltfläche **Neue Offsets** im Dialog *Signal Zuordnungen*.
- ❑ Klicken Sie im Dialog *Offsets nummerieren* auf die Schaltfläche **Nummerieren**.
- ❑ Schließen Sie das Dialogfenster mit **OK**.



Bild 13: Ausgangssignal der HIMatrix Steuerung

Schritt 12: Ermitteln Sie die minimale Token-Umlaufzeit

- Öffnen Sie das Register Übertragungsraten im Dialogfenster Eigenschaften.
- Vergleichen Sie MaxTsdr aus Tabelle 6 mit MaxTsdr aus dem Register Übertragungsraten. Verwenden Sie für die weitere Berechnung den größeren der beiden Werte für MaxTsdr.

Die folgenden Zeitkonstanten für eine Übertragungsrate von 500 kbit/s wurden aus der Tabelle 6 entnommen

MinTsdr	=	11	T_{bit}
MaxTsdr	=	100	T_{bit}
Tsl	=	200	T_{bit}
Tqui	=	0	T_{bit}
Tset	=	1	T_{bit}

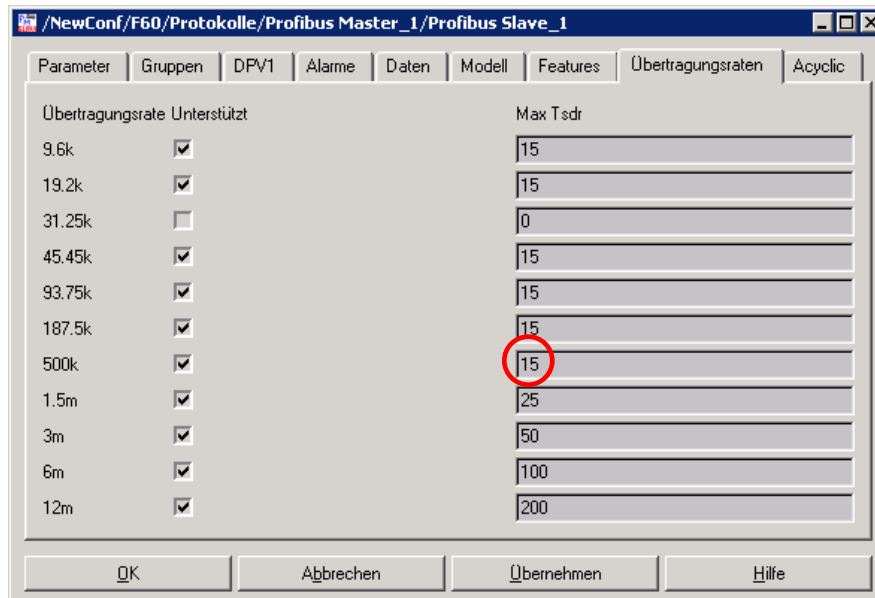


Bild 14: MaxTsdr des Slave Siemens ET 200M

Berechnung der Token-Umlaufzeit Ttr (siehe auch Kapitel 1.4.5.1)

$$T_0 = 35 + 2 \cdot T_{\text{set}} + T_{\text{qui}}$$

$$T_0 = 35 + 2 \cdot 1 + 0$$

$$T_0 = 37 T_{\text{bit}}$$

Da $T_0 > \text{MinTsdr}$: $T_1 = 37 T_{\text{bit}}$

Da $T_0 < \text{MaxTsdr}$: $T_2 = 37 T_{\text{bit}}$

Die ermittelten Werte werden in die Formel für die minimale Token-Umlaufzeit eingesetzt:

$$T_{tr_{min}} = n * (198 + T_1 + T_2) + b * 11 + 242 + T_1 + T_2 + T_{sl}$$

In diesem Beispiel ist ein Slave aktiv (**n = 1**)

3 E/A-Datenbytes werden vom Slave übertragen (Input und Output) (**b = 3**)

$$T_{tr_{min}} = 1 (198 + 37 + 100) + 3 * 11 + 242 + 37 + 37 + 200$$

$$T_{tr_{min}} [T_{bit}] = 884 T_{bit}$$

$$T_{tr_{min}} [\mu s] = 884 T_{bit} * (1/500 \text{ kBit}) = 1,768 \text{ ms}$$

Hinweis Die Abschätzung der minimalen Token-Umlaufzeit $T_{tr_{min}}$ gilt nur, wenn folgende Bedingungen eingehalten sind:

- Es wird nur ein Master am Bus betrieben
- Es werden keine Sendungen wiederholt
- Es findet kein azyklischer Datenverkehr statt

Stellen Sie auf keinen Fall ein kleineres $T_{tr_{min}}$ ein als mit obiger Formel berechnet, da sonst eine fehlerfreie Kommunikation nicht mehr garantiert werden kann.

Es wird empfohlen das Doppelte oder Dreifache der berechneten Token-Umlaufzeit $T_{tr_{min}}$ einzustellen.

$$T_{tr} = T_{tr_{min}} * 3$$

$$T_{tr} = 1768 \text{ ms} * 3 = 5,304 \text{ ms}$$

Ttr -> 6 ms

Schritt 13: Ermitteln der Nutzdatenüberwachungszeit und der Watchdog-Zeit:

Hinweis Die Nutzdatenüberwachungszeit läuft auf dem PROFIBUS-DP Master und die Watchdog-Zeit läuft auf dem PROFIBUS-DP Slave. Es sind nur Schritte im Raster von 10 ms möglich. Es wird empfohlen beide Zeiten gleich einzustellen.

- ❑ Ermitteln Sie mit **Ttr** die Nutzdatenüberwachungszeit
Nutzdatenüberwachungszeit $\geq 6 * T_{tr}$ -> **40 ms**
- ❑ Ermitteln Sie mit **Ttr** die Watchdog-Zeit des Slave
Watchdog-Zeit des Slave $\geq 6 * T_{tr}$ -> **40 ms**

Schritt 14: Parametrieren Sie die den PROFIBUS-DP Slave wie im folgenden Bild:

- ☐ Wählen Sie **Eigenschaften** aus dem Kontextmenü des Slaves.
- ☐ Wählen Sie das Register **Parameter** im Dialog *Eigenschaften*.
- ☐ Setzen Sie das Kontrollkästchen **Aktiv**, damit der Slave aktiv ist.
- ☐ Tragen Sie den Wert für die *Watchdog-Zeit [ms]* aus Schritt 13 ein.

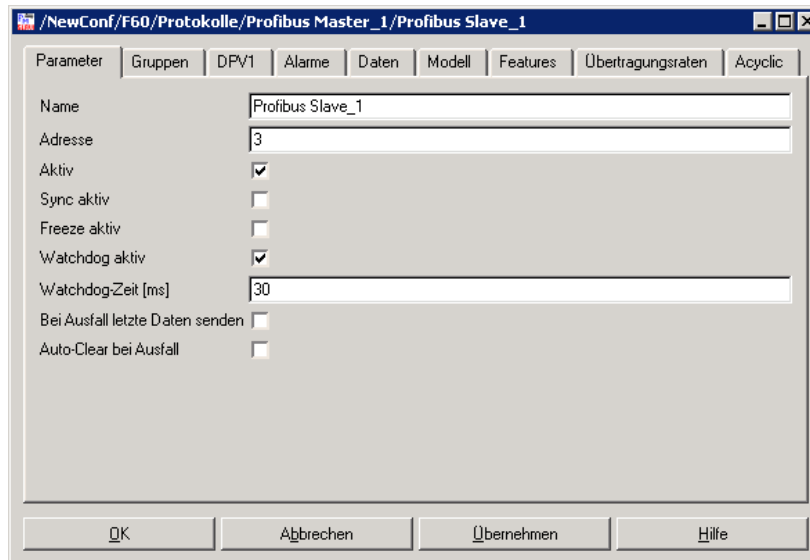


Bild 15: Register *Parameter* des Slave *Siemens ET 200M*

Hinweis Der Slave erkennt den Ausfall eines Masters, wenn innerhalb der Watchdogzeit kein Datenverkehr mit dem Master stattgefunden hat. Bei einem Ausfall des Masters geht der Slave in den Zustand STOPP.

Die Watchdog-Zeit des PROFIBUS-DP Slave hat nichts mit der Watchdog-Zeit zum Ausführen eines Programmzyklus zu tun, die eine Ressource maximal benötigen darf.

Für die Watchdog-Zeit gilt die folgende Randbedingung:

$$\text{Watchdog-Zeit des Slave} > 6 \cdot T_{tr}$$

Schritt 15: Ermitteln Sie den Parameter *Min. Slave Intervall [ms]*

- ❑ Wählen Sie **Eigenschaften** aus dem Kontextmenü des Slaves.
- ❑ Wählen Sie das Register **Features**.
- ❑ Notieren Sie sich den angezeigten Wert *Min. Slave Interval [ms]* für den Schritt 17.

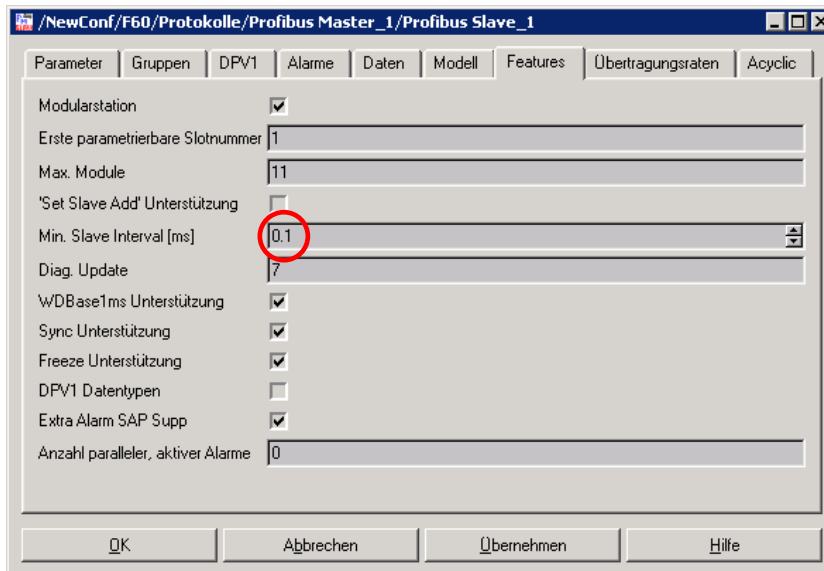


Bild 16: Register *Features* des Slave *Siemens ET 200M*

Schritt 16: Konfigurieren Sie den HIMatrix PROFIBUS-DP Master:

- ❑ Öffnen Sie das Kontextmenü des PROFIBUS-DP Master und wählen Sie *Eigenschaften*.
- ❑ Wählen Sie das Register **Allgemein** im Dialog *Eigenschaften* und konfigurieren Sie den PROFIBUS-DP Master wie im folgenden Bild:

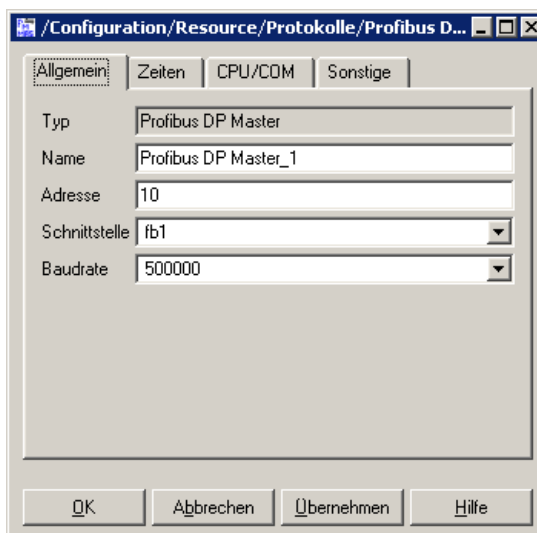


Bild 17: Register *Allgemein* des PROFIBUS-DP Master

Schritt 17: Konfigurieren Sie den HIMatrix PROFIBUS-DP Master wie im folgenden Bild:

- ❑ Wählen Sie das Register **Zeiten** im Dialog *Eigenschaften*.
- ❑ Tragen Sie den Wert für *Min. Slave Intervall [ms]* aus Schritt 15 ein.
- ❑ Tragen Sie den Wert für die *Nutzdatenüberwachungszeit [ms]* aus Schritt 13 ein.

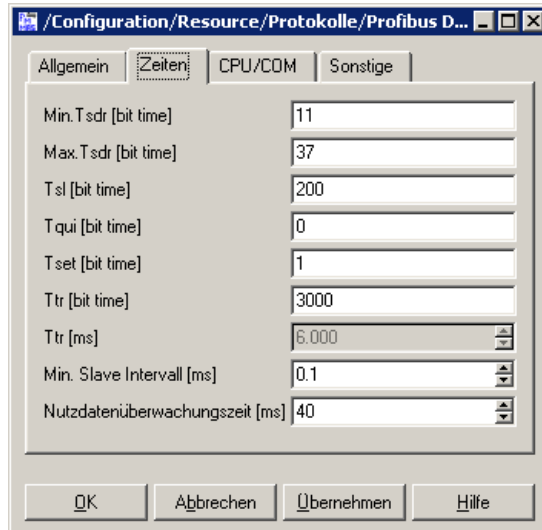


Bild 18: Register *Zeiten* des PROFIBUS-DP Master

Hinweis Sind mehrere Slaves konfiguriert, werden die höchsten Werte der Parameter *MaxT sdr [bit time]* und *Min. Slave Intervall [ms]* verwendet.

Schritt 18: Konfigurieren Sie die sonstigen Parameter des HIMatrix PROFIBUS-DP-Master wie im folgenden Bild:

- ❑ Wählen Sie das Register **Sonstige** im Dialog *Eigenschaften*.

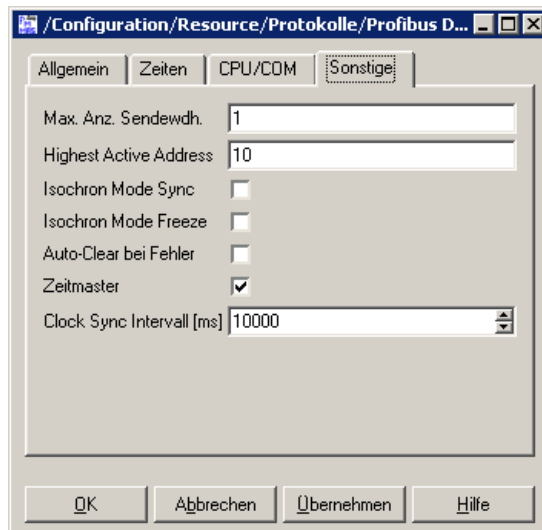


Bild 19: Register *Sonstige* des PROFIBUS-DP Master

Schritt 19: PROFIBUS-DP Master Konfiguration Prüfen:

- Öffnen Sie das Kontextmenü des PROFIBUS-DP Masters und wählen Sie **Validieren**.

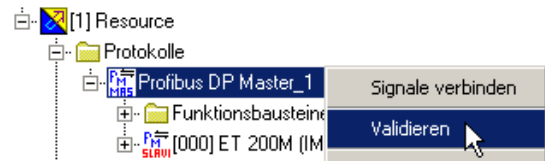


Bild 20: PROFIBUS-DP Master Konfiguration Prüfen

In der Fehler-Status-Anzeige werden nach der Validierung der PROFIBUS-DP Master Konfiguration eventuelle Fehler und Warnungen angezeigt.

11.05.2007 12:05:49.453, Info: [Profibus DP Master_1] Validieren gestartet.
 11.05.2007 12:05:49.453, Fehler: [Profibus DP Master_1] Bitte wählen sie eine Schnittstelle für den Profibus DP Master
 11.05.2007 12:05:49.453, Info: [Profibus DP Master_1] Validieren beendet. Warnungen: 0, Fehler: 1.

Bild 21: Fehler-Status-Anzeige

2 HIMatrix PROFIBUS-DP Funktionsbausteine

Mit den HIMatrix PROFIBUS-DP Funktionsbausteinen kann der Anwender den HIMatrix PROFIBUS-DP Master und die ihm zugeordneten PROFIBUS-DP Slaves optimal den Erfordernissen seines Projekts anpassen.

Die Funktionsbausteine werden im Anwenderprogramm parametrierbar, sodass die Funktionen des Masters und der Slaves (Alarmer, Diagnosedaten, Zustände) im Anwenderprogramm gesetzt und gelesen werden können.

Hinweis Funktionsbausteine werden nur für spezielle Anwendungen benötigt. Für den normalen zyklischen Datenverkehr zwischen Master und Slave sind diese Funktionsbausteine nicht erforderlich!

Es stehen die folgenden Funktionsbausteine zur Verfügung:

Funktionsbaustein	Beschreibung der Funktion	geeignet ab DP-Leistungsstufe
MSTAT	Zustand des Master durch das Anwenderprogramm steuern	DP-V0
RALRM	Alarmmeldungen der Slaves lesen	DP-V1
RDIAG	Diagnosemeldungen der Slaves lesen	DP-V0
RDREC	Azyklische Datensätze der Slaves lesen	DP-V1
SLACT	Zustand der Slaves durch das Anwenderprogramm steuern	DP-V0
WRREC	Azyklische Datensätze der Slaves schreiben	DP-V1

Tabelle 27: Erforderliche Leistungsstufen des PROFIBUS-DP Slaves für die Funktionsbausteine

Hinweis HIMatrix PROFIBUS-DP Master arbeiten mit der Leistungsstufe DP-V2. HIMatrix PROFIBUS-DP Slaves arbeiten mit der Leistungsstufe DP-V0. Beachten Sie, dass dadurch nicht alle Funktionsbausteine mit HIMA Slaves verwendet werden können.

2.1 Konfiguration der Funktionsbausteine

Das PROFIBUS-DP-Protokoll und somit auch die HIMatrix PROFIBUS-DP Funktionsbausteine laufen auf dem Kommunikationsprozessor der Steuerung. Daher müssen die PROFIBUS-DP Funktionsbausteine im ELOP II Factory Hardware-Management angelegt werden.

Um diese Funktionsbausteine mit dem Anwenderprogramm zu steuern werden im ELOP II Factory Projektmanagement Funktionsbausteine angelegt (siehe Kapitel 2.1.1), die wie Standard-Funktionsbausteine im FBS-Editor verwendet werden können.

Die Verbindung der Funktionsbausteine im Projektmanagement mit den entsprechenden Funktionsbausteinen im Hardware-Management erfolgt über gemeinsame Signale. Diese müssen zuvor vom Anwender im Signaleditor erstellt werden.

Vereinbarung:

Funktionsbaustein = Funktionsbaustein im Anwenderprogramm (Projektmanagement)

Funktionsbaustein (HWM) = Funktionsbaustein im Hardware-Management

2.1.1 PROFIBUS-DP Funktionsbausteinbibliothek

Die Funktionsbausteinbibliothek *PBM_V...* muss über die Funktion *Wiederherstellen...* (Kontextmenü des Projekts) dem Projekt hinzugefügt werden.

Die Funktionsbausteinbibliothek erhalten Sie auf Anfrage vom HIMA Support.

Tel.: +49-(0)6202-709 185 oder -259 / -261

E-Mail: support@hima.com

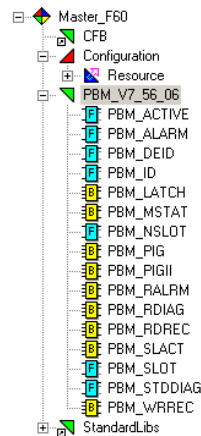


Bild 22: Funktionsbausteine und Hilfsfunktionsbausteine

2.1.2 Konfiguration der Funktionsbaustein im Anwenderprogramm

Die benötigten Funktionsbausteine können per Drag&Drop in das Anwenderprogramm kopiert werden. Konfigurieren Sie die Eingänge und Ausgänge nach der Beschreibung des jeweiligen Funktionsbausteins (ab Kapitel 2.2).

Oberer Teil des Funktionsbausteins

Der obere Teil des Funktionsbausteins entspricht der Benutzerschnittstelle über die der Funktionsbaustein vom Anwenderprogramm gesteuert wird.

Hier werden die Signale verbunden, die im Anwenderprogramm verwendet werden. Das *Präfix A* steht für Application.

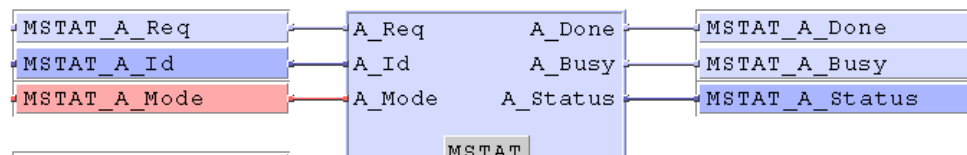


Bild 23: Oberer Teil des Funktionsbausteins

Unterer Teil des Funktionsbausteins

Der untere Teil des Funktionsbausteins stellt die Verbindung zum Funktionsbaustein im Hardware Management (HWM) dar.

Hier werden die Signale verbunden, die mit dem Funktionsbaustein im Hardware-Management verbunden werden müssen. Das *Präfix F* steht für Field.

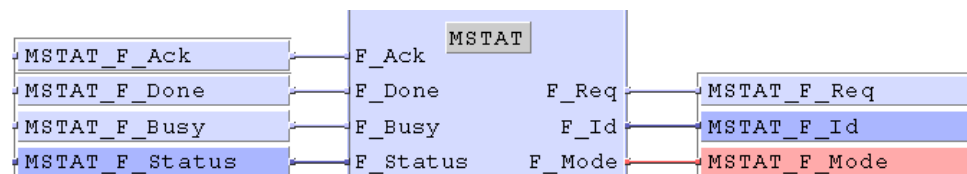


Bild 24: Unterer Teil des Funktionsbausteins

2.1.3 Konfiguration der Funktionsbausteine im Hardware Management

Durch die Auswahl von **Funktionsbausteine, Neu** im Strukturbaum des Hardware-Managements werden alle Funktionsbausteine (HWM) der PROFIBUS-DP-Kommunikation angezeigt.

