



Modulares Reglersystem KS VARIO



Schnittstellenbeschreibung
PROFIBUS - DP
9499 040 69718

gültig ab: 12/2004

SIMATIC® ist ein eingetragenes Warenzeichen der Siemens AG

 SIMATIC® eingetragenes Warenzeichen der Siemens AG

 PROFIBUS® ist ein eingetragenes Warenzeichen der

PROFIBUS Nutzerorganisation (PNO)

BluePort® ist ein eingetragenes Warenzeichen der PMA Prozeß- und Maschinen-Automation GmbH

BlueControl® ist ein eingetragenes Warenzeichen der PMA Prozeß- und Maschinen-Automation GmbH

© PMA Prozeß- und Maschinen-Automation GmbH -

Printed in Germany

Alle Rechte vorbehalten.

Ohne vorhergehende schriftliche Genehmigung ist der Nachdruck oder die auszugsweise fotomechanische oder anderweitige Wiedergabe dieses Dokumentes nicht gestattet.

Dies ist eine Publikation von PMA Prozeß- und Maschinen Automation

Postfach 310229

D-34058 Kassel

Germany

Inhalt

1. Allgemeines	5
1.1. Lieferumfang	6
2. Hinweise zum Betrieb	7
2.1. Anschluss der Schnittstelle	7
2.2. Bedeutung der Anzeige-LEDs am Buskoppler	7
2.3. Forcing	8
2.4. Fail-safe	8
3. Prozessdaten	9
3.1. Konfigurierbare Prozessdatenmodule	10
3.2. Definition der zu übertragenden Werte im Engineering Tool "BlueControl"	12
3.3. Aufbau des Daten-Cache-Speichers im Buskoppler	14
3.4. Datenaustausch über variables Multiplexermodul	15
3.4.1 Datentransfer vom Profibus-Master zum Buskoppler	15
3.4.2 Datentransfer vom Buskoppler zum Profibus-Master	16
3.5. Datenaustausch über Multiplexermodule mit fester Datenlänge	17
4. Parameterübertragung via Parameterkanal in DP-V0	18
4.1. Nachrichtenelemente	19
4.2. Allgemeiner Übertragungsaufbau	20
4.3. Ablauf - Daten schreiben	21
4.4. Ablauf - Daten lesen	21
4.5. Beispiele	22
4.5.1 Beispiel 1: Einzelzugriff, Lesen, Integerwert	22
4.5.2 Beispiel 2: Blockzugriff, Lesen, Floatwert	22
4.5.3 Beispiel 3: Einzelzugriff, Schreiben, Integerwert	23
4.5.4 Beispiel 4: Blockzugriff, Schreiben, Floatwert	24
5. Parameterübertragung via DP-V1	25
6. User-Parametrierung, Diagnose	26
6.1. PROFIBUS-DP Diagnoseinformationen	27
6.1.1 Standard - Diagnosenachricht	27
6.1.2 Gerätespezifische Diagnose	28
7. Schnelleinstieg, am Beispiel einer SIMATIC® S7	29
7.1. Benutzung des Parameterkanals	30
7.1.1 Einfacher Zugriff	30
7.2. Direkter Zugriff	31
7.3. Benutzung der Multiplexerfunktion	31

8. Funktionsbausteine für SIMATIC® S7	32
8.1. Aufbau FB106, FB107	32
8.2. Aufbau FB108	34
9. Anhang	36
9.1. Anlagenaufbau	36
9.1.1 Minimalausbau einer PROFIBUS-Anlage	36
9.1.2 Maximalausbau einer PROFIBUS-Anlage	36
9.1.3 Leitungsverlegung innerhalb von Gebäuden	36
9.1.4 Erdfreier Aufbau	37
9.2. Adressübersicht	37
9.3. Begriffe	40
9.4. GSD Datei	41

1. Allgemeines

Das modulare Reglersystem KS VARIO erlaubt den Anschluss von verschiedenen Feldbusschnittstellen. Hierzu wird der jeweilige Buskoppler als Kopfstation für ein Reglersystem benutzt.

Über einen dieser Buskoppler wird über eine frontseitige Schnittstelle (Sub-D-Stecker) der PROFIBUS-DP unterstützt. Hierüber wird eine Übertragung aller Prozeß-, Parameter- und Konfigurationsdaten ermöglicht.