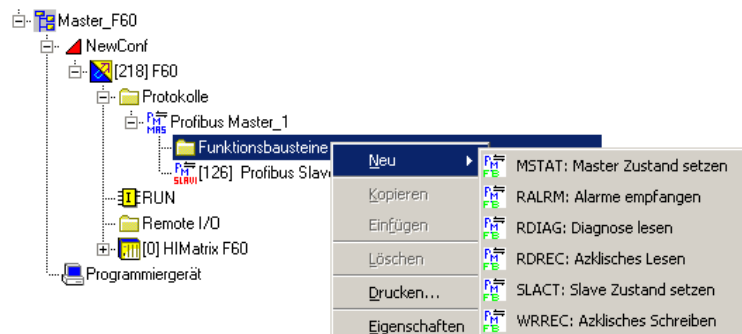


Bild 25: Funktionsbausteine im Hardware-Management (HWM)

So konfigurieren Sie den Funktionsbaustein im Hardware-Management:

- ☐ Öffnen Sie im Strukturbaum den Ordner der Ressource.
- ☐ Wählen Sie über **Protokolle, Profibus Master, Funktionsbausteine, Neu** den passenden Funktionsbaustein (HWM) aus.
- ☐ Öffnen Sie das Kontextmenü des Funktionsbaustein (HWM) und wählen Sie **Edit** um den Dialog *Signal Zuordnungen* zu öffnen

Die Eingänge des Funktionsbausteins (HWM) müssen mit den gleichen Signalen verbunden werden, die mit den *F_Ausgängen* des Funktionsbausteins im Anwenderprogramm verbunden sind.

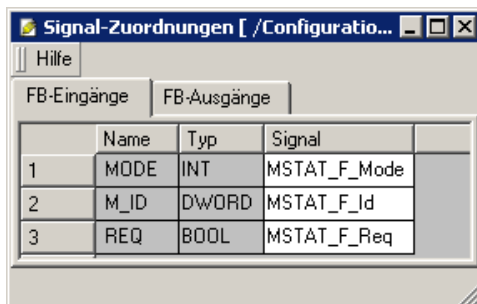


Bild 26: Eingänge des Funktionsbausteins (HWM)

Die Ausgänge des Funktionsbausteins (HWM) müssen mit den gleichen Signalen verbunden werden, die mit den *F_Eingängen* des Funktionsbausteins im Anwenderprogramm verbunden sind.

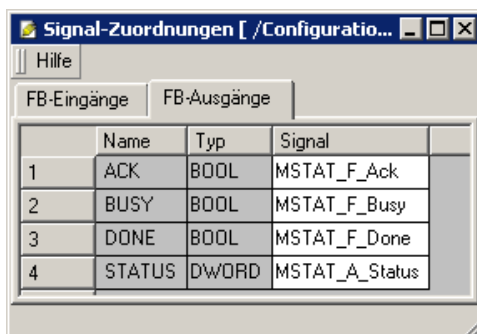


Bild 27: Ausgänge des Funktionsbausteins (HWM)

2.2 Funktionsbaustein *MSTAT*

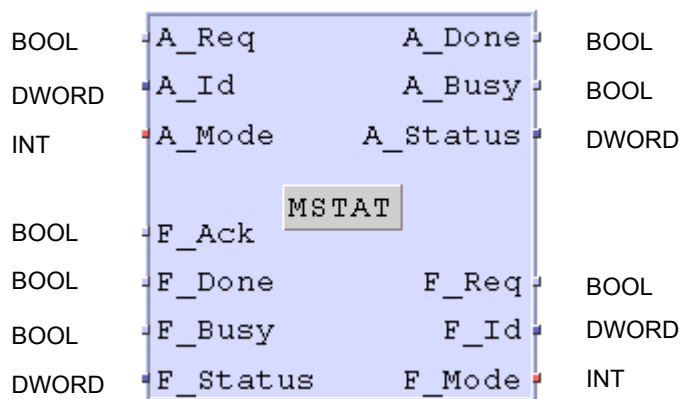


Bild 28: Funktionsbaustein *MSTAT*

Mit dem Funktionsbaustein *MSTAT* (ab DP-V0) kann der PROFIBUS-DP Master vom Anwenderprogramm gesteuert werden. Somit ist es möglich, den PROFIBUS-DP Master durch einen mechanischen Schalter an einem physikalischen Eingang, oder durch einen Timer in einen der folgenden Zustände zu setzen.

0	OFFLINE
1	STOP
2	CLEAR
3	OPERATE

Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag&Drop aus der Bausteinbibliothek in das Anwenderprogramm.

2.2.1 Ein- und Ausgänge des Funktionsbausteins mit dem *Präfix A*

Die Ein- und Ausgänge mit dem *Präfix A* entsprechen der Benutzerschnittstelle über die der Funktionsbaustein vom Anwenderprogramm gesteuert wird.

A_Eingänge	Beschreibung	Typ
A_Req	Positive Flanke startet den Baustein	BOOL
A_Id	Master Id (nicht genutzt)	DWORD
A_Mode	In folgende Zustände kann der PROFIBUS-DP Master gesetzt werden 0: OFFLINE 1: STOP 2: CLEAR 3: OPERATE	INT

Tabelle 28: *A_Eingänge* des Funktionsbausteins *MSTAT*

A_Ausgänge	Beschreibung	Typ
A_Done	TRUE: Der PROFIBUS-DP Master wurde in den am Eingang <i>A_Mode</i> definierten Zustand gesetzt.	BOOL
A_Busy	TRUE: Das setzen des PROFIBUS-DP Master ist noch nicht beendet.	BOOL
A_Status	Status oder Fehlercode (Siehe 2.8 Fehlercodes der Funktionsbausteine)	DWORD

Tabelle 29: A_Ausgänge des Funktionsbausteins *MSTAT*

2.2.2 Ein- und Ausgänge des Funktionsbausteins mit dem Präfix *F*

Die Ein- und Ausgänge des Funktionsbausteins mit dem Präfix *F* dienen der Verbindung mit dem Funktionsbaustein *MSTAT* im Hardware-Management.

Hinweis Die Verbindung des Funktionsbausteins im Projektmanagement mit dem Funktionsbausteins im Hardware-Management erfolgt über gemeinsame Signale. Diese müssen zuvor vom Anwender im Signaleditor erstellt werden.

Die *F_Eingänge* des Funktionsbausteins *MSTAT* im Anwenderprogramm müssen mit den gleichen Signalen verbunden werden, mit denen auch die Ausgänge des Funktionsbausteins *MSTAT* im Hardware-Management verbunden werden.

F_Eingänge	Typ
F_ACK	BOOL
F_DONE	BOOL
F_BUSY	BOOL
F_STATUS	DWORD

Tabelle 30: F_Eingänge des Funktionsbausteins *MSTAT*

Die *F_Ausgänge* des Funktionsbausteins *MSTAT* im Anwenderprogramm müssen mit den gleichen Signalen verbunden werden, mit denen auch die Eingängen des Funktionsbausteins *MSTAT* im Hardware-Management verbunden werden.

F_Ausgänge	Typ
F_REQ	BOOL
F_ID	DWORD
F_MODE	INT

Tabelle 31: F_Ausgänge des Funktionsbausteins *MSTAT*

2.2.3 Funktionsbaustein *MSTAT* im Hardware-Management erstellen

So konfigurieren Sie den Funktionsbaustein *MSTAT* im Hardware-Management:

- ❑ Öffnen Sie im Strukturbaum den Ordner der Ressource.
- ❑ Wählen Sie **Protokolle, Profibus DP Master, Funktionsbausteine, Neu, MSTAT**
- ❑ Öffnen Sie das Kontextmenü des Funktionsbausteins *MSTAT* im Hardware-Management und wählen Sie **Signale verbinden** um den Dialog *Signal Zuordnungen* zu öffnen.

Die folgenden Eingänge des Funktionsbausteins *MSTAT* im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Ausgänge* des Funktionsbausteins *MSTAT* im Anwenderprogramm verbunden sind.

Eingänge	Typ
M_ID	DWORD
REQ	BOOL
MODE	INT

Tabelle 32: Eingänge des Dialogs *MSTAT*

Die folgenden Ausgänge des Funktionsbausteins *MSTAT* im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Eingänge* des Funktionsbausteins *MSTAT* im Anwenderprogramm verbunden sind.

Ausgänge	Typ
ACK	BOOL
BUSY	BOOL
DONE	BOOL
STATUS	DWORD

Tabelle 33: Ausgänge des Dialogs *MSTAT*

2.2.4 Funktionsablauf

Für die Bedienung des Funktionsbausteins *MSTAT* sind die folgenden Schritte erforderlich:

1. Im Anwenderprogramm den Eingang **A_Mode** auf den gewünschten Zustand setzen. Wird *A_Mode* nicht gesetzt, wird nach Schritt 2 ein Fehlercode am Ausgang *A_Status* ausgegeben und der PROFIBUS-DP Master wird nicht gesetzt.
2. Im Anwenderprogramm den Eingang **A_Req** auf TRUE setzen.

Hinweis Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

3. Der Ausgang *A_Busy* geht solange auf TRUE, bis der *MSTAT*-Befehl abgearbeitet ist. Danach geht *A_Busy* auf FALSE und *A_Done* auf TRUE.
4. Konnte der vorgegebene Mode nicht gesetzt werden, wird ein Fehlercode am Ausgang *A_Status* ausgegeben.
5. Der aktuelle Mode des Masters kann dem Signal Master-Status entnommen werden (siehe Statussignale des PROFIBUS-DP Master).

2.3 Funktionsbaustein *RALRM*

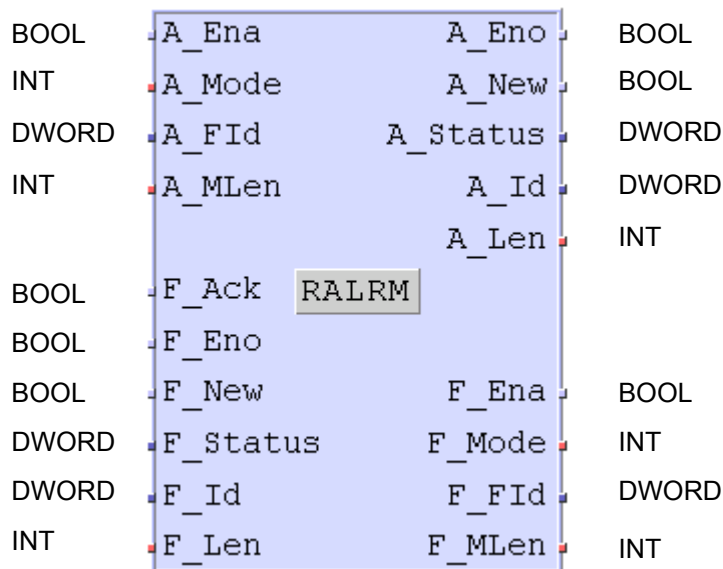


Bild 29: Funktionsbaustein *RALRM*

Der Funktionsbaustein *RALRM* (ab DP-V1) dient zur Auswertung von Alarmen.

Alarme sind eine spezielle Form von Diagnosemeldungen, die vorrangig behandelt werden. Alarme melden der Anwendung, wichtige Ereignisse, die Reaktionen seitens der Anwendung erfordern (z.B. ein WRREC). Dies ist jedoch Hersteller abhängig und kann dem Gerätehandbuch des PROFIBUS-DP Slaves entnommen werden.

Solange der Funktionsbaustein *RALRM* aktiv ist, wartet dieser auf Alarmmeldungen der Slaves. Wird ein Alarm empfangen, wird der Ausgang *A_NEW* für mindestens einen Zyklus auf TRUE geschaltet und die Alarmdaten können per Alarmtelegramm ausgelesen werden. Vor dem nächsten Alarm geht *A_NEW* für mindesten einen Zyklus auf FALSE. Alle Alarme werden implizit bestätigt. Es geht kein Alarm verloren.

Das Anwenderprogramm ist bei Verwendung mehrerer Funktionsbausteine *RALRM* so anzulegen, dass immer nur ein Funktionsbaustein *RALRM* aktiv ist.

Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag&Drop aus der Bausteinbibliothek in das Anwenderprogramm.

2.3.1 Ein- und Ausgänge des Funktionsbausteins mit dem *Präfix A*

Die Ein- und Ausgänge mit dem *Präfix A* entsprechen der Benutzerschnittstelle über die der Funktionsbaustein vom Anwenderprogramm gesteuert wird.

A_Eingänge	Beschreibung	Typ
A_Ena	Mit TRUE wird der Funktionsbaustein freigegeben	BOOL
A_Mode	Nicht genutzt	INT
A_FID	Nicht genutzt	DWORD
A_MLen	Maximal erwartete Länge der zu empfangenden Alarmdaten in Bytes	INT

Tabelle 34: *A_Eingänge* des Funktionsbausteins *RALRM*

A_Ausgänge	Beschreibung	Typ
A_Eno	TRUE: Funktionsbaustein aktiv FALSE: Funktionsbaustein nicht aktiv	BOOL
A_New	TRUE: Neuer Alarm wurde empfangen FALSE: Kein neuer Alarm	BOOL
A_Status	Status oder Fehlercode (Siehe 2.8 Fehlercodes der Funktionsbausteine)	DWORD
A_ID	Identifikationsnummer des Alarm auslösenden Slave	DWORD
A_Len	Länge der empfangenen Alarmdaten in Bytes	INT

Tabelle 35: A_Ausgänge des Funktionsbausteins RALRM

2.3.2 Ein- und Ausgänge des Funktionsbausteins mit dem Präfix F

Die Ein- und Ausgänge des Funktionsbausteins mit dem Präfix F dienen der Verbindung mit dem Funktionsbaustein RALRM im Hardware-Management.

Hinweis Die Verbindung des Funktionsbausteins im Projektmanagement mit dem Funktionsbausteins im Hardware-Management erfolgt über gemeinsame Signale. Diese müssen zuvor vom Anwender im Signaleditor erstellt werden.

Die F_Eingänge des Funktionsbausteins RALRM im Anwenderprogramm müssen mit den gleichen Signalen verbunden werden, mit denen auch die Ausgänge des Funktionsbausteins RALRM im Hardware-Management verbunden werden.

F_Eingänge	Typ
F_ACK	BOOL
F_ENO	BOOL
F_NEW	BOOL
F_STATUS	DWORD
F_ID	DWORD
F_LEN	INT

Tabelle 36: F_Eingänge des Funktionsbausteins RALRM

An die F_Ausgängen des Funktionsbausteins RALRM im Anwenderprogramm müssen mit den gleichen Signale verbunden werden, mit denen auch die Eingängen des Funktionsbausteins RALRM im Hardware-Management verbunden werden.

F_Ausgänge	Typ
F_Ena	BOOL
F_MODE	INT
F_FID	DWORD
F_MLEN	INT

Tabelle 37: F_Ausgänge des Funktionsbausteins RALRM

2.3.3 Funktionsbaustein *RALRM* im Hardware-Management erstellen

So konfigurieren Sie den Funktionsbaustein *RALRM* im Hardware-Management:

- ❑ Öffnen Sie im Strukturbaum den Ordner der Ressource.
- ❑ Wählen Sie **Protokolle, Profibus DP Master, Funktionsbausteine, Neu, RALRM**
- ❑ Öffnen Sie das Kontextmenü des Funktionsbausteins *RALRM* im Hardware-Management und wählen Sie **Signale verbinden** um den Dialog *Signal Zuordnungen* zu öffnen

Die folgenden Eingänge des Funktionsbausteins *RALRM* im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Ausgänge* des Funktionsbausteins *RALRM* im Anwenderprogramm verbunden sind.

Eingänge	Typ
EN	BOOL
F_ID	DWORD
MLen	INT
MODE	INT

Tabelle 38: Eingänge des Dialogs *RALRM*

Die folgenden Ausgänge des Funktionsbausteins *RALRM* im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Eingänge* des Funktionsbausteins *RALRM* im Anwenderprogramm verbunden sind.

Ausgänge	Typ
ACK	BOOL
ENO	BOOL
ID	DWORD
LEN	INT
NEW	BOOL
STATUS	DWORD

Tabelle 39: Ausgänge des Dialogs *RALRM*

Alarmdaten

Im Register *Daten* des Dialogs *RALRM* sind Signale zu definieren, deren Struktur zu den Alarmdaten passen muss. Werden keine Signale definiert, können Alarmdaten zwar angefordert, aber nicht gelesen werden.

Eine Alarmmeldung enthält mindestens vier Bytes. Die ersten vier Bytes der Alarmmeldung enthalten die Standard-Alarmdaten.

Zur vereinfachten Auswertung der Standard-Alarme stellt HIMA den Hilfsfunktionsbaustein *ALARM* (siehe 2.9.2.3) bereit. Wenn Sie diesen Baustein verwenden wollen, fassen Sie die ersten vier Bytes in einem Signal vom Typ DWORD zusammen und geben Sie dieses Signal auf den Eingang *IN* des Hilfsfunktionsbausteins *ALARM*.

Hinweis Enthält ein Alarmtelegramm mehr Bytes als im Register *Daten* definiert wurden, wird nur die Anzahl der definierten Bytes übernommen. Der Rest wird abgeschnitten.

Alarmdaten	Beschreibung
Byte 0	Länge der Alarmmeldung in Byte (4 bis 126)
Byte 1	Kennung für den Alarmtyp 1: Diagnosealarm 2: Prozessalarm 3: Ziehenalarm 4: Steckenalarm 5: Statusalarm 6: Updatealarm Sonst: Herstellerspezifisch. Die Bedeutung muss der Herstellerbeschreibung des Geräts entnommen werden.
Byte 2	Steckplatznummer der alarmlösenden Komponente
Byte 3	Bit 0..1: 0: keine weitere Information 1: ankommender Alarm, Slot gestört 2: ausgehender Alarm, Slot nicht mehr gestört 3: ausgehender Alarm, Slot weiterhin gestört Bit 2: AddAck siehe Tabelle 83 Bit 3 bis 7: Alarm-Sequenznummer
Byte 4 bis 126	Die Bedeutung muss der Herstellerbeschreibung des Geräts entnommen werden

Tabelle 40: Das Alarmtelegramm

Hinweis Die Struktur der Standard-Alarme (Bytes 0...3) ist normiert und für alle Hersteller identisch. Für die herstellerspezifisch genutzten Bytes 4...126 schlagen Sie im Gerätehandbuch des PROFIBUS-DP Slave nach.
Beachten Sie, dass Geräte nach dem DP-V0-Standard keine Alarmtelegramme unterstützen.

2.3.4 Funktionsablauf

Für die Bedienung des Funktionsbausteins *RALRM* sind die folgenden Schritte erforderlich:

1. Im Anwenderprogramm am Eingang **A_Mlen** die Anzahl der maximal zu erwartenden Alarmdaten in Bytes definieren. Während des Betriebs kann *A_Mlen* nicht geändert werden.
2. Im Anwenderprogramm den Eingang **A_Ena** auf TRUE setzen.

Hinweis Im Gegensatz zu den anderen Funktionsbausteinen, ist der Funktionsbaustein *RALRM* nur aktiv, solange der Eingang *A_Ena* TRUE ist.

3. Wurde der Baustein erfolgreich gestartet, dann geht der Ausgang *A_Eno* auf TRUE. Konnte der Baustein nicht gestartet werden, wird ein Fehlercode am Ausgang *A_Status* ausgegeben.
4. Trifft ein neuer Alarm ein, geht der Ausgang *A_New* für mindestens einen Zyklus auf TRUE. Für diese Zeit enthalten die Ausgänge die Alarmdaten des alarmlösenden Slaves, die ausgewertet werden können.
5. Danach geht der Ausgang *A_New* wieder für mindestens einen Zyklus auf FALSE. Die Ausgänge *A_Id* und *A_Len* werden auf Null zurückgesetzt, bevor die nächste Alarmmeldung empfangen und ausgewertet werden kann.

2.4 Funktionsbaustein *RDIAG*

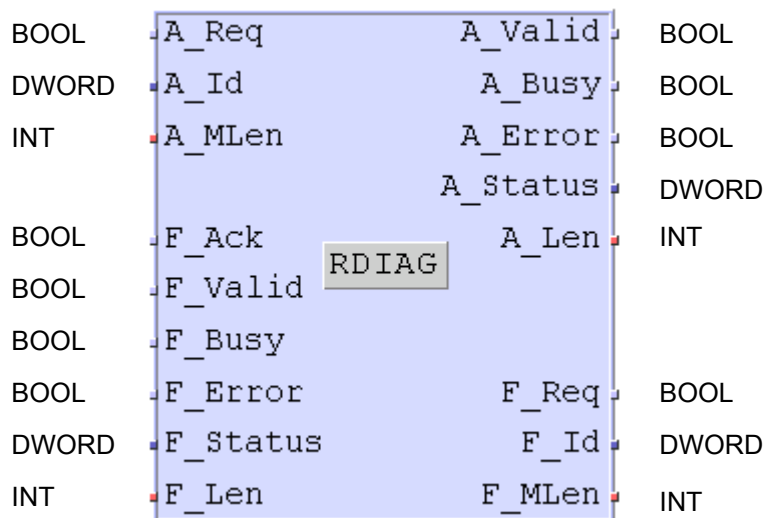


Bild 30: Funktionsbaustein *RDIAG*

Der Funktionsbaustein *RDIAG* (ab DP-V0) dient zum Auslesen der aktuellen Diagnosemeldung (6 Byte bis 240 Bytes) eines Slaves.

Im HIMatrix PROFIBUS-DP Master dürfen beliebig viele *RDIAG*-Bausteine gleichzeitig aktiv sein.

Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag&Drop aus der Bausteinbibliothek in das Anwenderprogramm.

2.4.1 Ein- und Ausgänge des Funktionsbausteins mit dem *Präfix A*

Die Ein- und Ausgänge mit dem *Präfix A* entsprechen der Benutzerschnittstelle über die der Funktionsbaustein vom Anwenderprogramm gesteuert wird.

A_Eingänge	Beschreibung	Typ
A_Req	Positive Flanke startet die Anforderung einer Diagnosemeldung	BOOL
A_ID	Identifikationsnummer des Slave (siehe Hilfsfunktionsbaustein ID)	DWORD
A_MLen	Maximal erwartete Länge der zu lesenden Diagnosemeldung in Bytes	INT

Tabelle 41: *A_Eingänge* des Funktionsbausteins *RDIAG*

A_Ausgänge	Beschreibung	Typ
A_Valid	Eine neue Diagnosemeldung wurde empfangen und ist gültig	BOOL
A_Busy	TRUE: Das Lesen ist noch nicht beendet	BOOL
A_Error	TRUE: Beim Lesen trat ein Fehler auf	BOOL
A_Status	Status oder Fehlercode (Siehe 2.8 Fehlercodes der Funktionsbausteine)	DWORD
A_Len	Länge der gelesenen Diagnosedaten in Bytes	INT

Tabelle 42: *A_Ausgänge* des Funktionsbausteins *RDIAG*

2.4.2 Ein- und Ausgänge des Funktionsbausteins mit dem Präfix F

Die Ein- und Ausgänge des Funktionsbausteins mit dem Präfix *F* dienen der Verbindung mit dem Funktionsbaustein *RDIAG* im Hardware-Management.

Hinweis Die Verbindung des Funktionsbausteins im Projektmanagement mit dem Funktionsbaustein im Hardware-Management erfolgt über gemeinsame Signale. Diese müssen zuvor vom Anwender im Signaleditor erstellt werden.

Die *F_Eingänge* des Funktionsbausteins *RDIAG* im Anwenderprogramm müssen mit den gleichen Signalen verbunden werden, mit denen auch die Ausgänge des Funktionsbausteins *RDIAG* im Hardware-Management verbunden werden.

F_Eingänge	Typ
F_ACK	BOOL
F_VALID	BOOL
F_BUSY	BOOL
F_ERROR	BOOL
F_Status	DWORD
F_LEN	INT

Tabelle 43: F_Eingänge des Funktionsbausteins RDIAG

An die *F_Ausgängen* des Funktionsbausteins *RDIAG* im Anwenderprogramm müssen mit den gleichen Signale verbunden werden, mit denen auch die Eingängen des Funktionsbausteins *RDIAG* im Hardware-Management verbunden werden.

F_Ausgänge	Typ
F_Req	BOOL
F_Id	DWORD
F_MLen	INT

Tabelle 44: F_Ausgänge des Funktionsbausteins RDIAG

2.4.3 Funktionsbaustein *RDIAG* im Hardware-Management erstellen

So konfigurieren Sie den Funktionsbaustein *RDIAG* im Hardware-Management:

- ❑ Öffnen Sie im Strukturbaum den Ordner der Ressource.
- ❑ Wählen Sie **Protokolle, Profibus DP Master, Funktionsbausteine, Neu, RDIAG**
- ❑ Öffnen Sie das Kontextmenü des Funktionsbausteins *RDIAG* im Hardware-Management und wählen Sie **Signale verbinden** um den Dialog *Signal Zuordnungen* zu öffnen

Die folgenden Eingänge des Funktionsbausteins *RDIAG* im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Ausgänge* des Funktionsbausteins *RDIAG* im Anwenderprogramm verbunden sind.

Eingänge	Typ
ID	DWORD
MLen	INT
REQ	BOOL

Tabelle 45: Eingänge des Dialogs *RDIAG*

Die folgenden Ausgänge des Funktionsbausteins *RDIAG* im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Eingänge* des Funktionsbausteins *RDIAG* im Anwenderprogramm verbunden sind.

Ausgänge	Typ
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
Status	DWORD
VALID	BOOL

Tabelle 46: Ausgänge des Dialogs *RDIAG*

Diagnosedaten

Im Register *Daten* sind Signale zu definieren, deren Struktur zu den Diagnosedaten passen muss. Eine Diagnosemeldung enthält mindestens sechs Bytes und maximal 240 Bytes. Die ersten vier Bytes der Diagnosemeldung enthalten die Standard-Diagnose.

Zur vereinfachten Auswertung der Standarddiagnose stellt HIMA den Hilfsfunktionsbaustein *STDDIAG* (siehe 2.9.2.2) bereit. Wenn Sie diesen Baustein verwenden wollen, fassen Sie die ersten vier Bytes in einem Signal vom Typ DWORD zusammen und geben Sie dieses Signal auf den Eingang *IN* des Hilfsfunktionsbausteins *STDDIAG*.

Hinweis	Enthält ein Diagnosetelegramm mehr Bytes als im Register <i>Daten</i> definiert wurden, wird nur die Anzahl der definierten Bytes übernommen. Der Rest wird abgeschnitten.
----------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Diagnosedaten	Beschreibung
Byte 0	Byte 0 bis 3 enthalten die Standarddiagnose. Die Standarddiagnose kann als Signal vom Typ DWORD mit dem Hilfsfunktionsbaustein <i>STDDIAG</i> dekodiert werden.
Byte 1	
Byte 2	
Byte 3	Busadresse des Masters, dem ein Slave zugeordnet ist.
Byte 4	High-Byte (Herstellerkennung)
Byte 5	Low-Byte (Herstellerkennung)
Byte 6 bis 240	Spezifische Slave-Diagnosedaten Die Bedeutung muss der Herstellerbeschreibung des Geräts entnommen werden

Tabelle 47: Der Aufbau des Diagnosetelegramms

Hinweis Die HIMA Slaves liefern ein Diagnosetelegramm von sechs Bytes Länge. Die Bedeutung der Bytes ist standardisiert.
Für Slaves anderer Hersteller sind nur die ersten sechs Bytes funktionell identisch. Für weitere Informationen über das Diagnosetelegramm beachten Sie die Herstellerbeschreibung des Slaves.

2.4.4 Funktionsablauf

Für die Bedienung des Funktionsbausteins *RDIAG* sind die folgenden Schritte erforderlich:

1. Im Anwenderprogramm die Slaveadresse an den Eingang **A_ID** setzen.
2. Im Anwenderprogramm am Eingang **A_Mlen** die Anzahl der maximal zu erwartenden Diagnosedaten in Bytes definieren.
3. Im Anwenderprogramm den Eingang **A_Req** auf TRUE setzen.

Hinweis Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

4. Der Ausgang *A_Busy* geht solange auf TRUE, bis die Diagnoseanforderung abgearbeitet ist. Danach gehen die Ausgänge *A_Busy* auf FALSE und *A_Valid* oder *A_Error* auf TRUE.
5. Ist das Diagnosetelegramm gültig, geht der Ausgang *A_Valid* auf TRUE.
Die Diagnosedaten können über die im Register *Daten* definierten Signale ausgewertet werden.
Der Ausgang *A_Len* enthält die Anzahl der Diagnosedaten in Bytes, die tatsächlich ausgelesen wurden.
6. Konnte das Diagnosetelegramm nicht erfolgreich gelesen werden, dann ist der Ausgang *A_Error* auf TRUE und am Ausgang *A_Status* wird ein Fehlercode ausgegeben.

2.5 Funktionsbaustein *RDREC*

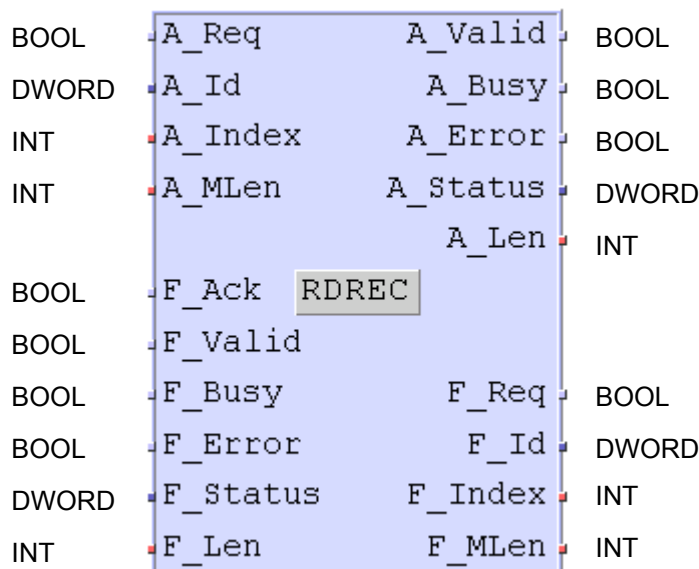


Bild 31: Funktionsbaustein *RDREC*

Der Funktionsbaustein *RDREC* dient zum azyklischen Lesen eines am Eingang *A_Index* adressierten Datensatzes von einem Slave. Welche Daten gelesen werden können, muss der Betriebsanleitung des Slaves entnommen werden.

Diese Funktion ist erst ab DP-V1 definiert und optional!

Im HIMatrix PROFIBUS-DP Master können gleichzeitig bis zu 32 *RDREC*- und/oder *WRREC*-Bausteine aktiv sein.

Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag&Drop aus der Bausteinbibliothek in das Anwenderprogramm.

2.5.1 Ein- und Ausgänge des Funktionsbausteins mit dem *Präfix A*

Die Ein- und Ausgänge mit dem *Präfix A* entsprechen der Benutzerschnittstelle über die der Funktionsbaustein vom Anwenderprogramm gesteuert wird.

A_Eingänge	Beschreibung	Typ
A_Req	Positive Flanke startet die Anforderung zum Lesen	BOOL
A_Id	Identifikationsnummer des Slave (siehe Hilfsfunktionsbaustein ID)	DWORD
A_Index	Datensatznummer des zu lesenden Datensatzes Die Bedeutung muss der Herstellerbeschreibung des Geräts entnommen werden.	INT
A_MLen	Maximale Länge der zu lesenden Daten in Bytes	INT

Tabelle 48: *A_Eingänge* des Funktionsbausteins *RDREC*

A_Ausgänge	Beschreibung	Typ
A_Valid	Ein neuer Datensatz wurde empfangen und ist gültig.	BOOL
A_Busy	TRUE: Der Lesevorgang ist noch nicht beendet.	BOOL
A_Error	TRUE: Ein Fehler ist aufgetreten FALSE: Kein Fehler	BOOL
A_Status	Status oder Fehlercode (Siehe 2.8 Fehlercodes der Funktionsbausteine)	DWORD
A_Len	Länge der gelesenen Datensatzinformation in Bytes.	INT

Tabelle 49: A_Ausgänge des Funktionsbausteins RDREC

2.5.2 Ein- und Ausgänge des Funktionsbausteins mit dem Präfix F

Die Ein- und Ausgänge des Funktionsbausteins mit dem Präfix F dienen der Verbindung mit dem Funktionsbaustein RDREC im Hardware-Management.

Hinweis Die Verbindung des Funktionsbausteins im Projektmanagement mit dem Funktionsbausteins im Hardware-Management erfolgt über gemeinsame Signale. Diese müssen zuvor vom Anwender im Signaleditor erstellt werden.

Die F_Eingänge des Funktionsbausteins RDREC im Anwenderprogramm müssen mit den gleichen Signalen verbunden werden, mit denen auch die Ausgänge des Funktionsbausteins RDREC im Hardware-Management verbunden werden.

F_Eingänge	Typ
F_Ack	BOOL
F_Valid	BOOL
F_Busy	BOOL
F_Error	BOOL
F_Status	DWORD
F_Len	INT

Tabelle 50: F_Eingänge des Dialogs RDREC

An die F_Ausgängen des Funktionsbausteins RDREC im Anwenderprogramm müssen mit den gleichen Signalen verbunden werden, mit denen auch die Eingängen des Funktionsbausteins RDREC im Hardware-Management verbunden werden.

F_Ausgänge	Typ
F_Req	BOOL
F_Id	DWORD
F_Index	INT
F_Mlen	INT

Tabelle 51: F_Ausgänge des Dialogs RDREC

2.5.3 Funktionsbaustein *RDREC* im Hardware-Management erstellen

So konfigurieren Sie den Funktionsbaustein *RDREC* im Hardware-Management:

- ❑ Öffnen Sie im Strukturbaum den Ordner der Ressource.
- ❑ Wählen Sie **Protokolle, Profibus DP Master, Funktionsbausteine, Neu, RDREC**
- ❑ Öffnen Sie das Kontextmenü des Funktionsbausteins *RDREC* im Hardware-Management und wählen Sie **Signale verbinden** um den Dialog *Signal Zuordnungen* zu öffnen

Die folgenden Eingänge des Funktionsbausteins *RDREC* im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Ausgänge* des Funktionsbausteins *RDREC* im Anwenderprogramm verbunden sind.

Eingänge	Typ
ID	DWORD
INDEX	INT
MLEN	INT
REQ	BOOL

Tabelle 52: Eingänge des Dialogs *RDREC*

Die folgenden Ausgänge des Funktionsbausteins *RDREC* im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Eingänge* des Funktionsbausteins *RDREC* im Anwenderprogramm verbunden sind.

Ausgänge	Typ
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
LEN	INT
STATUS	DWORD
VALID	BOOL

Tabelle 53: Ausgänge des Dialogs *RDREC*

Daten	Beschreibung
Es sind keine Signale vorgegeben	Im Register <i>Daten</i> kann eine beliebige Datenstruktur definiert werden, die allerdings auf die Struktur des Datensatzes passen muss. Die Struktur des Datensatzes muss aus der Bedienungsanleitung vom Hersteller des Slaves entnommen werden

Tabelle 54: Daten des Dialogs *RDREC*

2.5.4 Funktionsablauf

Für die Bedienung des Funktionsbausteins *RDREC* sind die folgenden Schritte erforderlich:

1. Im Anwenderprogramm die Slave-Adresse am Eingang **A_ID** setzen.
2. Im Anwenderprogramm den Slave-spezifischen Index für den Datensatz (Handbuch des Herstellers) am Eingang **A_Index** setzen.
3. Im Anwenderprogramm die Länge des zu lesenden Datensatzes am Eingang **A_Len** setzen.
4. Im Anwenderprogramm den Eingang **A_Req** auf TRUE setzen.

Hinweis Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

5. Der Ausgang *A_Busy* geht solange auf TRUE, bis die Datensatzanforderung abgearbeitet ist. Danach gehen die Ausgänge *A_Busy* auf FALSE und *A_Valid* oder *A_Error* auf TRUE.
6. Ist der Datensatz gültig, geht der Ausgang *A_Valid* auf TRUE. Der Datensatz kann über die im Register *Daten* definiert Signale ausgewertet werden. Der Ausgang *A_Len* enthält die tatsächliche Länge des ausgelesenen Datensatz.
7. Konnte der Datensatz nicht erfolgreich gelesen werden, geht der Ausgang *A_Error* auf TRUE und am Ausgang *A_Status* wird ein Fehlercode ausgegeben.

Da der Datensatz des Slaves über den Bus übertragen wird, kann dieser Funktionsaufruf sehr lange dauern.

2.6 Funktionsbaustein SLACT

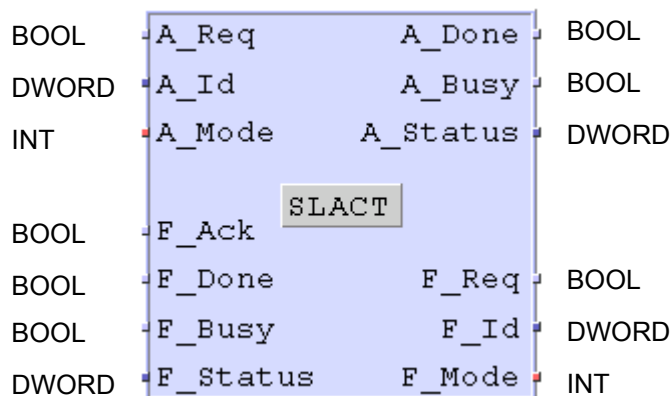


Bild 32: Funktionsbaustein SLACT

Der Funktionsbaustein *SLACT* (ab DP-V0) dient zum Aktivieren und Deaktivieren eines Slaves aus dem Anwenderprogramm des PROFIBUS-DP Master. Somit ist es möglich, den Slave durch einen mechanischen Schalter an einem physikalischen Eingang des PROFIBUS-DP Master, oder durch einen Timer in einen der folgenden Zustände zu setzen.

≠ 0: Aktiv

= 0: Nicht aktiv

Das Anwenderprogramm ist bei Verwendung mehrerer Funktionsbausteine *SLACT* so anzulegen, dass immer nur ein Funktionsbaustein *SLACT* aktiv ist.

Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag&Drop aus der Bausteinbibliothek in das Anwenderprogramm.

2.6.1 Ein- und Ausgänge des Funktionsbausteins mit dem Präfix A

Die Ein- und Ausgänge mit dem Präfix *A* entsprechen der Benutzerschnittstelle über die der Funktionsbaustein vom Anwenderprogramm gesteuert wird.

A_Eingänge	Beschreibung	Typ
A_Req	Positive Flanke startet den Funktionsbaustein	BOOL
A_ID	Identifikationsnummer des Slave (siehe Hilfsfunktionsbaustein ID)	DWORD
A_Mode	Zustand, in den der PROFIBUS-DP Slave gesetzt werden soll: ≠ 0: Aktiv (Verbunden) = 0: Nicht aktiv (Deaktiviert)	INT

Tabelle 55: A_Eingänge des Funktionsbausteins SLACT

A_Ausgänge	Beschreibung	Typ
A_Done	TRUE: Der PROFIBUS-DP Slave wurde in den am Eingang <i>A_Mode</i> definierten Zustand gesetzt.	BOOL
A_Busy	TRUE: Das setzen des PROFIBUS-DP Slave ist noch nicht beendet.	BOOL
A_Status	Status oder Fehlercode (Siehe 2.8 Fehlercodes der Funktionsbausteine)	DWORD

Tabelle 56: A_Ausgänge des Funktionsbausteins SLACT

2.6.2 Ein- und Ausgänge des Funktionsbausteins mit dem *Präfix F*

Die Ein- und Ausgänge des Funktionsbausteins mit dem *Präfix F* dienen der Verbindung mit dem Funktionsbaustein *SLACT* im Hardware-Management.

Hinweis Die Verbindung des Funktionsbausteins im Projektmanagement mit dem Funktionsbaustein im Hardware-Management erfolgt über gemeinsame Signale. Diese müssen zuvor vom Anwender im Signaleditor erstellt werden.

Die *F_Eingänge* des Funktionsbausteins *SLACT* im Anwenderprogramm müssen mit den gleichen Signalen verbunden werden, mit denen auch die Ausgänge des Funktionsbausteins *SLACT* im Hardware-Management verbunden werden.

F_Eingänge	Typ
F_Ack	BOOL
F_Done	BOOL
F_Busy	BOOL
F_Status	DWORD

Tabelle 57: *F_Eingänge* des Funktionsbausteins *SLACT*

An die *F_Ausgängen* des Funktionsbausteins *SLACT* im Anwenderprogramm müssen mit den gleichen Signalen verbunden werden, mit denen auch die Eingängen des Funktionsbausteins *SLACT* im Hardware-Management verbunden werden.

F_Ausgänge	Typ
F_Req	BOOL
F_Id	DWORD
F_Mode	INT

Tabelle 58: *F_Ausgänge* des Funktionsbaustein *SLACT*

2.6.3 Funktionsbaustein **SLACT** im Hardware-Management erstellen

So konfigurieren Sie den Funktionsbaustein **SLACT** im Hardware-Management:

- ❑ Öffnen Sie im Strukturbaum den Ordner der Ressource.
- ❑ Wählen Sie **Protokolle, Profibus DP Master, Funktionsbausteine, Neu, SLACT**
- ❑ Öffnen Sie das Kontextmenü des Funktionsbausteins **SLACT** im Hardware-Management und wählen Sie **Edit** um den Dialog *Signal Zuordnungen* zu öffnen

Die folgenden Eingänge des Funktionsbausteins **SLACT** im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Ausgänge* des Funktionsbausteins **SLACT** im Anwenderprogramm verbunden sind.

Eingänge	Typ
ID	DWORD
REQ	BOOL
MODE	INT

Tabelle 59: Eingänge des Dialogs **SLACT**

Die folgenden Ausgänge des Funktionsbausteins **SLACT** im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Eingänge* des Funktionsbausteins **SLACT** im Anwenderprogramm verbunden sind.

Ausgänge	Typ
ACK	BOOL
BUSY	BOOL
DONE	BOOL
STATUS	DWORD

Tabelle 60: Ausgänge des Dialogs **SLACT**

2.6.4 Funktionsablauf

Für die Bedienung des Funktionsbausteins **SLACT** sind die folgenden Schritte erforderlich:

1. Im Anwenderprogramm den gewünschten Zustand des Slaves am Eingang **A_Mode** setzen.
2. Im Anwenderprogramm den Identifier mit der Slaveadresse am Eingang **A_ID** setzen.
3. Im Anwenderprogramm den Eingang **A_Req** auf TRUE setzen.

Hinweis Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

4. Der Ausgang *A_Busy* geht solange auf TRUE, bis der SLACT-Befehl abgearbeitet ist. Danach geht *A_Busy* auf FALSE und *A_Done* auf TRUE.
5. Am Ausgang *A_Status* wird der Slave-Mode ausgegeben, wenn der Slave-Mode gesetzt werden konnte.
6. Im Ausgang *A_Status* wird ein Fehlercode ausgegeben, wenn der Slave-Mode nicht gesetzt werden konnte.

Die Ausgänge des Dialogs müssen über Signale mit den *F_Eingängen* des Funktionsbausteins verbunden werden.

2.7 Funktionsbaustein *WRREC*

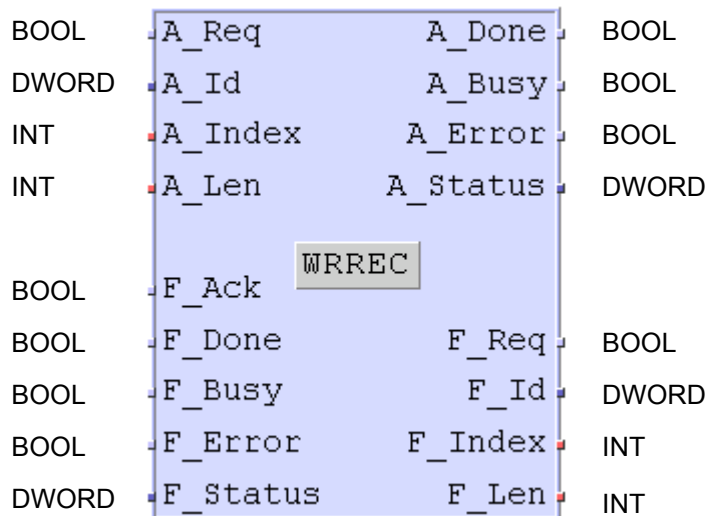


Bild 33: Funktionsbaustein *WRREC*

Der Funktionsbaustein *WRREC* (ab DP-V1) dient zum azyklischen Schreiben eines mit *A_Index* adressierten Datensatzes an einen Slave. Welche Daten geschrieben werden können, muss der Betriebsanleitung des Slaves entnommen werden.

Im HIMatrix PROFIBUS-DP Master können gleichzeitig bis zu 32 *RDREC*- und/oder *WRREC*-Bausteine aktiv sein.

Zur Konfiguration ziehen Sie den Funktionsbaustein per Drag&Drop aus der Bausteinbibliothek in das Anwenderprogramm.

2.7.1 Ein- und Ausgänge des Funktionsbausteins mit dem *Präfix A*

Die Ein- und Ausgänge mit dem *Präfix A* entsprechen der Benutzerschnittstelle über die der Funktionsbaustein vom Anwenderprogramm gesteuert wird.

A_Eingänge	Beschreibung	Typ
A_Req	Positive Flanke startet die Anforderung zum Schreiben eines Datensatzes.	BOOL
A_ID	Identifikationsnummer des Slaves (siehe Hilfsfunktionsbaustein ID)	DWORD
A_Index	Datensatznummer des zu schreibenden Datensatzes. Die Bedeutung muss der Herstellerbeschreibung des Geräts entnommen werden.	INT
A_Len	Länge des zu schreibenden Datensatzes in Bytes	INT

Tabelle 61: A_Eingänge des Funktionsbausteins *WRREC*

A_Ausgänge	Beschreibung	Typ
A_Done	TRUE: Funktionsbaustein hat den Schreibvorgang beendet.	BOOL
A_Busy	TRUE: Funktionsbaustein hat den Schreibvorgang noch nicht beendet.	BOOL
A_Error	TRUE: Beim Schreibvorgang trat ein Fehler auf.	BOOL
A_Status	Status oder Fehlercode (Siehe 2.8 Fehlercodes der Funktionsbausteine)	DWORD

Tabelle 62: A_Ausgänge des Funktionsbausteins WRREC

2.7.2 Ein- und Ausgänge des Funktionsbausteins mit dem Präfix F

Die Ein- und Ausgänge des Funktionsbausteins mit dem Präfix *F* dienen der Verbindung mit dem Funktionsbaustein *WRREC* im Hardware-Management.

Hinweis Die Verbindung des Funktionsbausteins im Projektmanagement mit dem Funktionsbaustein im Hardware-Management erfolgt über gemeinsame Signale. Diese müssen zuvor vom Anwender im Signaleditor erstellt werden.

Die *F_Eingänge* des Funktionsbausteins *WRREC* im Anwenderprogramm müssen mit den gleichen Signalen verbunden werden, mit denen auch die Ausgänge des Funktionsbausteins *WRREC* im Hardware-Management verbunden werden.

F_Eingänge	Typ
F_Ack	BOOL
F_Done	BOOL
F_Busy	BOOL
F_Error	BOOL
F_Status	DWORD

Tabelle 63: F_Eingänge des Dialogs WRREC

An die *F_Ausgängen* des Funktionsbausteins *WRREC* im Anwenderprogramm müssen mit den gleichen Signale verbunden werden, mit denen auch die Eingängen des Funktionsbausteins *WRREC* im Hardware-Management verbunden werden.

F_Ausgänge	Typ
F_Req	BOOL
F_Id	DWORD
F_Index	INT
F_Len	INT

Tabelle 64: F_Ausgänge des Dialogs WRREC

2.7.3 Funktionsbaustein *WRREC* im Hardware-Management erstellen

So konfigurieren Sie den Funktionsbaustein *WRREC* im Hardware-Management:

- ❑ Öffnen Sie im Strukturbaum den Ordner der Ressource.
- ❑ Wählen Sie **Protokolle, Profibus DP Master, Funktionsbausteine, Neu, WRREC**
- ❑ Öffnen Sie das Kontextmenü des Funktionsbausteins *WRREC* im Hardware-Management und wählen Sie **Signale verbinden** um den Dialog *Signal Zuordnungen* zu öffnen.

Die folgenden Eingänge des Funktionsbausteins *WRREC* im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Ausgänge* des Funktionsbausteins *WRREC* im Anwenderprogramm verbunden sind.

Eingänge	Typ
ID	DWORD
INDEX	INT
LEN	INT
REQ	BOOL

Tabelle 65: Eingänge des Dialogs *WRREC*

Die folgenden Ausgänge des Funktionsbausteins *WRREC* im Hardware-Management müssen mit den gleichen Signalen verbunden werden, mit denen auch die *F_Eingänge* des Funktionsbausteins *WRREC* im Anwenderprogramm verbunden sind.

Ausgänge	Typ
ACK	BOOL
BUSY	BOOL
ERROR	BOOL
STATUS	DWORD
DONE	BOOL

Tabelle 66: Ausgänge des Dialogs *WRREC*

Daten	Beschreibung
Es sind keine Signale vorgegeben	Im Register <i>Daten</i> kann eine beliebige Datenstruktur definiert werden, die allerdings auf die Struktur des Datensatzes passen muss. Die Struktur des Datensatzes muss aus der Bedienungsanleitung vom Hersteller des Slaves entnommen werden.

Tabelle 67: Daten des Dialogs *WRREC*

2.7.4 Funktionsablauf

Für die Bedienung des Funktionsbausteins *WRREC* sind die folgenden Schritte erforderlich:

1. Im Anwenderprogramm die Slave-Adresse am Eingang **A_ID** setzen.
2. Im Anwenderprogramm den Slave-spezifischen Index für den Datensatz (Handbuch des Herstellers) am Eingang **A_Index** setzen.
3. Im Anwenderprogramm die Länge des zu schreibenden Datensatzes am Eingang **A_Len** setzen.
4. Im Anwenderprogramm den Datensatz, wie im Register *Daten* definiert einstellen.
5. Im Anwenderprogramm den Eingang **A_Req** auf TRUE setzen.

Hinweis Der Funktionsbaustein reagiert auf einen positiven Flankenwechsel an *A_Req*.

6. Der Ausgang *A_Busy* geht solange auf TRUE, bis der Datensatz geschrieben ist. Danach gehen die Ausgänge *A_Busy* auf FALSE und *A_Done* auf TRUE.
7. Konnte der Datensatz nicht erfolgreich geschrieben werden, geht der Ausgang *A_Error* TRUE und am Ausgang *A_Status* wird ein Fehlercode ausgegeben.

2.8 Fehlercodes der Funktionsbausteine

Wenn ein Funktionsbaustein ein Kommando nicht korrekt ausführen konnte, wird am Ausgang *A_Status* (des Funktionsbausteins) und *STATUS* (im Funktionsbaustein im HWM) ein Fehlercode ausgegeben. Die Bedeutung des Fehlercodes entnehmen Sie der folgenden Tabelle.

Fehlercode	Symbol	Erklärung
16#40800800	TEMP_NOT_AVAIL	Dienst steht vorübergehend nicht zur Verfügung
16#40801000	INVALID_PARA	Ungültiger Parameter
16#40801100	WRONG_STATE	Slave unterstützt kein DP-V1
16#40808000	FATAL_ERR	Fataler Programmfehler
16#40808100	BAD_CONFIG	Konfigurationsfehler im Datenbereich
16#40808200	PLC_STOPPED	Steuerung wurde gestoppt
16#4080A000	READ_ERR	Fehler beim Lesen eines Records
16#4080A100	WRITE_ERR	Fehler beim Schreiben eines Records
16#4080A200	MODULE_FAILURE	Fehler nicht näher spezifizierbar
16#4080B000	INVALID_INDEX	Index ist ungültig
16#4080B100	WRITE_LENGTH	Falsche Länge beim Schreiben
16#4080B200	INVALID_SLOT	Slot-Nummer ist ungültig
16#4080B300	TYPE_CONFLICT	Falscher Typ
16#4080B400	INVALID_AREA	Falscher Lese- oder Schreibbereich
16#4080B500	STATE_CONFLICT	Master im falschen Zustand
16#4080B600	ACCESS_DENIED	Slave nicht aktiv (oder ähnliches)
16#4080B700	INVALID_RANGE	Falscher Lese- oder Schreibbereich
16#4080B800	INVALID_PARAMETER	Falscher Parameterwert
16#4080B900	INVALID_TYPE	Falscher Parametertyp
16#4080C300	NO_RESOURCE	Slave nicht vorhanden
16#4080BA00	BAD_VALUE	Ungültiger Wert
16#4080BB00	BUS_ERROR	Busfehler
16#4080BC00	INVALID_SLAVE	Ungültige Slave-Id
16#4080BD00	TIMEOUT	Timeout aufgetreten
16#4080C000	READ_CONSTRAIN	Lesebeschränkung
16#4080C100	WRITE_CONSTRAIN	Schreibbeschränkung
16#4080C200	BUSY	ein Baustein dieser Art ist bereits aktiv
16#4080C300	NO_RESOURCE	Slave nicht aktiv

Tabelle 68: Fehlercodetabelle der Funktionsbausteine

2.9 Hilfsfunktionsbausteine

Die folgenden Hilfsfunktionsbausteine laufen komplett im Anwenderprogramm auf der CPU ab.

Die Hilfsfunktionsbausteine befinden sich in der gleichen Funktionsbausteinbibliothek **PBM_V...** wie die Funktionsbausteine (siehe Kapitel 2.1.1). Der Anwender wählt die benötigten Hilfsfunktionsbausteine aus und zieht diese einzeln per Drag&Drop direkt in das Anwenderprogramm.

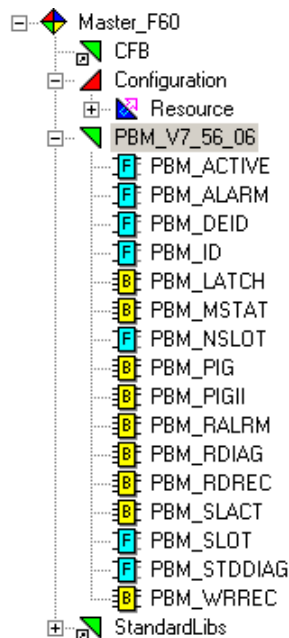


Bild 34: Funktionsbausteine und Hilfsfunktionsbausteine

Die folgenden Hilfsfunktionsbausteine stehen dem Anwender zur Verfügung.

Hilfsfunktionsbausteine	Beschreibung
ID	Die Funktion ID generiert aus vier Bytes einen Identifier
SLOT	SLOT Identifikationsnummer mit Slot-Nummer erstellen
NSLOT	Fortlaufende Identifikationsnummer für die Slots erstellen
DEID	Identifikationsnummer dekodieren
ACTIVE	Ist der Slave Aktiv oder Inaktiv
STDDIAG	Standarddiagnose eines Slaves dekodieren
ALARM	Dekodieren der Alarmdaten
LATCH	Wird nur innerhalb anderer Funktionsbausteine verwendet
PIG	Wird nur innerhalb anderer Funktionsbausteine verwendet
PIGII	Wird nur innerhalb anderer Funktionsbausteine verwendet

Tabelle 69: Die Hilfsfunktionsbausteine und Ihre Funktion

Hinweis Die Signale für die Hilfsfunktionsbausteine müssen im Editor des Hardware-Managements erstellt werden. Die Signale werden dann per Drag&Drop in das Anwenderprogramm eingefügt.

2.9.1 Hilfsfunktionsbausteine die mit dem Identifier arbeiten

Die vier folgenden Hilfsfunktionsbausteine arbeiten mit einem Identifier (Identifikationsnummer) vom Typ DWORD.

Diese Hilfsfunktionsbausteine bereiten den Identifier so vor, dass dieser von den HIMatrix PROFIBUS-DP Funktionsbausteinen verwendet werden kann, um Master, Segment, Slaves und Module/Slots zu identifizieren.

2.9.1.1 Hilfsfunktionsbaustein *ID*

(Identifikationsnummer erstellen)

Die Funktion *ID* generiert aus vier Bytes einen Identifier. Der Eingang Master ist für zukünftige Funktionen und wird z.Z. nicht verwendet, weil jeder PROFIBUS-DP Funktionsbaustein eindeutig einem Master zugeordnet ist. Die Segment-Nummer ist nur für Klasse-2-Master von Bedeutung und wird deshalb nicht verwendet.

Diese beiden Eingänge müssen offen gelassen oder mit Null belegt werden.

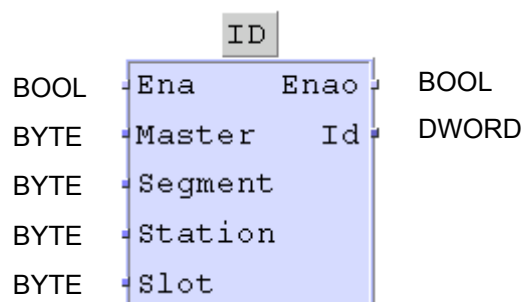


Bild 35: Hilfsfunktionsbaustein *ID*

Eingänge	Beschreibung	Typ
Ena	Nicht genutzt	BOOL
Master	Nicht genutzt	BYTE
Segment	Nicht genutzt	BYTE
Station	Busadresse des Slave	BYTE
Slot	Slot- oder Modul Nummer	BYTE

Tabelle 70: Eingänge des Hilfsfunktionsbausteins *ID*

Ausgänge	Beschreibung	Typ
Enao	Nicht genutzt	BOOL
Ido	Identifikationsnummer des Slaves (Slave-Id und Slot-Nummer)	DWORD

Tabelle 71: Ausgänge des Hilfsfunktionsbausteins *ID*

2.9.1.2 Hilfsfunktionsbaustein *SLOT*

(Identifikationsnummer mit Slot-Nummer erstellen)

Die Funktion *SLOT* generiert aus einem Identifier und einer Slot-Nummer einen neuen Identifier, der den gleichen Slave adressiert, wie der alte Identifier, jedoch mit der neuen Slot-Nummer.

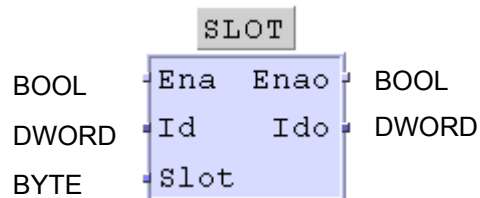


Bild 36: Hilfsfunktionsbaustein *SLOT*

Eingänge	Beschreibung	Typ
Ena	Nicht genutzt	BOOL
Id	Logische Adresse der Slave-Komponente (Slave-Id und Slot-Nummer)	DWORD
Slot	Neue Slot- oder Modul Nummer	BYTE

Tabelle 72: Eingänge des Hilfsfunktionsbausteins *SLOT*

Ausgänge	Beschreibung	Typ
Enao	Nicht genutzt	BOOL
Id	Identifikationsnummer des Slaves (Slave-Id und Slot-Nummer)	DWORD

Tabelle 73: Ausgänge des Hilfsfunktionsbausteins *SLOT*

2.9.1.3 Hilfsfunktionsbaustein *NSLOT*

(Fortlaufende Identifikationsnummer für die Slots erstellen)

Die Funktion *NSLOT* generiert aus einem Identifier einen neuen Identifier, der den nächsten Slot im gleichen Slave adressiert. Ena muss auf TRUE gesetzt werden, damit der Hilfsfunktionsbaustein läuft.

Enao ist TRUE, wenn das Ergebnis am Ausgang Ido gültig ist.

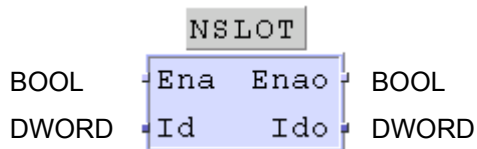


Bild 37: Hilfsfunktionsbaustein *NSLOT*

Eingänge	Beschreibung	Typ
Ena	Solange TRUE anliegt, läuft der Baustein.	BOOL
Id	Identifikationsnummer des Slaves (Slave-Id und Slot-Nummer)	DWORD

Tabelle 74: Eingänge des Hilfsfunktionsbausteins *NSLOT*

Ausgänge	Beschreibung	Typ
Enao	TRUE = Ergebnis gültig	BOOL
Ido	Identifikationsnummer des Slaves (Slave-Id und Slot-Nummer)	DWORD

Tabelle 75: Ausgänge des Hilfsfunktionsbausteins *NSLOT*

2.9.1.4 Hilfsfunktionsbaustein *DEID* (Identifikationsnummer dekodieren)

Die Funktion *DEID* dekodiert einen Identifier und zerlegt ihn in seine vier Bestandteile.

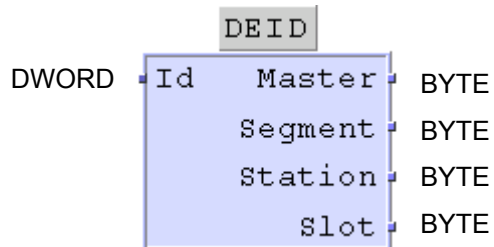


Bild 38: Hilfsfunktionsbaustein *DEID*

Eingänge	Beschreibung	Typ
Id	Identifikationsnummer des Slaves (Slave-Id und Slot-Nummer)	DWORD

Tabelle 76: Eingänge des Hilfsfunktionsbausteins *DEID*

Ausgänge	Beschreibung	Typ
Master	Busadresse des Master	BYTE
Segment	Segment	BYTE
Station	Busadresse des Slave	BYTE
Slot	Slot- oder Modul Nummer	BYTE

Tabelle 77: Ausgänge des Hilfsfunktionsbausteins *DEID*

2.9.2 Hilfsfunktionsbausteine die mit der Standarddiagnose arbeiten

Die folgenden zwei Hilfsfunktionsbausteine arbeiten mit der Standarddiagnose eines Slaves.

Die ersten vier Bytes der Standarddiagnose eines Slaves kann der Anwender auf ein Signal vom Typ DWORD legen.

- ❑ Erzeugen Sie im Signaleditor das Signal **Stddiag** vom Typ DWORD.
- ❑ Selektieren Sie im Hardware-Management einen Slave.
- ❑ Öffnen Sie das Kontextmenü des Slave
- ❑ Wählen Sie **Signale verbinden**.
- ❑ Ziehen Sie das Signal **Stddiag** per Drag&Drop aus dem Signaleditor in das Register *Stati* des Dialogfensters *Signal Zuordnungen*.

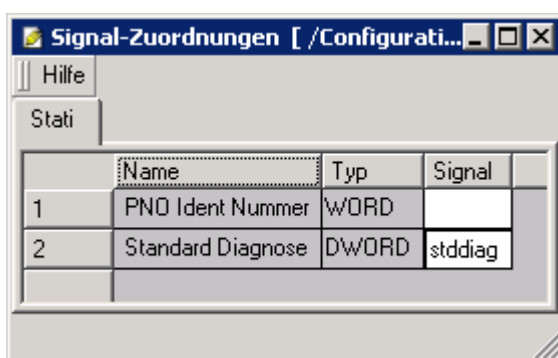


Bild 39: Auslesen der Standarddiagnose eines PROFIBUS-DP Slave

2.9.2.1 Hilfsfunktionsbaustein **ACTIVE**

(Ist der Slave Aktiv oder Inaktiv)

Die Funktion ACTIVE ermittelt aus der Standarddiagnose eines Slaves, ob der Slave gerade aktiv ist oder nicht.

DWORD **IN** Active **OUT** BOOL

Bild 40: Hilfsfunktionsbaustein **ACTIVE**

Eingänge	Beschreibung	Typ
IN	Standarddiagnose des Slave	DWORD

Tabelle 78: Eingänge des Hilfsfunktionsbausteins **ACTIVE**

Ausgänge	Beschreibung	Typ
OUT	TRUE: Slave ist aktiv FALSE: Slave ist inaktiv	BOOL

Tabelle 79: Ausgänge des Hilfsfunktionsbausteins **ACTIVE**

2.9.2.2 Hilfsfunktionsbaustein *STDDIAG*

(Standarddiagnose eines Slaves dekodieren)

Die Funktion *STDDIAG* dekodiert die Standarddiagnose vom Typ DWORD eines Slaves.

Die Ausgänge vom Typ BOOL des Funktionsbaustein *STDDIAG* sind TRUE, wenn das dazugehörige Bit in der Standarddiagnose gesetzt ist.

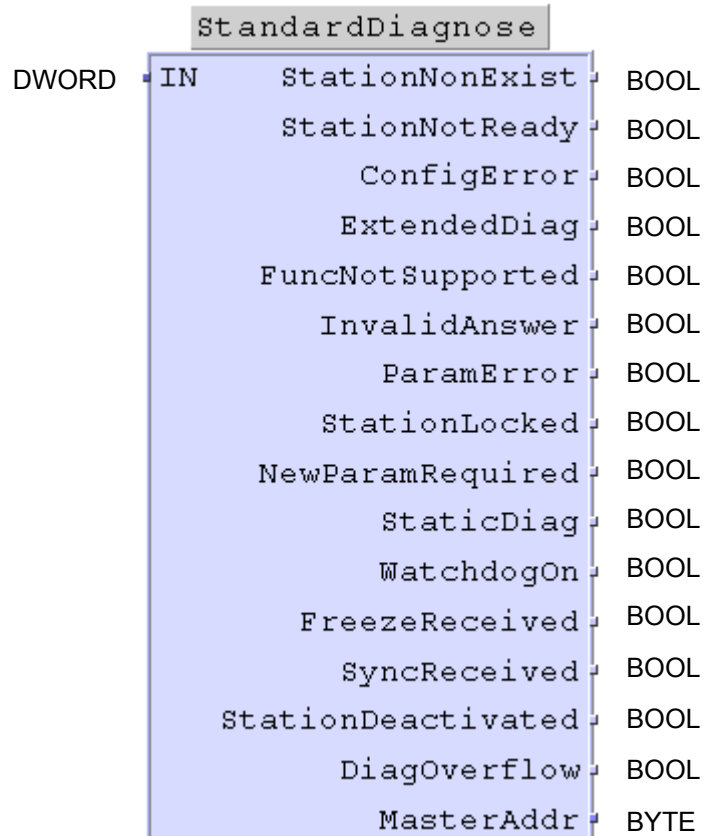


Bild 41: Hilfsfunktionsbaustein *STDDIAG*

Eingänge	Beschreibung	Typ
IN	Standarddiagnose des Slave	DWORD

Tabelle 80: Eingänge des Hilfsfunktionsbausteins *STDDIAG*

Ausgänge	Beschreibung	Typ
StationNonExist	Slave existiert nicht	BOOL
StationNotReady	Slave nicht bereit	BOOL
ConfigError	Konfigurationsfehler	BOOL
ExtendedDiag	Erweiterte Diagnose folgt	BOOL
FuncNotSupported	Funktion nicht unterstützt	BOOL
InvalidAnswer	Ungültige Antwort vom Slave	BOOL
ParamError	Parametrierfehler	BOOL
StationLocked	Slave von anderem Master gesperrt	BOOL
NewParamRequired	Neue Parametrierdaten erforderlich	BOOL
StaticDiag	Statische Diagnose	BOOL
WatchdogOn	Watchdog aktiviert	BOOL
FreezeReceived	Freeze-Kommando erhalten	BOOL
SyncReceived	Sync-Kommando erhalten	BOOL
StationDeactivated	Slave wurde deaktiviert	BOOL
DiagOverflow	Diagnose überlauf	BOOL
MasterAddr	Busadresse des Masters	BYTE

Tabelle 81: Ausgänge des Hilfsfunktionsbausteins *STDDIAG*

2.9.2.3 Hilfsfunktionsbaustein *Alarm*

(Dekodieren der Alarmdaten)

Die Funktion *Alarm* operiert auf den Standard-Alarmdaten, das sind die ersten vier Bytes einer Alarmmeldung, die zu einem DWORD zusammengefasst werden.

Der Hilfsfunktionsbaustein Alarm dekodiert die Alarmdaten.

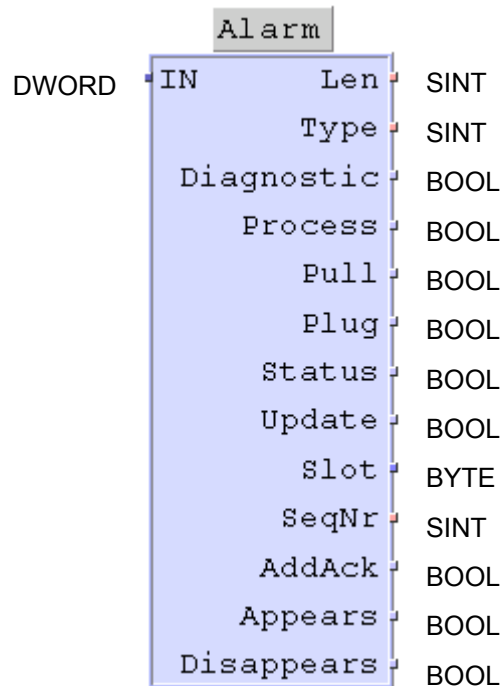


Bild 42: Hilfsfunktionsbaustein *Alarm*

Eingänge	Beschreibung	Typ
IN	Standarddiagnose	DWORD

Tabelle 82: Eingänge des Hilfsfunktionsbausteins *Alarm*

Ausgang	Beschreibung		Typ
Len	Länge der gesamten Alarmmeldung		SINT
Type	1=Diagnosealarm 2=Prozessalarm 3=Ziehenalarm 4=Steckenalarm 5=Statusalarm 6=Updatealarm Andere Nummern sind entweder reserviert oder herstellerspezifisch. Die Bedeutung der jeweiligen Alarme muss dem Handbuch entnommen werden.		SINT
Diagnostic	True = Diagnosealarm		BOOL
Process	True = Prozessalarm		BOOL
Pull	True = Modul wurde gezogen		BOOL
Plug	True = Modul wurde wieder gesteckt		BOOL
Status	True = Status-Alarm		BOOL
Update	True = Update-Alarm		BOOL
Slot	Alarmauslösendes Modul		BYTE
SeqNr	Alarm-Sequenznummer		SINT
AddAck	TRUE bedeutet, dass der Slave, der diesen Alarm ausgelöst hat, eine zusätzliche Bestätigung durch die Anwendung erwartet. Welche genau, muss dem Handbuch entnommen werden.		BOOL
Appears Disappears	FALSE FALSE	Sind beide FALSE, dann ist bis zu diesem Zeitpunkt kein Fehler aufgetreten.	BOOL
Appears Di- sappears	TRUE FALSE	Ein Fehler ist aufgetreten und steht noch an.	
Appears Disappears	FALSE TRUE	Ein Fehler war aufgetreten und verschwindet gerade.	
Appears Di- sappears	TRUE TRUE	Sind beide TRUE, dann verschwindet der Fehler, der Slave ist aber weiterhin gestört.	

Tabelle 83: Beschreibung der Funktionen des Hilfsfunktionsbausteins *Alarm*

2.10 Beispiel: Konfiguration des Funktionsbausteins *RDIAG*

In diesem Beispiel wird die Einrichtung und Anwendung des Funktionsbausteins *RDIAG* anhand der Standarddiagnose vom Typ DWORD beschrieben. Die Standarddiagnose wird über den Funktionsbaustein *RDIAG* in das Anwenderprogramm eingelesen und dort mit dem Hilfsfunktionsbaustein *STDDIAG* dekodiert.

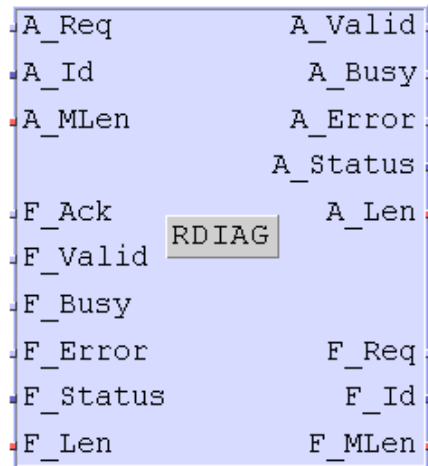


Bild 43: Funktionsbaustein *RDIAG*

Starten Sie **ELOP II Factory** und erstellen Sie ein neues Projekt, oder laden Sie ein vorhandenes Projekt.

Schritt 1: Erstellen Sie den Funktionsbaustein *RDIAG* im Hardware-Management

- ❑ Wählen Sie im Strukturbaum der Ressource **Protokolle, Profibus Master, Funktionsbausteine**.
- ❑ Öffnen Sie das Kontextmenü der *Funktionsbausteine* und wählen Sie **Neu, RDIAG**.

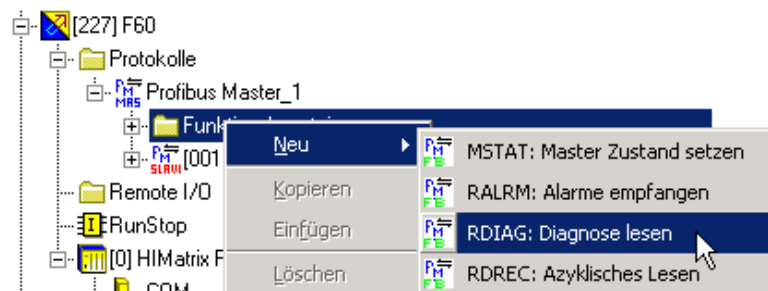


Bild 44: Funktionsbaustein *RDIAG* erstellen

Schritt 2:

Erstellen Sie die für den Funktionsbaustein benötigten Signale:

- ☐ Öffnen Sie im Hauptmenü den Signaleditor mit **Signale, Editor**.
- ☐ Erstellen Sie die folgenden Signale im *Signaleditor*.

Signale für <i>FB Ausgänge</i>		Typ
RDIAG_Ack		BOOL
RDIAG_Busy		BOOL
RDIAG_Error		BOOL
RDIAG_Len		INT
RDIAG_Status		DWORD
RDIAG_Valid		BOOL
Signale für <i>FB Eingänge</i>		
RDIAG_Id		DWORD
RDIAG_Mlen		INT
RDIAG_Req		BOOL
Signale für <i>FB Daten</i>		
stddiag		DWORD
Signale	Wert	Typ
Slaveld	1	Byte
Request	-	BOOL
A_Mlen	4	INT

Tabelle 84: Signale für den Dialog *RDIAG*

Schritt 3:

Verbinden Sie die Signale mit den *FB Ausgängen*:

- ☐ Wählen Sie **Signale verbinden** aus dem Kontextmenü von *RDIAG* um den Dialog *Signal-Zuordnungen* zu öffnen.
- ☐ Ziehen Sie die Signale aus dem Signaleditor in das Register **FB Ausgänge** des Dialogfensters *Signal-Zuordnungen*.

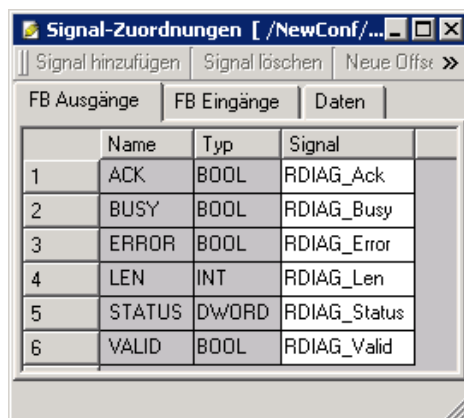


Bild 45: Den *FB Ausgängen* des Dialogs *RDIAG* Signale zuweisen

Schritt 4: Verbinden Sie die Signale mit den *FB Eingängen*:

- ❑ Wählen Sie **Signale verbinden** aus dem Kontextmenü von *RDIAG* um den Dialog *Signal-Zuordnungen* zu öffnen.
- ❑ Ziehen Sie die Signale aus dem Signaleditor in das Register **FB Eingänge** des Dialogfensters *Signal-Zuordnungen*.

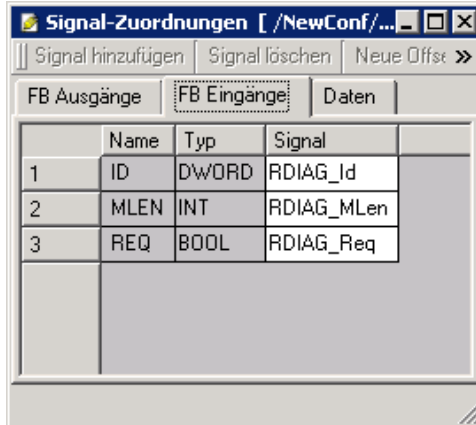


Bild 46: Den *FB_Eingängen* des Dialogs *RDIAG* Signale zuweisen

Schritt 5: Verbinden Sie das Signal **stddiag** mit den *Daten*:

- ❑ Wählen Sie **Signale verbinden** aus dem Kontextmenü von *RDIAG* um den Dialog *Signal-Zuordnungen* zu öffnen.
- ❑ Ziehen Sie das Signal **stddiag** aus dem Signaleditor in das Register **Daten** des Dialogfensters *Signal-Zuordnungen*.

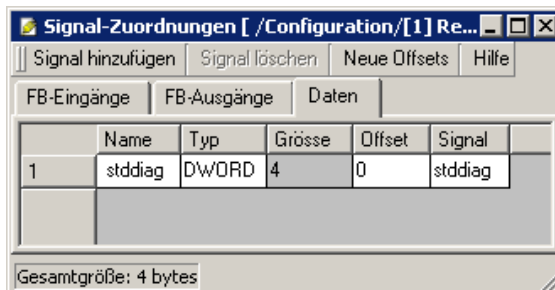


Bild 47: Den *Daten* des Dialogs *RDIAG* Signale zuweisen

Schritt 6: Öffnen Sie das Anwenderprogramm im FBS-Editor:

- ❑ Wechseln Sie in das Sie das **Projektmanagement**.
- ❑ Wählen Sie die Ressource in welcher der Funktionsbaustein *RDIAG* angelegt wurde.
- ❑ Öffnen Sie das **Anwenderprogramm**.

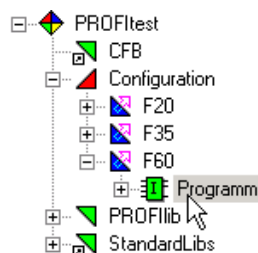


Bild 48: Anwenderprogramm

Schritt 7: Funktionsbausteine in den FBS-Editor einfügen:

- ❑ Öffnen Sie im Strukturbaum die Funktionsbausteinbibliothek **PROFilib**.
- ❑ Ziehen Sie den Funktionsbaustein **RDIAG** per Drag&Drop in den FBS-Editor.
- ❑ Ziehen Sie den Hilfsfunktionsbaustein **STDDIAG** per Drag&Drop in den FBS-Editor.
- ❑ Ziehen Sie den Hilfsfunktionsbaustein **ID** per Drag&Drop in den FBS-Editor

Schritt 8: Erstellen Sie die folgende Logik im FBS-Editor:

- ❑ Ziehen Sie per Drag&Drop alle *Signale* in den FBS-Editor, die Sie in Schritt 3 erstellt haben.
- ❑ Verbinden Sie die *Signale* mit den *Funktionsbausteinen* siehe Bild unten.
- ❑ Erstellen Sie zur Überwachung *Online-Test-Felder* an den Ausgängen der Funktionsbausteine.
- ❑ Schließen Sie den FBS-Editor.

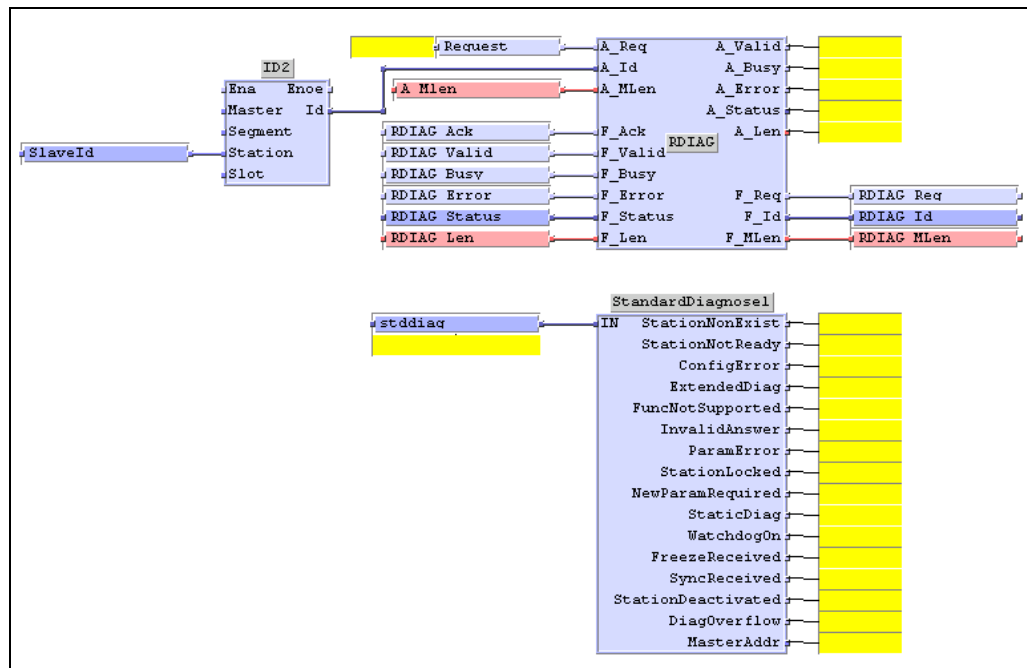


Bild 49: Die fertige Logik mit dem Funktionsbaustein **RDIAG** im FBS-Editor

Schritt 9: Laden Sie den Code in die Steuerung:

- ❑ Starten Sie den Code Generator für die Ressource.
- ❑ Stellen Sie sicher, dass der Code fehlerfrei generiert wurde (siehe Fehler-Status-Anzeige).
- ❑ Laden Sie den Code in die Steuerung und lassen Sie das PADT für den nächsten Schritt mit der Steuerung verbunden.

Schritt 10: Testen Sie die Logik mit dem *Online-Test*.

- ❑ Öffnen Sie das Kontextmenü der Ressource im Strukturbaum und wählen Sie **Online-Test**.
- ❑ Setzen Sie das Signal **Request** auf TRUE, um die Standarddiagnose des Slave (SlaveId = 1) auszulesen.
- ❑ Aktualisieren Sie die Standarddiagnose indem Sie *Request* für kurze Zeit auf FALSE und dann wieder auf TRUE setzen.

Weitere Informationen zur Funktion des Funktionsbaustein *RDIAG* finden Sie in Kapitel 2.4.

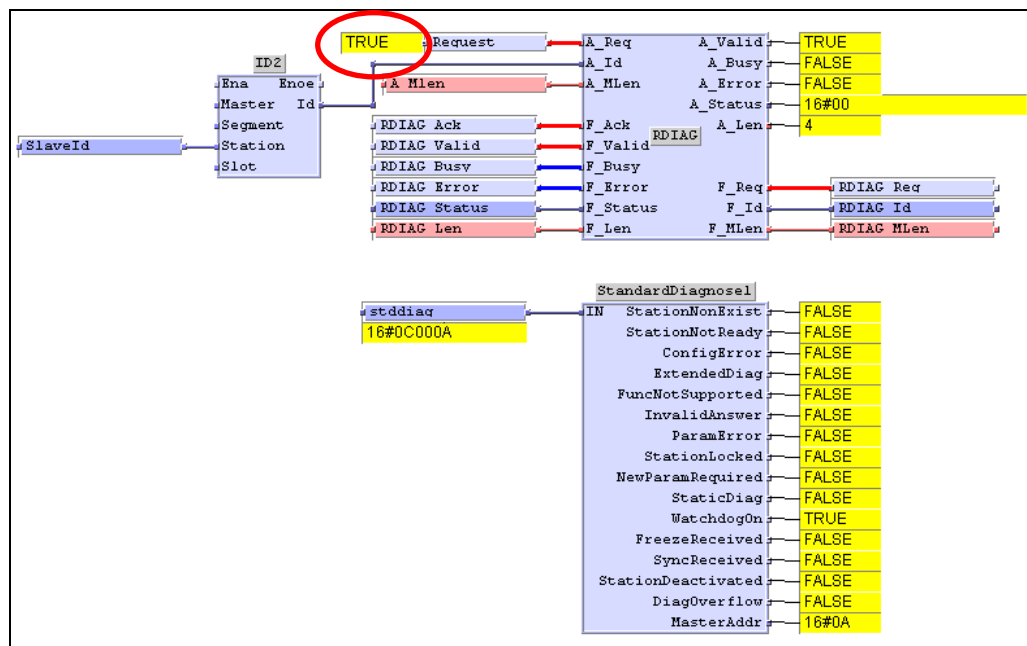


Bild 50: Online-Test

3 HIMatrix PROFIBUS-DP Slave


Dieses Kapitel beschreibt die Eigenschaften des HIMatrix PROFIBUS-DP Slave und die Menüfunktionen und Dialoge in **ELOP II Factory**, die zur Konfiguration des HIMatrix PROFIBUS-DP Slave benötigt werden.

Hinweis:	Für jeden HIMatrix-Steuerungs Typ steht die jeweilige Systemdokumentation mit den elektrischen und mechanischen Daten zur Verfügung. (Siehe Projektierungshandbuch HI 800 100 und Datenblätter der jeweiligen HIMatrix Steuerung).
-----------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.1 Benötigte Ausstattung und Systemanforderungen

HIMA ELOP II Factory	Ab Version 3.2.0
HIMatrix Steuerungen	F20 F30, F35 und F60 ab Hardware Revision: 00
Betriebssystemversionen der HIMatrix Steuerungen	COM BS ab Version 3.14 CPU BS ab Version 3.14
HIMatrix PROFIBUS-DP Slave Modul	Die HIMatrix Steuerung muss an der verwendeten seriellen Feldbusschnittstelle (FB1 oder FB2) mit einem optionalen HIMatrix PROFIBUS-DP Slave Modul ausgerüstet sein.
Lizenznummer	Wird nicht benötigt, Freischaltung durch Modul.

3.2 HIMatrix PROFIBUS-DP Slave Eigenschaften

Typ des PB Slaves	DP-V0
Übertragungsrate	9,6 kbit/s bis 12 Mbit/s
Busadresse	0 bis 125
Max. Anzahl Slaves	Pro Ressource kann immer nur ein HIMatrix PROFIBUS-DP Slave konfiguriert werden.
Prozessdatenmenge HIMA PB Slaves	DP-Output/ELOP-Import: max. 192 Bytes DP-Input/ELOP-Export: max. 240 Bytes Insgesamt jedoch: max. 256 Bytes
Protokoll Watchdog	<div style="display: flex; align-items: center;">  <div> <p>Ist der PROFIBUS-DP Slave im Zustand DATA EXCHANGE und die Verbindung zum PROFIBUS-DP Master geht verloren, dann wird dies vom PROFIBUS-DP Slave nach Ablauf der Watchdog-Zeit [ms] erkannt (Parameter im Master, siehe Kapitel 1.5.7.1).</p> <p>In diesem Fall wird das Statussignal <i>Daten gültig</i> auf FALSE und der Verbindungszustand auf OFFLINE gesetzt, siehe Kapitel 3.4.1.1. Die Eingangssignale vom PROFIBUS-DP Master werden ignoriert und stattdessen die Initialwerte verwendet.</p> </div> </div>

3.3 Funktion der FBx LED beim PROFIBUS-DP Slave

Die COM signalisiert den Zustand des lokalen PROFIBUS-DP-Slave-Protokolls mittels einer der jeweiligen Feldbusschnittstelle zugeordneten LED. Die Zustände dieser LED sind in der folgenden Tabelle dargestellt:

FBx LED	Beschreibung
AUS	Das PROFIBUS-DP-Slave-Protokoll ist nicht aktiv! D.h. die Ressource ist im Zustand STOPP oder es ist kein PROFIBUS-DP Slave konfiguriert.
Blinkt (0,5 Hz)	Kein Datenaustausch! Der PROFIBUS-DP Slave ist konfiguriert und aktiv. Es besteht aber keine Verbindung zum PROFIBUS-DP Master.
Bis COM BS V10: Blinkt (0,5 Hz) Ab COM BS V11: Blinkt (2,5 Hz)	Ablehnung der Parametrier-/Konfigurierdaten des PROFIBUS-DP Masters durch den PROFIBUS-DP Slave! D.h. die Konfigurationen des PROFIBUS-DP Masters und/oder des PROFIBUS-DP Slaves sind fehlerhaft bzw. passen nicht zu-einander.
AN	Das PROFIBUS-DP-Slave-Protokoll ist aktiv und befindet sich im Datenaustausch mit dem PROFIBUS-DP Master.

3.4 HIMatrix PROFIBUS-DP Slave Kontextmenü

Das Kontextmenü des HIMatrix PROFIBUS-DP Slave enthält die folgenden Funktionen.

PROFIBUS-DP Slave
Signale verbinden
Validieren
Neu
Import
Export
Kopieren
Einfügen
Löschen
Drucken
Eigenschaften

Zum Verhalten der Standard Menüfunktionen Validieren, Neu, Import, Export, Kopieren, Einfügen, Löschen und Drucken siehe Kapitel 1.4.2.

3.4.1 Menüfunktion *Signale verbinden*

Die Menüfunktion *Signale verbinden* aus dem Kontextmenü des PROFIBUS-DP Slave öffnet den Dialog *Signal-Zuordnungen*.

3.4.1.1 Register *Eingänge*

Im Register *Eingänge* wird festgelegt, welche Signale in die Steuerung eingelesen werden sollen.

Zudem befinden sich im Register *Eingänge* die folgenden Statussignale vom PROFIBUS-DP Slave, die im Anwenderprogramm ausgewertet werden können.

Signal	Beschreibung	Typ
Aktuelle Baudrate	<p>Baudrate, mit der das PROFIBUS-DP-Slave Protokoll aktuell arbeitet. Mögliche Werte (dezimal) sind:</p> <p>0 (noch keine Baudrate ermittelt)</p> <p>9600 (9,6 kBaud)</p> <p>19200 (19,2 kBaud)</p> <p>45450 (45,45 kBaud)</p> <p>93750 (93,75 kBaud)</p> <p>187500 (187,5 kBaud)</p> <p>500000 (500 kBaud)</p> <p>1500000 (1,5 MBaud)</p> <p>3000000 (3 MBaud)</p> <p>6000000 (6 MBaud)</p> <p>12000000 (12 MBaud)</p> <p>Standardwert: 0</p>	UDINT

Fehlercode	<p>Ist in dem PROFIBUS-DP-Slave Protokoll ein Fehler aufgetreten, so wird dies in diesem Signal übertragen. Es wird jeweils der aktuell aufgetretene Fehler angezeigt. Mögliche Werte (hexadezimal) sind:</p> <p>0x00: kein Fehler</p> <p>0xE1: falsche Parametrierung durch den PROFIBUS-DP Master</p> <p>0xD2: falsche Konfigurierung durch den PROFIBUS-DP Master</p> <p>Standardwert: 0x00</p>	DWORD
Protokollzustand	<p>Beschreibt den Zustand des PROFIBUS-DP Slave-Protokolls.</p> <p>Mögliche Werte (hexadezimal) sind:</p> <p>OFFLINE (0xE1): Die Steuerung ist vom Bus getrennt bzw. nicht aktiv.</p> <p>WAIT (0xD2): Die Steuerung wartet auf eine Konfiguration durch den PROFIBUS-DP Master.</p> <p>DATA EXCHANGE (0xC3): Die Steuerung tauscht zyklisch Daten mit dem PROFIBUS-DP Master aus.</p> <p>Standardwert: 0xE1</p>	BYTE
Slave-ID	<p>Dieses Signal enthält die PROFIBUS-DP Slave-ID der Steuerung, mit der sie am Bus arbeitet. Diese Slave-ID muss zuvor vom Anwender im PADT konfiguriert werden.</p> <p>Mögliche Werte (dezimal) sind:</p> <p>0-125: PROFIBUS-DP-Slave-ID</p> <p>Standardwert: 0xFF</p>	USINT
Master-ID	<p>Dies ist die ID des PROFIBUS-DP Masters, der den eigenen PROFIBUS-DP-Slave parametriert und konfiguriert hat.</p> <p>Mögliche Werte (dezimal) sind:</p> <p>0-125: ID des Masters</p> <p>255: Slave ist aktuell keinem Master zugeordnet</p> <p>Standardwert: 0xFF</p>	USINT
Daten gültig	<p>Ist das Statussignal <i>Daten gültig</i> auf TRUE gesetzt, dann hat der Slave gültige Import-Daten des Masters empfangen.</p> <p>Das Statussignal <i>Daten gültig</i> wird auf FALSE gesetzt wenn die Watchdog-Zeit beim Slave abgelaufen ist. PROFIBUS-DP Signale sind ungültig, Initialwerte liegen an!</p> <p>Standardwert: FALSE</p> <p>Hinweis:</p> <p>Wurde der Watchdog des Slaves durch den Master nicht aktiviert und die Verbindung geht verloren, so behält das Statussignal <i>Daten gültig</i> den Wert TRUE, da der PROFIBUS-DP Slave keine Möglichkeit hat, den Verbindungsverlust zu erkennen.</p> <p>Dieser Umstand ist bei der Verwendung dieses Signals unbedingt zu beachten!</p>	BOOL

Tabelle 85: Statussignale des HIMatrix PROFIBUS-DP Slave

3.4.1.2 Register *Ausgänge*

Im Register *Ausgänge* wird festgelegt, welche Signale aus der Steuerung exportiert werden sollen.

3.4.2 Menüfunktion *Eigenschaften*

Mit *Eigenschaften* im Kontextmenü des HIMatrix PROFIBUS-DP Slave wird der Dialog *Eigenschaften* geöffnet.

Das Dialogfenster enthält die Register *Allgemein* und *CPU/COM*.

3.4.2.1 Register *Allgemein*

Element	Beschreibung	Wert
Typ	PROFIBUS-DP Slave	Nur Anzeige
Stationsadresse	Stationsadresse des Slaves Die Stationsadresse des Slaves darf auf dem Bus nur einmal vorhanden sein.	Min: 0 Max: 125 Standard: 0
Schnittstelle	Feldbusschnittstelle, die für den PROFIBUS-DP Slave benutzt werden soll.	FB1, FB2, FB3
Baudrate [bps]	Baudrate, mit welcher der Bus betrieben wird. Mögliche Werte: 9600 (9,6 kBaud) 19200 (19,2 kBaud) 45450 (45,45 kBaud) 93750 (93,75 kBaud) 187500 (187,5 kBaud) 500000 (500 kBaud) 1500000 (1,5 MBaud) 3000000 (3 MBaud) 6000000 (6 MBaud) 12000000 (12 MBaud)	

Tabelle 86: Allgemeine Einstellungen des PROFIBUS-DP Slave im Dialog *Eigenschaften*

3.4.2.2 Register CPU/COM

Die Vorgabewerte für die Parameter sorgen für einen schnellen Datenaustausch der PROFIBUS DP Daten zwischen dem COM-Prozessor (COM) und der PROFIBUS-DP Slave Hardware der *HIMatrix* Steuerung.

Diese Parameter sollten nur dann geändert werden, wenn eine Reduzierung der COM-Auslastung für eine Anwendung erforderlich ist und der Prozess dies zulässt.



Die Änderung der Parameter wird nur für den erfahrenen Programmierer empfohlen.

Eine Erhöhung der Aktualisierungszeit der COM/ PROFIBUS-DP Hardware bedeutet auch, dass die tatsächliche Aktualisierungszeit der PROFIBUS-DP Daten erhöht wird. Die Zeitanforderungen der Anlage sind zu prüfen.

Beachten Sie auch den Parameter *Min. Slave Intervall [ms]* (siehe 1.4.3.2), der die minimale Aktualisierungszeit der PROFIBUS-DP Daten zwischen PROFIBUS-DP Master und PROFIBUS-DP Slave festlegt.

Element	Beschreibung
Refresh Rate [ms]	Aktualisierungszeit in Millisekunden, mit der die Daten des Protokolls zwischen COM und der PROFIBUS-DP Slave Hardware ausgetauscht werden. Wertebereich: 4 bis 1000 Vorgabewert: 10
In einem Zyklus	<div>Aktiviert Transfer der gesamten Daten des Protokolls von der CPU zur COM innerhalb eines Zyklus der CPU.</div> <div>Deaktiviert Transfer der gesamten Daten des Protokolls von der CPU zur COM, verteilt über mehrere CPU Zyklen zu je 900 Byte pro Datenrichtung. Damit kann eventuell auch die Zykluszeit der Steuerung reduziert werden.</div> <div>Vorgabewert: Aktiviert</div>

Tabelle 87: Eigenschaften des PROFIBUS-DP Slave

3.5 Beispiel: Konfiguration eines HIMatrix PROFIBUS-DP Slave

In diesem Beispiel tauscht ein HIMatrix PROFIBUS-DP Slave Signale mit einem HIMatrix PROFIBUS-DP Master aus.

Dabei wird gezeigt, wie für die Signale im HIMatrix PROFIBUS-DP Slave die dafür entsprechenden Module im HIMatrix PROFIBUS-DP Master angelegt und parametrisiert werden müssen.

3.5.1 Signale im HIMatrix PROFIBUS-DP Slave zuordnen

Hinweis Die Startadresse der HIMatrix PROFIBUS-DP Slave Ein- und Ausgangssignale beginnt immer bei 0. Erwartet der PROFIBUS-DP Master (eines anderen Herstellers) eine höhere Startadresse, müssen Dummy-Signale vor den Nutzsignalen eingefügt werden.

In diesem Beispiel werden die folgenden Signale im PROFIBUS-DP Slave angelegt:

Die Ausgangssignale bestehen aus **vier Signalen** mit insgesamt 11 Bytes.
Das Ausgangssignal mit dem niedrigsten Offset hat die Startadresse 0.

	Name	Typ	Grösse	Offset	Signal
1	PB_Slave_Master1	UINT	2	0	PB_Slave_Master1
2	PB_Slave_Master2	DWORD	4	2	PB_Slave_Master2
3	PB_Slave_Master3	DWORD	4	6	PB_Slave_Master3
4	PB_Slave_Master4	BYTE	1	10	PB_Slave_Master4

Gesamtgröße: 11 bytes

Bild 51: Register *Ausgänge* im HIMatrix PROFIBUS-DP Slave

Die Eingangssignale bestehen aus **zwei Signalen** mit insgesamt 3 Bytes.
Das Eingangssignal mit dem niedrigsten Offset hat die Startadresse 0. Die Offsets der Systemsignale (gegraut) werden nicht mitgezählt.

	Name	Typ	Grösse	Offset	Signal
1	Aktuelle Baudrate	UDINT	4	0	
2	Fehlercode	DWORD	4	4	
3	Protokollzustand	BYTE	1	8	
4	Slave-ID	USINT	1	9	
5	Master-ID	USINT	1	10	
6	Daten gültig	BOOL	1	11	
7	PB_Master_Slave1	UINT	2	12	PB_Master_Slave1
8	PB_Master_Slave2	BYTE	1	14	PB_Master_Slave2

Gesamtgröße: 15 bytes

Bild 52: Register *Eingänge* im HIMatrix PROFIBUS-DP Slave

3.5.1.1 PROFIBUS-DP Slave Konfiguration Prüfen

Klicken nach dem Anlegen der Signale auf die Schaltfläche **Neue Offsets** um die Offsets neu zu nummerieren.

Öffnen Sie nach der Konfiguration das Kontextmenü des PROFIBUS-DP Slaves und wählen Sie **Validieren**. In der *Fehler-Status-Anzeige* werden nach der Validierung der PROFIBUS-DP Slave Konfiguration eventuelle Fehler und Warnungen angezeigt.

3.5.2 Konfiguration des PROFIBUS-DP Slave im PROFIBUS-DP Master

3.5.2.1 Anlegen der HIMatrix PROFIBUS-DP-Module

Im PROFIBUS-DP Master muss die Anzahl der tatsächlich zu übertragenden Bytes konfiguriert werden. Dies geschieht durch Auswahl von *Modulen*, die in der GSD-Datei des PROFIBUS-DP Slave definiert sind.

Um die Anzahl der Bytes für die Eingangs- und Ausgangssignale des PROFIBUS-DP Master zu konfigurieren, wählt man mehrere Module, bis die physikalische Konfiguration des Slaves erreicht ist.

Die GSD-Datei des HIMatrix PROFIBUS-DP Slave heißt *hix100ea.gsd* und stellt die folgenden Module bereit:

PROFIBUS-DP Master Eingangs Module	Anzahl	Typ
DP-Input/ELOP-Export	1	Byte
DP-Input/ELOP-Export	2	Bytes
DP-Input/ELOP-Export	4	Bytes
DP-Input/ELOP-Export	8	Bytes
DP-Input/ELOP-Export	16	Bytes
DP-Input/ELOP-Export	1	Word
DP-Input/ELOP-Export	2	Words
DP-Input/ELOP-Export	4	Words
DP-Input/ELOP-Export	8	Words
DP-Input/ELOP-Export	16	Words
PROFIBUS-DP Master Ausgangs Module	Anzahl	Typ
DP-Output/ELOP-Import	1	Byte
DP-Output/ELOP-Import	2	Bytes
DP-Output/ELOP-Import	4	Bytes
DP-Output/ELOP-Import	8	Bytes
DP-Output/ELOP-Import	16	Bytes
DP-Output/ELOP-Import	1	Word
DP-Output/ELOP-Import	2	Words
DP-Output/ELOP-Import	4	Words
DP-Output/ELOP-Import	8	Words
DP-Output/ELOP-Import	16	Words

Bild 53: Module der HIMatrix GSD-Datei *hix100ea.gsd*

Hinweis Es ist nicht von Bedeutung, wie viele Module verwendet werden, um auf die erforderliche Anzahl an Bytes zu kommen, so lange die Anzahl von maximal 32 Modulen nicht überschritten wird.

Um die Konfiguration des PROFIBUS-DP Master nicht unnötig zu erschweren, sollte die Zahl der gewählten Module möglichst klein gehalten werden.

In diesem Beispiel werden im PROFIBUS-DP Master die folgenden passenden Module angelegt, um vom PROFIBUS-DP Slave **11 Bytes** zu empfangen und **3 Bytes** zu senden.

Nummerieren Sie die *HIMatrix* PROFIBUS-DP-Module bei **0** beginnend in aufsteigender Reihenfolge und ohne Lücken.

Die Reihenfolge der PROFIBUS-DP Module ist dabei für die Funktion nicht von Bedeutung. Zur besseren Übersicht sollten jedoch die DP-Input Module und die DP-Output Module geordnet angelegt werden.

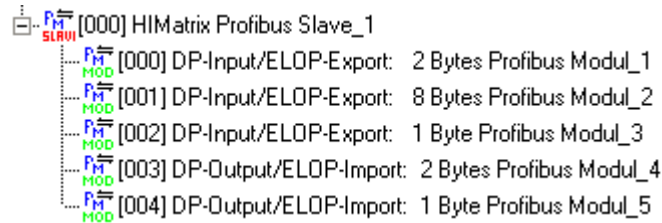


Bild 54: Passende Module aus der *HIMatrix* GSD-Datei für dieses Beispiel

3.5.2.2 Signal Zuordnung in den PROFIBUS-DP Modulen

Wählen Sie im Kontextmenü des jeweiligen PROFIBUS-DP Moduls **Signale verbinden** um den Dialog *Signal-Zuordnungen* zu öffnen.

Hinweis Die Summe der Signale in Byte muss mit der Größe des jeweiligen Moduls in Byte übereinstimmen

3.5.2.2.1 Signal Zuordnung in den Eingangsmodulen

In das Register *Eingänge* der Eingangsmodule *DP-Input/ELOP-Export* werden die Signale eingetragen, die der Master vom Slave empfängt.

Klicken Sie nach dem Anlegen der Signale auf die Schaltfläche **Neue Offsets** um die Offsets neu zu nummerieren.



Bild 55: Register *Eingänge* des Moduls [000] DP-Input/ELOP-Export: 2 Bytes

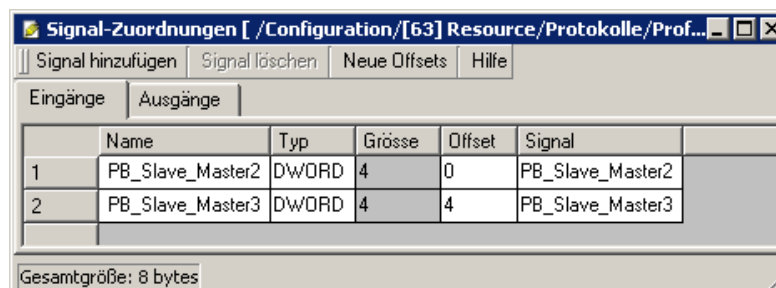


Bild 56: Register *Eingänge* des Moduls [001] DP-Input/ELOP-Export: 8 Bytes



Bild 57: Register *Eingänge* des Moduls [002] DP-Input/ELOP-Export: 1 Byte

3.5.2.2.2 Signal Zuordnung in den Ausgangsmodule

In das Register *Ausgänge* der Ausgangsmodule *DP-Output/ELOP-Import*: werden die Signale eingetragen, die der Master zum Slave sendet.

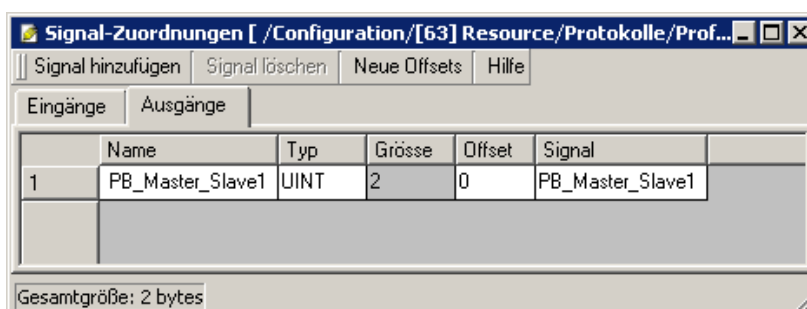


Bild 58: Register *Ausgänge* des Moduls [003] DP-Output/ELOP-Import 2 Bytes



Bild 59: Register *Ausgänge* des Moduls [004] DP-Output/ELOP-Import 1 Byte

3.5.2.3 Anlegen der Benutzerdaten im PROFIBUS-DP Master

In den Benutzerdaten werden die **Startadresse** und die **Anzahl der Signale** definiert. Die Anzahl der tatsächlich zu übertragenden Bytes wurde bereits durch die Auswahl von **Modulen** aus der GSD-Datei in Kapitel 3.5.2.1 konfiguriert.

Im HIMatrix PROFIBUS-DP Master wird das 32 Byte lange Benutzerdatenfeld des HIMatrix PROFIBUS-DP Slaves wie in Bild 60 dargestellt.

In diesem Dialog werden die Benutzerdaten zusätzlich in einer Tabelle dargestellt, um die Eingabe der Benutzerdaten **Startadresse** und **Anzahl der Signale** zu vereinfachen.

In diesem Beispiel werden im PROFIBUS-DP Master die folgenden passenden Benutzerdaten angelegt, um vom PROFIBUS-DP Slave **vier Signale** zu empfangen und **zwei Signale** zu senden.

Die Startadresse der Eingangs- und der Ausgangssignale beginnt jeweils bei **0**.

	Name	Wert	Standard
1	Range[I.1] Start index	0	Standard
2	Range[I.1] Signal count	4	Standard
3	Range[I.2] Start index	0	Standard
4	Range[I.2] Signal count	0	Standard
5	Range[I.3] Start index	0	Standard
6	Range[I.3] Signal count	0	Standard
7	Range[I.4] Start index	0	Standard
8	Range[I.4] Signal count	0	Standard
9	Range[O.1] Start index	0	Standard
10	Range[O.1] Signal count	2	Standard

Buttons: OK, Abbrechen, Übernehmen, Standard, Hilfe

Bild 60: 32 Byte Benutzerdaten des HIMatrix PROFIBUS-DP Slave im Master

3.5.2.4 PROFIBUS-DP Master Konfiguration Prüfen

Öffnen Sie nach der Konfiguration das Kontextmenü des PROFIBUS-DP Masters und wählen Sie **Validieren**.

In der Fehler-Status-Anzeige werden nach der Validierung der PROFIBUS-DP Master Konfiguration eventuelle Fehler und Warnungen angezeigt.

11.05.2007 12:05:49.453, Info: [Profibus DP Master_1] Validieren gestartet.
 11.05.2007 12:05:49.453, Fehler: [Profibus DP Master_1] Bitte wählen sie eine Schnittstelle für den Profibus DP Master
 11.05.2007 12:05:49.453, Info: [Profibus DP Master_1] Validieren beendet. Warnungen: 0, Fehler: 1.

Bild 61: Fehler-Status-Anzeige

4 HIMatrix PROFIBUS-DP Grundlagen

In diesem Kapitel werden die allgemeinen Grundlagen und die Umsetzung der PROFIBUS-DP Kommunikation für die HIMatrix Steuerungen beschrieben.

Wenn Sie an Hintergrundinformationen zur PROFIBUS-DP Kommunikation interessiert sind, dann finden Sie hier Informationen zu den

- Hardwaregrundlagen der seriellen Datenübertragung über RS-485
- PROFIBUS-DP Telegrammen und deren Schutzmechanismen
- PROFIBUS-DP Zyklen (azyklisch, zyklisch und isochron)

Für weitere Informationen zu PROFIBUS-DP verweist HIMA auf die folgenden Spezifikationen der PROFIBUS Nutzerorganisation e.V (Siehe www.profibus.de):

-PROFIBUS Technologie und Anwendung, Oktober 2002

-PROFIBUS Guideline-Ordner No. 2.182 Version 1.2, Juli 2001

Hinweis:	Für jeden HIMatrix-Steuerungs Typ steht die jeweilige Systemdokumentationen mit den elektrischen und mechanischen Daten zur Verfügung. (Siehe Projektierungshandbuch HI 800 100 und Datenblätter der jeweiligen HIMatrix Steuerung www.hima.com).
-----------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.1 DP-Leistungsstufen

Die Leistungsstufen von DP (Decentralized Peripherals) sind in der IEC 61158 spezifiziert.

DP-V0 Stellt die Grundfunktionalitäten von DP zur Verfügung. Dazu gehören der zyklische Datenaustausch sowie die stations-, modul- und kanalspezifische Diagnose.

DP-V1 Enthält Ergänzungen mit Ausrichtung auf die Prozessautomatisierung, vor allem den azyklischen Datenverkehr für Parametrierung, Bedienung, Beobachtung und Alarmbehandlung intelligenter Feldgeräte, parallel zum zyklischen Nutzdatenverkehr

DP-V2 Enthält weitere Ergänzungen wie z.B. den isochronen Datenaustausch.

4.2 PROFIBUS-DP Gerätetypen

Definition der PROFIBUS-DP Gerätetypen

- Ein Master Klasse 1 ist eine aktive Station und tauscht Nutzdaten mit dem Slave aus.
- Ein Master Klasse 2 ist eine aktive Station und meist ein PC (PADT). Mit dem Master Klasse 2 wird der PROFIBUS-DP konfiguriert und parametrier.
- Ein Slave ist eine passive Station und tauscht Nutzdaten mit dem Master aus. Ein Slave kann ein einfaches Feldgerät, aber auch eine komplexe Steuerung sein.

4.3 Hardwaregrundlagen der seriellen Datenübertragung

Auf der physikalischen Schicht von PROFIBUS-DP findet eine symmetrische Datenübertragung nach dem RS-485-Standard statt.

4.3.1 Grundlegende technische Eigenschaften der RS-485-Übertragung

In der folgenden Tabelle sind die grundlegenden technischen Eigenschaften der RS-485-Übertragung, die für den PROFIBUS-DP verwendet wird, dargestellt.

Bereich	Größen	Bemerkung
Netzwerk Topologie	Linearer Bus, aktiver Busabschluss an beiden Enden	Stichleitungen sind nur bei Baudraten bis 1,5 MBit/s zulässig
Medium	Abgeschirmtes verdrehtes Kabel	Schirmung kann abhängig von den Umgebungsbedingungen entfallen
Anzahl der Busteilnehmer	32 Busteilnehmer in jedem Segment ohne Repeater	Mit drei Repeatern erweiterbar bis zu max. 122 Slaves an einem Master.
Steckverbinder	9-pol-SUB-D Steckverbinder	Bei HIMA erhältlich

Tabelle 88: Grundlegende technische Eigenschaften der RS-485-Übertragung

Die Baudrate (Übertragungsgeschwindigkeit) ist im Bereich zwischen 9,6 kBit/s und 12 MBit/s wählbar und gilt für alle Busteilnehmer, die an den Bus angeschlossen sind. Die maximale Leitungslänge hängt von der gewählten Baudrate ab. Die Angaben zur Leitungslänge in der Tabelle 89 beziehen sich auf Kabeltyp A.

4.3.2 Reichweite in Abhängigkeit von der Baudrate

Baudrate	Reichweite pro Segment
9,6 kBit/s	1200 m
19,2 kBit/s	1200 m
93,75 kBit/s	1200 m
187,5 kBit/s	1000 m
500 kBit/s	400 m
1,5 MBit/s	200 m
3 MBit/s	100 m
6 MBit/s	100 m
12 MBit/s	100 m

Tabelle 89: Reichweite in Abhängigkeit von der Baudrate

Hinweis Eine Vergrößerung der Leitungslänge lässt sich mittels bidirektionaler Repeater erreichen. Maximal dürfen drei Repeater zwischen zwei Teilnehmer geschaltet werden. Somit ist eine Leitungslänge von 4,8 km möglich.
Bei zeitkritischen Anwendungen sollten nicht mehr als 32 Busteilnehmer angeschlossen werden. Für nicht zeitkritische Anwendung sind bis zu 123 Busteilnehmer (mit drei Repeatern) zulässig.

4.3.3 Busanschluss und Busabschluss

Das ankommende und das abgehende Datenkabel können direkt im Busanschlusstecker verbunden werden. Dadurch werden Stichleitungen vermieden und der Busanschlusstecker kann jederzeit, ohne Unterbrechung des Datenverkehrs, am Feldgerät auf- und abgesteckt werden.

In der IEC 61158 wird für PROFIBUS-DP ein 9-poliger Sub-D-Stecker empfohlen. Je nach Schutzart des Feldgerätes sind auch andere verfügbare Stecker erlaubt.

Die Steckerbelegung des 9-poligen Sub-D-Steckers ist in Bild 62 dargestellt. Am Feldgerät ist der Busanschluss als Buchse ausgelegt.

Der PROFIBUS-DP Busabschluss besteht aus einer Widerstandskombination, durch die ein definiertes Ruhepotential auf der Busleitung sichergestellt wird. Die Widerstandskombination ist in den PROFIBUS-DP Busanschlussteckern integriert und kann über Brücken oder Schalter aktiviert werden.

Der Busteilnehmer, an dem der Bus endet, sollte zudem eine 5-V-Spannung an Pin 6 anbieten.

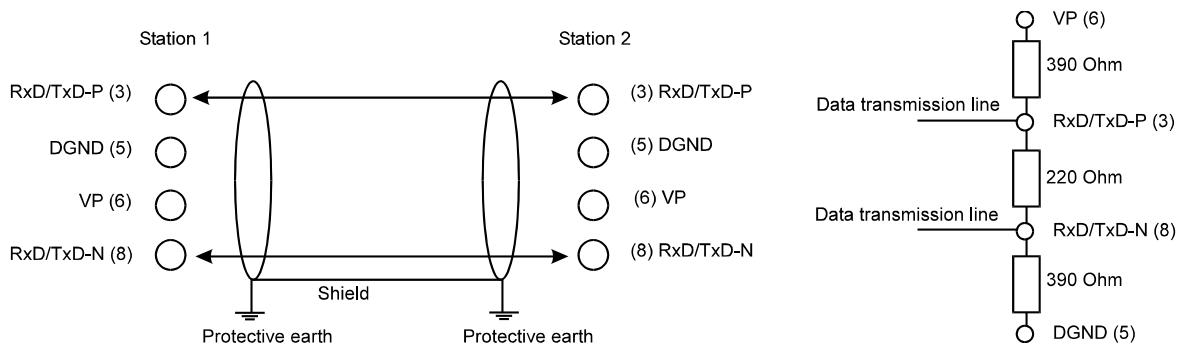


Bild 62: Busanschluss und Busabschluss, Pin-Belegung der Feldbus-Schnittstellen

Die Signale auf den Stiften 3, 5, 6 und 8 sind so genannte Mandatory-Signale und müssen zur Verfügung stehen.

Stift	Signal	Beschreibung
1	-	Nicht belegt
2	-	Nicht belegt
3	RxD/TxD-P	Empfangs-/Sendedaten-Plus (B-Leitung)
4	RTS	Richtungssteuerung für LWL-Umsetzer (TTL-Signal)
5	DGND	Datenübertragungspotential (Masse zu 5 V)
6	VP	Versorgungsspannung der Abschlusswiderstände-P (+5V)
7	-	Nicht belegt
8	RxD/TxD-N	Empfangs-/Sendedaten-Minus (A-Leitung)
9	-	Nicht belegt

Tabelle 90: Stiftbelegung der Schnittstellen FB1 und FB2 der HIMatrix Steuerungen

4.3.4 PROFIBUS-DP Buskabel

In der IEC 61158 werden zwei Busleitungen angegeben. Leitungstyp A kann für alle Übertragungsraten bis 12 Mbit/s genutzt werden. Leitungstyp B ist veraltet und sollte nicht mehr verwendet werden.

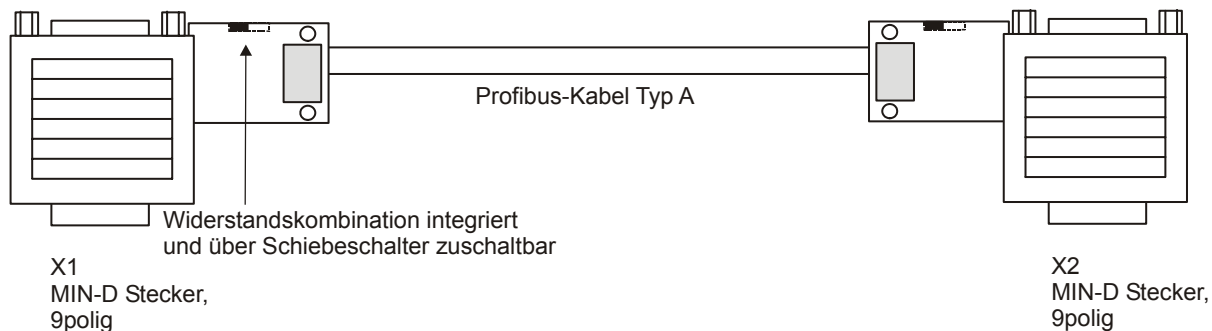


Bild 63: PROFIBUS-DP Buskabel Kabel Typ A mit Busanschlusssteckern

Als Übertragungsmedium ist eine geschirmte, symmetrische Zweidrahtleitung mit den folgenden Parametern vorgesehen:

Parameter	Kabeltyp A	
Wellenwiderstand	135 bis 165	Ω
Kapazitätsbelag	≤ 30	pf / m
Schleifenwiderstand	≤ 110	Ω / km
Aderndurchmesser	$> 0,64$	mm
Aderquerschnitt	$> 0,34$	mm ²

Tabelle 91: Parameter des PROFIBUS-DP RS-485-Kabels Typ A

4.3.5 Bus-Topologie

Alle Busteilnehmer werden an einem gemeinsamen Bus angeschlossen. Pro RS-485-Segment können bis zu 32 Busteilnehmer angeschlossen werden. Werden mehr Busteilnehmer benötigt, müssen weitere Segmente mit Repeatern angekoppelt werden. Anfang und Ende eines jeden Segments werden mit einem aktiven Busabschluss (bus termination) versehen. Der Busabschluss ist üblicherweise in den Busteilnehmern oder in den Busanschlussteckern zuschaltbar.

Für einen störungsfreien Betrieb muss sichergestellt werden, dass beide Busabschlüsse mit Spannung versorgt werden.

Die Busteilnehmer haben eindeutige Busadressen im Bereich 0..125. Bei mehr als 32 Busteilnehmern oder zur Erweiterung der Netzausdehnung müssen Leitungsverstärker (Repeater) eingesetzt werden, welche die einzelnen Bussegmente verbinden.

Pro eingesetztem Repeater reduziert sich die maximale Zahl der Busteilnehmer in diesem Segment um 1. Das bedeutet, dass in diesem Segment maximal 31 Busteilnehmer betrieben werden können. Nach der Norm sind insgesamt drei Repeater zulässig, so dass maximal 122 Slaves pro serielle Schnittstelle eines Master Klasse 1 angeschlossen werden können.

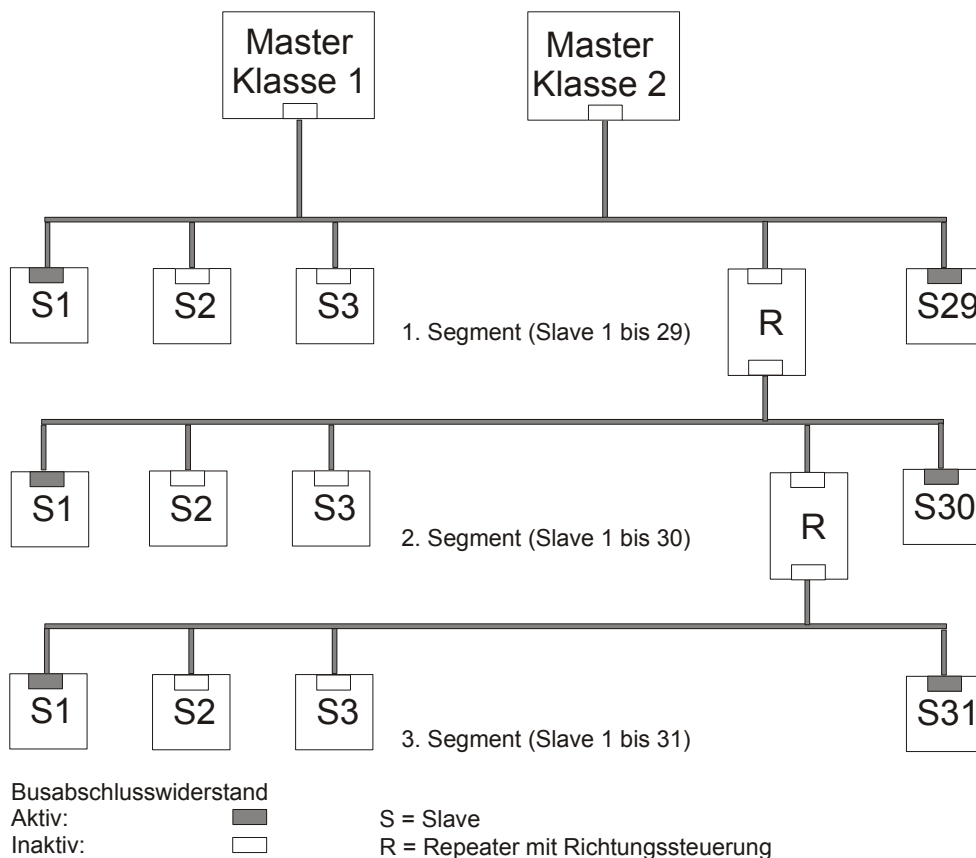


Bild 64: Bus-Topologie Linienstruktur mit Repeatern

Die Zeit, bis die Information eines Slaves beim Master verfügbar ist, steigt mit der Anzahl der Slaves am Bus an. Je mehr Slaves am Bus angeschlossen sind, umso schlechter werden die Reaktionszeiten des Systems.

Hinweis Bei Übertragungsraten $\geq 1,5$ Mbit/s sind Stichleitungen unbedingt zu vermeiden. Verwenden Sie darum nur geeignete Busanschlusstecker.

4.4 Die PROFIBUS-DP Telegramme

Für die PROFIBUS-DP Telegramme werden die beiden folgenden FDL Buszugriffsprotokolle verwendet:

FDL Buszugriffsprotokoll	Beschreibung
SDN (Send Data with No Acknowledge)	Daten ohne Bestätigung an einen einzelnen Teilnehmer, eine Gruppe oder an alle Teilnehmer senden.
SRD (Send and Request Data with Reply)	Daten mit Bestätigung an einen einzelnen Teilnehmer senden und empfangen.

Tabelle 92: Buszugriffsprotokolle in PROFIBUS-DP

Die PROFIBUS-DP Übertragung ist zeichenbasiert. Jedes übertragene Zeichen besteht aus elf Bits (ein UART-Zeichen, d.h. acht Datenbits und drei Steuerbits).

1 Startbit	8 Datenbit (1 Byte)	1 Paritätsbit	1 Stopbit
-------------------	----------------------------	----------------------	------------------

Bild 65: Ein UART-(Universal Asynchronous Receiver/ Transmitter) Zeichen

Die Übertragung der UART-Zeichen erfolgt blockweise in so genannten Telegrammen, d. h. einer Kette von UART-Zeichen. Die folgende Tabelle beschreibt die Funktionsbytes für die verwendeten Telegramme.

4.4.1 Funktionsbytes für die PROFIBUS-DP-Telegramme

Funktion	Beschreibung der Funktionsbytes	
SD1	Start Delimiter 1 (16#10)	Mit einem Startbyte (SD1 bis SD4) beginnt jedes der vier Telegrammformate.
SD2	Start Delimiter 2 (16#68)	
SD3	Start Delimiter 3 (16#A2)	
SD4	Start Delimiter 4 (16#DC)	
FCS	Frame Check Sequence	Enthält die Telegrammprüfsumme
DA	Destination Address	Stationsadresse des Empfänger
SA	Source Address	Stationsadresse des Sender
FC	Function Control	Kontrollbyte
L	Length Field	Anzahl der Datenbytes
Lr	Length Field redundant	Anzahl der Datenbytes
ED	End Delimiter (16#16)	Ende des Telegramms
SC	Single Char (16#E5)	Einzelzeichen für Quittierung

Tabelle 93: Beschreibung der Funktionsbytes für die PROFIBUS-DP-Telegramme

4.4.2 Verwendete PROFIBUS-DP Telegramme der HIMatrix Steuerungen

Telegramm mit fester Länge (L = 3) ohne Daten

SD1	DA	SA	FC	FCS	ED
10H	XX	XX	XXH	X	16H

Telegramm mit variabler Länge (L = 4 bis 249) mit bis zu 246 Byte Daten

SD2	L	Lr	SD2	DA	SA	FC	Daten[1...246]	FCS	ED
68H	X	X	68H	XX	XX	X	X....	X	16H

Telegramm mit fester Länge (L = 11) mit acht Byte Daten

SD3	DA	SA	FC	Daten[8]	FCS	ED
A2H	XX	XX	X	X....	X	16H

Token-Telegramm

SD4	DA	SA
-----	----	----

Bestätigung ohne Daten (Quittierung)

SC

4.4.3 Mögliche Stationsadressen in den Telegrammfeldern DA und SA

Adresse	Beschreibung
0 .. 125	Stationsadressen
126	Standardadresse für Busteilnehmer, die ihre endgültige Adresse dynamisch zugewiesen bekommen
127	Broad- oder Multicast-Adresse

Tabelle 94: Mögliche Stationsadressen in den Telegrammfeldern DA und SA

4.4.4 Schutzmechanismen der PROFIBUS-DP Telegramme zur Datensicherung

Die PROFIBUS-DP Telegramme ermöglichen eine hohe Übertragungssicherheit, die in der internationalen Norm IEC 870-5-1 festgelegt ist. Fehlerhafte Telegramme werden automatisch bis zu siebenmal wiederholt (siehe Tabelle 5 *Max. Anzahl Sendewdh.*).

PROFIBUS-DP ist in der Lage, ausgefallene Teilnehmer zu erkennen und der Konfiguration entsprechend zu reagieren.

Folgende Fehler werden erkannt:

- Start- und End-Delimiter-Fehler
- Falsche Telegrammlänge
- Falsches Zeichenformat
- Frame-Check-Byte-Fehler
- Protokollfehler

Ein Fehler führt zu folgenden Teilnehmerreaktionen:

- Ein fehlerhaft empfangener Funktionsaufruf wird grundsätzlich nicht ausgeführt. Der Funktionsaufruf muss wiederholt werden.
- Wird eine gestörte Antwort erhalten, muss der Funktionsaufruf ebenfalls wiederholt werden.

4.4.5 Die PROFIBUS-DP Buszugriffsverfahren

Das Buszugriffsverfahren stellt jedem Busteilnehmer ein definiertes Zeitfenster zur Verfügung, in dem der Busteilnehmer seine Kommunikationsaufgabe erfüllen muss.

4.4.5.1 Master/Slave-Protokoll

Die Buszuteilung zwischen einem PROFIBUS-DP Master und einem PROFIBUS-DP Slave wird über das Master/Slave-Verfahren sichergestellt.

Ein aktiver PROFIBUS-DP Master kommuniziert mit passiven PROFIBUS-DP Slaves. Der PROFIBUS-DP Master, der das Token besitzt, hat die Sendeberechtigung und kann mit dem ihm zugewiesenen PROFIBUS-DP Slaves kommunizieren. Der Master teilt einem Slave den Bus für bestimmte Zeit zu, innerhalb welcher der Slave antworten muss.

4.4.5.2 Token-Protokoll

Die Buszuteilung zwischen Automatisierungsgeräten (Master Klasse 1) und/oder Programmiergeräten (Master Klasse 2) wird über *Token Passing* sichergestellt.

Alle PROFIBUS-DP Master, die gemeinsam an einem Bus angeschlossen sind, bilden einen Token-Ring. Der aktive PROFIBUS-DP Master, der im Besitz des Tokens ist, übernimmt in dieser Zeit die Masterfunktion am Bus.

Die PROFIBUS-DP Master werden im Token-Ring nach aufsteigenden Stationsadressen geordnet und das Token wird in dieser Reihenfolge bis zum PROFIBUS-DP Master mit der höchsten Stationsadresse weitergegeben.

Dieser gibt das Token an den Master mit der niedrigsten Stationsadresse weiter, um den Token-Ring zu schließen.

Die Token-Umlaufzeit entspricht dem einmaligen Umlauf des Token über alle PROFIBUS-DP Master. Die Target Rotation Time T_{tr} ist die maximal erlaubte Zeit für einen Token-Umlauf.

Die Liste aller aktiven Busteilnehmer (LAS) enthält alle Master, die momentan am Bus aktiv sind. Sie dient dazu, neue Master einzutragen und ausgefallene Master zu löschen, ohne die Kommunikation am Bus zu stören.

Das Gap

Das Gap (Lücke) ist der Adressbereich von einer aktiven Station (Master) zur nächsten aktiven Station, nach aufsteigenden Busadressen geordnet.

Das Gap der letzten aktiven Station (HSA) umfasst zusätzlich den Bereich von Null bis zur ersten aktiven Station (m).

Dabei ist HSA (Highest Active Station) die höchste Adresse, die berücksichtigt wird.

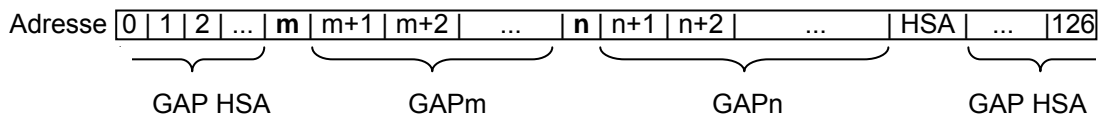


Bild 66: GAPS der Master im Token-Ring

Aufnahme eines Master in den Token-Ring

Jeder aktive PROFIBUS-DP Master pflegt seine Gap-Liste, indem er in aufsteigender Reihenfolge periodisch ein *FDL_Status_Req* an die einzelnen Adressen in seinem Gap schickt. Ist unter einer Adresse ein aktiver oder passiver PROFIBUS-DP Master zu erreichen, antwortet dieser.

Wird ein neuer PROFIBUS-DP Master an den Bus angeschlossen, verhält er sich passiv. Der neue PROFIBUS-DP Master baut bereits seine Liste aller aktiven Busteilnehmer (LAS) auf, bis er von einem PROFIBUS-DP Master, in dessen Gap er fällt, durch ein *FDL_Status_Req* angesprochen wird. Wenn er mit *Bereit* für den Token-Ring antwortet, wird er in den Token-Ring aufgenommen. PROFIBUS-DP Master, deren Adresse jenseits von HSA liegen, werden nicht angesprochen, also auch nicht in den Token-Ring aufgenommen.

Wenn nach dem Einschalten der Stromversorgung oder nach einem Token-Verlust, kein Verkehr auf dem Bus herrscht, warten die PROFIBUS-DP Master eine vordefinierte Zeit. Der PROFIBUS-DP Master mit der niedrigsten Adresse wird als erster aktiv und schickt das Token zweimal an sich selbst, als Zeichen für die anderen Busteilnehmer, dass er jetzt im alleinigen Tokenbesitz ist. Die anderen PROFIBUS-DP Master erhalten das Token dann per Token Passing.

4.5 Isochroner PROFIBUS-DP Zyklus (ab DP-V2)

Der PROFIBUS-DP Zyklus besteht hier aus einem festen, zyklischen und einem ereignisbedingten, azyklischen Telegrammteil.

Der azyklische Telegrammteil in einem PROFIBUS-DP Zyklus kann den PROFIBUS-DP Zyklus entsprechend verlängern, was in bestimmten Anwendungen, wie z.B. in der Antriebstechnik, unerwünscht ist.

Um eine konstante Zykluszeit (t_{const}) zu erreichen, wird im Master der Isochron-Mode aktiviert, bei dem der Parameter *Min. Slave Intervall [ms]* die konstante Zykluszeit (t_{const}) vorgibt. Der so parametrisierte isochrone PROFIBUS-DP Zyklus besitzt eine Taktgenauigkeit mit einer Abweichung von $< 10 \mu\text{s}$.

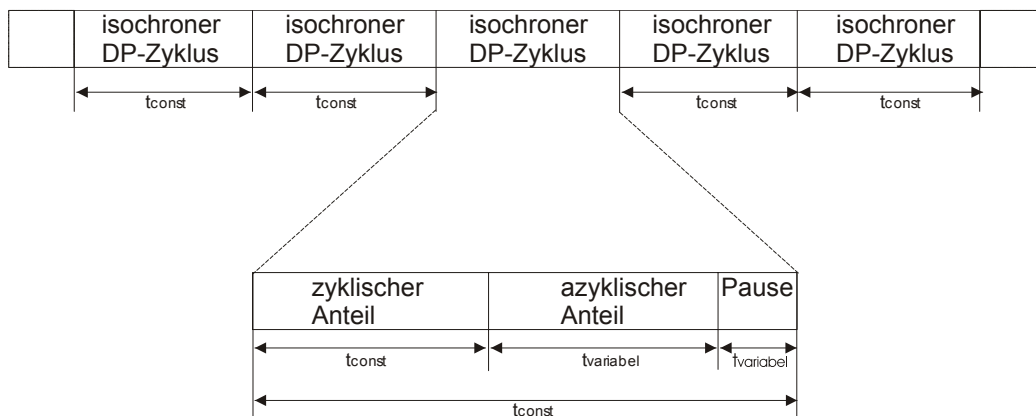


Bild 67: Isochroner PROFIBUS-DP Zyklus

Um den zyklischen Anteil zu ermitteln, muss der Anwender die minimale Token-Umlaufzeit berechnen.

Zusätzlich muss ein ausreichend großes Zeitintervall (typisch zwei- bis dreimal minimale Token-Umlaufzeit T_{tr}) für den azyklischen Anteil reserviert werden. Wird die reservierte Zeit nicht benötigt, wird eine Pause vor dem nächsten Zyklus eingelegt, um die Zykluszeit konstant zu halten. (siehe auch Kapitel 1.4.5, Berechnung der T_{tr}).

Hinweis Der Master wird über *Min. Slave Intervall [ms]* mit der vom Anwender ermittelte DP-Zykluszeit parametrisiert.
Damit der *Isochron Mode* wirksam ist, muss mindestens einer der beiden Parameter *Isochron Mode Sync* oder *Isochron Mode Freeze* im Master aktiviert werden.
An dem Bus darf dann nur ein Master im *isochron Mode* betrieben werden. Weitere Master sind nicht zulässig.

4.6 Zyklischer PROFIBUS-DP Zyklus (ab DP-V0)

Zyklische Master/Slave-Kommunikation zwischen einem Klasse-1-Master und einem Slave. Dies ist die normale PROFIBUS-DP Verbindung zum zyklischen Austausch von E/A-Daten.

Die Master/Slave-Kommunikation besteht aus den Phasen

- Verbindungsaufbau
- Datenaustausch und
- Verbindungsabbau.

Beim Verbindungsaufbau fordert der Master vom Slave ein Diagnosetelegramm an, um festzustellen, ob der Slave bereit ist. Ist der Slave bereit, sendet der Master dem Slave ein Parametertelegramm mit gesetztem *Lock Bit*. Der Slave antwortet mit einer Bestätigung (SC) und ist ab jetzt für andere Master gesperrt.

Danach sendet der Master dem Slave die Konfigurationsdaten, die der Slave mit einer Bestätigung (SC) beantwortet.

Am Ende des Verbindungsaufbaus fordert der Master vom Slave eine Diagnose an, um zu überprüfen, ob Parametrierung und Konfiguration fehlerfrei waren. Wird kein Fehler festgestellt bleibt die Verbindung erhalten.

Zum Verbindungsabbau werden Parametrierdaten mit gesetztem *Unlock Bit* gesendet. Der Slave antwortet wieder mit einer Bestätigung (SC).

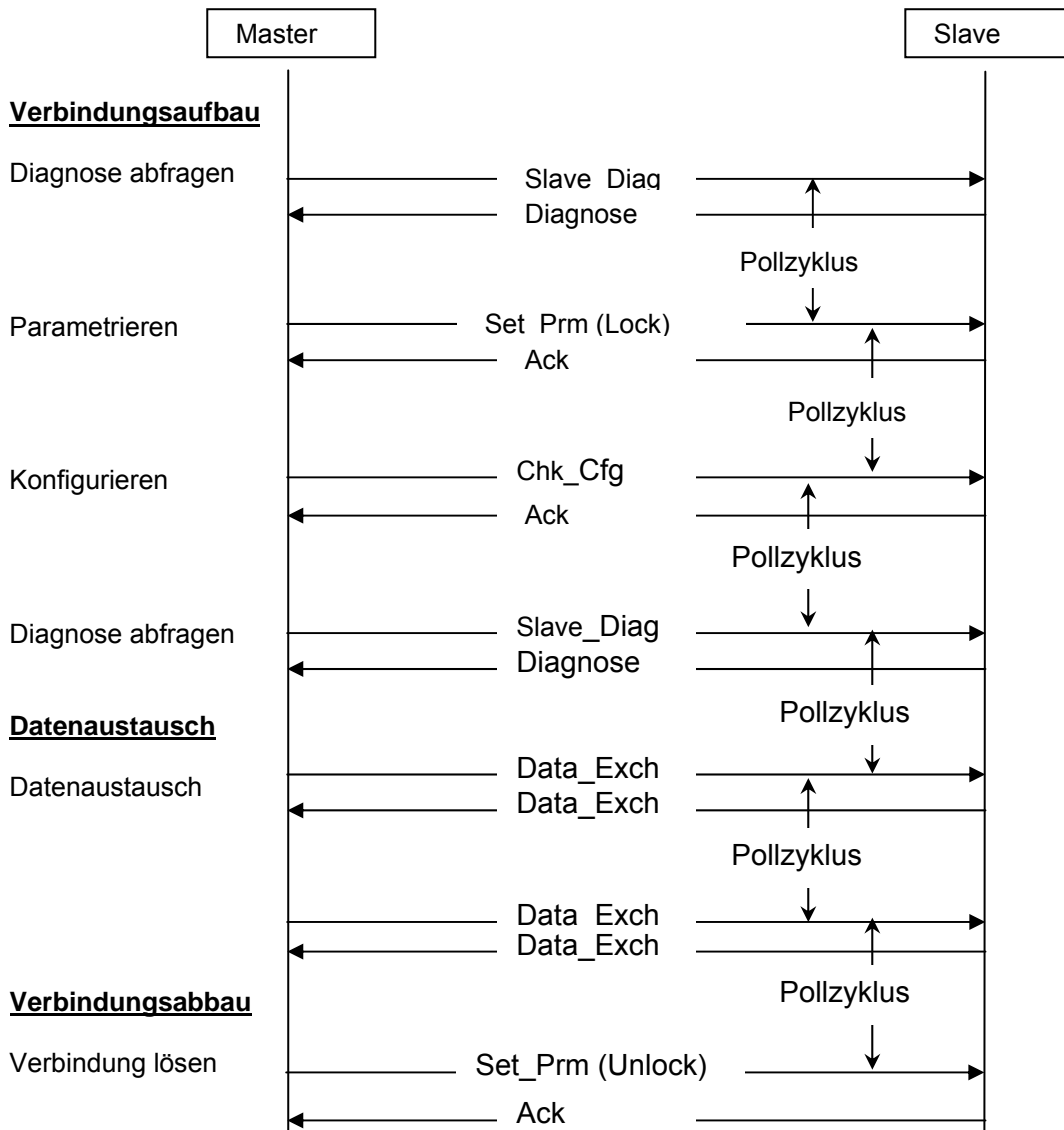


Bild 68: Master-Slave-Kommunikation

4.6.1 Pollzyklus

Der PROFIBUS-DP Master kommuniziert mit seinen Slaves in Pollzyklen. Jeder Pollzyklus besteht aus einem zyklischen Teil, einem optionalen azyklischen Teil, sowie der Pflege der Gap-Liste und der Token-Weitergabe.

- Ein Pollzyklus beginnt damit dass der Master seinen Zustand an alle Slaves mitteilt.
- Im zyklischen Teil wird von jedem Slave genau ein Telegramm (entweder Daten, Parametrierung, Konfiguration oder Diagnose) angefordert.
- Anschließend wird eine einzelne Adresse *Station_x* im Gap des *Master_1* per *FDL_Status_Req* (Telegramm zum Erkennen eines neuen Master) abgefragt.
- Danach wird die azyklischen Kommunikation ausgeführt.
- Am Ende eines Pollzyklus wird das Token weitergegeben. Gibt es keinen weiteren Master am Bus, gibt der Master das Token an sich selbst.

Die Zeiten *Tid*, *Tsl* usw. werden im anschließenden Kapitel erklärt.

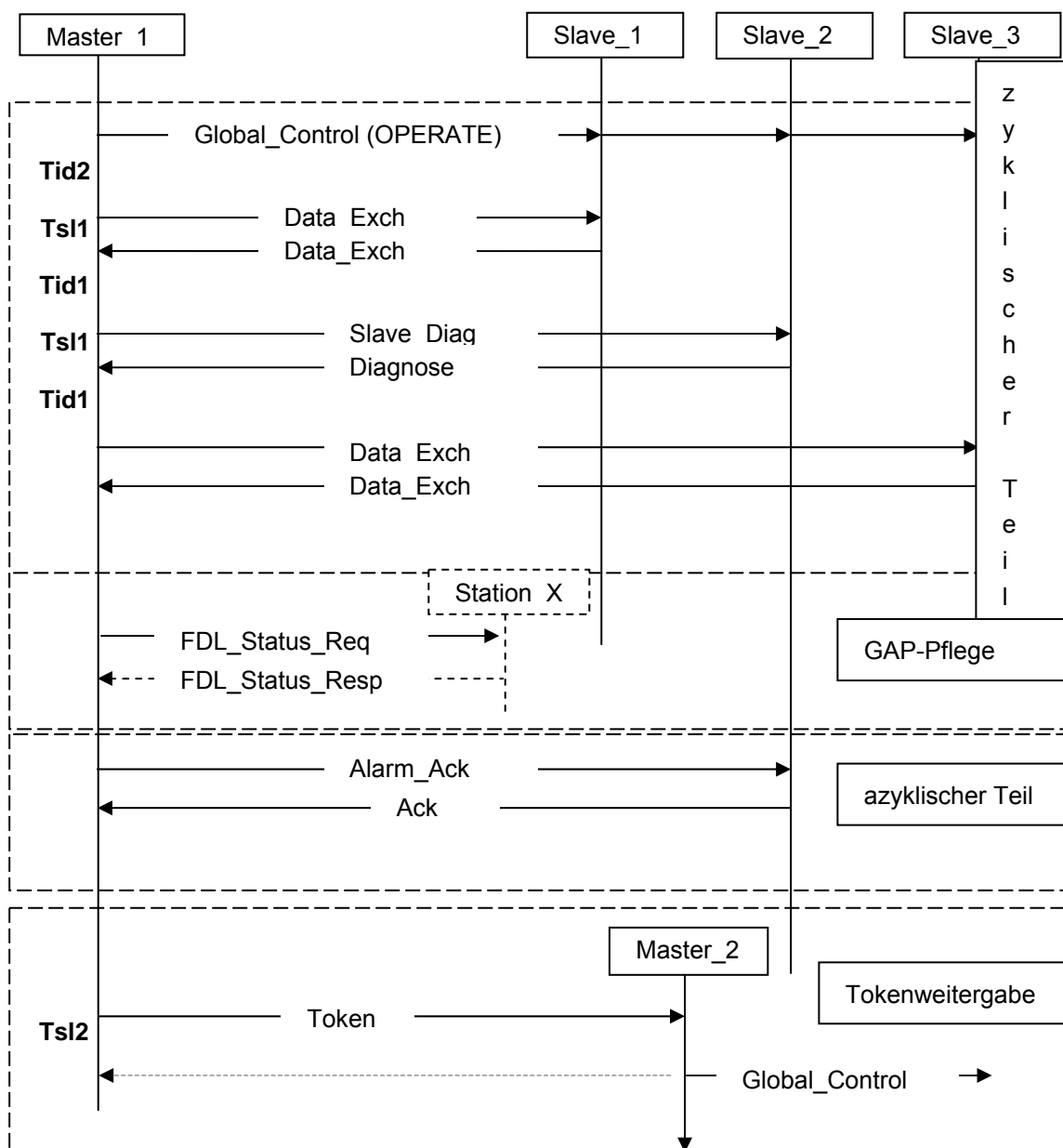


Bild 69: Pollzyklen zwischen Master und Slaves

4.6.2 Zeiten des Pollzyklus

4.6.2.1 Idle Time (Tid)

Die Idle Time ist die Zeit, die bei einer Station zwischen zwei Frames oder zwei unbestätigten Paketen vergeht.

$$Tid \geq T_{syn} + T_{sm}$$

Es werden zwei Zeiten unterschieden:

Tid1

Zeit, die nach dem Empfang eines Antwort- oder Token-Frames und dem Aussenden des nächsten Frames verstreicht.

$$Tid1 = \max(T_{syn} + T_{sm}, MinT_{sdr}, T_{sdi})$$

Tid2

Zeit, die nach dem Aussenden eines unbestätigten Paketes (SDN, siehe Tabelle 92) und dem Aussenden des nächsten Paketes verstreicht.

$$Tid2 = \max(T_{syn} + T_{sm}, MaxT_{sdr})$$

4.6.2.2 Slot Time (Tsl)

Die Slot Time ist die maximale Zeit, welche eine Station nach dem Aussenden eines Requests auf den Empfang des ersten Zeichens (11 Bit) einer Antwort wartet.

$$Tsl = \max(Tsl1, Tsl2)$$

Es werden zwei (theoretische) Zeiten unterschieden:

Tsl1

Zeit, welche die sendende Station nach einem Request auf den Beginn des Antwortpaketes wartet.

$$Tsl1 = MaxT_{sdr} + 2 * T_{td} + T_{sm} + 11$$

Tsl2

Zeit, welche die sendende Station nach einer Token-Weitergabe auf den Beginn des nächsten Paketes wartet. Der Token-Absender erkennt daran, dass das Token angekommen ist.

$$Tsl2 = Tid1 + 2 * T_{td} + T_{sm} + 11$$

4.6.2.3 Synchronization Time (Tsyn)

Minimale Zeit, die das Übertragungsmedium im Leerlauf (Idle, binäre eins) verharren muss, bevor ein neuer Request- oder Token-Frame beginnt.

$$T_{syn} = 33 \text{ tBit}$$

4.6.2.4 Station Delay Time (Tsdx)

Zeit, die zwischen dem Empfang des letzten Bits eines Datenpakets und dem Aussenden des ersten Bits des nächsten Pakets verstreichen kann (vom Übertragungsmedium aus gesehen).

Es gibt mehrere Varianten:

Station Delay of Initiators (Tsd_i)

Zeit, die beim Initiator (Master) zwischen dem Empfang eines Antwortpaketes und dem Aussenden des nächsten Request- oder Token-Paketes verstreichen kann.

Station Delay of Responders (Tsd_r)

$$\text{MinTsd}_r \leq Tsd_r \leq \text{MaxTsd}_r$$

Zeit, die beim Responder (Slave) zwischen dem Empfang eines Request-Paketes und dem Aussenden des Antwortpaketes verstreichen kann.

Für Tsd_r gibt es zwei Grenzwerte:

MinTsd_r (Minimum Station Delay of Responders)

Zeit, die mindestens verstreichen muss

MaxTsd_r (Maximum Station Delay of Responders)

Zeit, die maximal verstreichen darf

4.6.2.5 Quiet Time (Tqui)

Modulatorausklingzeit, während der Senden und Empfangen deaktiviert sein sollte. Notfalls muss MinTsd_r erhöht werden, falls folgende Bedingung nicht erfüllt ist.

$$T_{qui} < \text{MinTsd}_r$$

4.6.2.6 Safety Margin (Tsm)

Sicherheitszuschlag

$$T_{sm} = 2 + 2 * T_{set} + T_{qui}$$

4.6.2.7 Time-out Time (Tto)

Zeit zum Überwachen der Busaktivität. Verstreicht diese Zeit ohne Busaktivität, wird der Bus als tot betrachtet (Token-Verlust).

$$T_{to} = 6 * T_{sl} + 2 * n * T_{sl}$$

Bei aktiven Busteilnehmern ist n die Busadresse, bei passiven Busteilnehmern ist $n = 130$.

4.6.2.8 Weitere Zeiten des Pollzyklus

Setup Time (Tset):

Zeit, die benötigt wird, um auf ein Ereignis zu reagieren (Interrupt-Handler).

Transmission Delay Time (Ttd):

Zeit, die für die Übertragung auf dem Medium - incl. Repeatern - benötigt wird.

Target Rotation Time (Ttr):

Projektierte Zeit für einen Token-Umlauf.

Real Rotation Time (Trr):

Tatsächliche Token-Umlaufzeit.

Min Slave Interval (Msi):

Mindestzeit, die zwischen zwei zyklischen Abfragen eines Slave verstreichen muss.

Literaturverzeichnis

[1] Erste Schritte ELOP II Factory

HIMA GmbH Brühl, 2006: HI 800 005

[2] Online Hilfe im ELOP II Factory Hardware-Management

HIMA GmbH Brühl, 2006

[3] HIMA HIMatrix Projektierungshandbuch

HIMA GmbH Brühl, 2009: HI 800 100

[4] HIMA Feldbus-Kommunikationsmodule

HIMA GmbH Brühl, 2004: HI 800 128

[5] Manfred Popp: PROFIBUS-DP/DP-V1-Grundlagen, Tipps und Tricks für Anwender,

Hüthig Verlag Heidelberg, 2000 ISBN 3-7785-2781-9

[6] PROFIBUS Nutzerorganisation e.V.: PROFIBUS Technologie und Anwendung,

Oktober 2002

[7] PROFIBUS Nutzerorganisation e.V.: PROFIBUS Guideline-Ordner No. 2.182

Version 1.2, Juli 2001

<http://www.PROFIBUS.de/>

[8] EN 50170 Band 2/3 PROFIBUS 50170 A1 + A3

HIMA
...die sichere Entscheidung.



~~www~~ ~~www~~ HIMA Paul Hildebrandt GmbH
Industrie-Automatisierung
Postfach 1261 • 68777 Brühl

Telefon: (06202) 709-0 • Telefax: (06202) 709-107
E-mail: info@hima.com • Internet: www.hima.de

(0908)