Diese Kommunikationsschnittstelle ermöglicht Verbindungen zu übergeordneten Steuerungen, Visualisierungstools etc..

Eine weitere, standardmäßig immer vorhandene Schnittstelle befindet sich auf den Reglerbausteinen KS VARIO. Diese vollwertige RS232 Schnittstelle dient dem Anschluß des 'BlueControl'-Tools, das auf einem PC abläuft.

Die Kommunikation erfolgt nach dem Master/Slave-Prinzip. Der KS VARIO ist immer Slave.

Das Leitungsmedium sowie die physikalischen und elektrischen Eigenschaften der Schnittstelle:

Netzwerk Topologie

Linearer Bus mit aktivem Busabschluss an beiden Enden. Bei Übertragungsraten ab 1,5 Mbit/s sind Stichleitungen unbedingt zu vermeiden.

Übertragungsmedium

geschirmte, verdrehte 2-Drahtleitung Die Eigenschaften der Busleitung sind in der EN 50 170 spezifiziert. Mit dem Leitungstyp A können alle Übertragungsraten bis 12 Mbit/s genutzt werden. Neben der Standardleitung sind auch Leitungen für Erdverlegung, Girlandenaufhängung und Schleppkabel verfügbar. Die empfohlenen Leitungsparameter sind wie folgt:

	Leitungstyp A
Wellenwiderstand in Ω	135 ... 165 bei 3 ... 20 MHz
Betriebskapazität (pF/m)	<30
Schleifenwiderstand Ω /km	< 110
Aderndurchmesser (mm)	> 0,64
Aderquerschnitt (qmm)	> 0,34

Leitungslängen

Die maximale Leitungslänge ist abhängig von der verwendeten Übertragungsrate. Die Baudrate wird durch die Masterkonfiguration vorgegeben und wird automatisch vom KS VARIO erkannt. Durch Einsatz von Repeatern kann die Leitungslänge vergrößert werden.

Baudraten

Automatische Baudratenerkennung	Baudrate	Maximale Leitungslänge
	9,6 / 19,2 / 31,25 / 45,45 / 93,75 kbit/s	1200 m
	187,5 kbit/s	1000 m
	500 kbit/s	400 m
	1,5 Mbit/s	200 m
	3 / 6 / 12 Mbit/s	100m

Schnittstelle RS485 ; Vorort montierbar

Adressierung 0 ... 126 (Auslieferungszustand: 01)

Die Einstellung erfolgt über die 2 Drehschalter des Buskoppler oder über das Engineering-Tool (Blue Control). Bei Einstellung der Adresse über BlueControl müssen die 2 Adress-Drehschalter die Schalterstellung "00" aufweisen.

Die Einstellung OFF schaltet die Busfunktion ab (Betrieb als wäre keine Schnittstelle vorhanden, inkl. Abschalten von Busfehlermeldungen). Beim Wechsel von OFF auf eine gültige Adresse wird das DP-System zurückgesetzt (Reset des DP-Prozessorsystems).

Anzahl der Stationen 32 Geräte in einem Segment. Mit Repeatern auf 126 erweiterbar.

Diagnose und Überwachung Schnelle Übertragung von Prozesswerten

Anzeige von Busfehlern
- Errormeldung
- LED

Besonderheiten Konfigurierbare Prozessdatenmodule

Direktes Lesen und Schreiben von Ein- und Ausgängen
Forcing von Eingängen
Back-up Reglerfunktion
Einfache Anbindung selbst an kleine Steuerungen

1.1. Lieferumfang

Das Engineering Set besteht aus:
Diskette

Pma_1030.gsd GSD-Datei

Pma_sup.arj Step7® Funktionsbaustein als S7- Bibl.

KSvario.zip Beispielprojekt in Step7® für S7-300

Dev_KSvario-1.bct Beispielkonfiguration für BlueControl®

Schnittstellenbeschreibung für PROFIBUS-DP – Prozessdaten und Parameterdaten

- Das Dokument "KS VARIO PROFIBUS-DP" (9499-040-69718) gibt grundlegende Erläuterungen zum Anschluss des KS VARIO an PROFIBUS-DP - Netze. Es enthält Hinweise für den zyklischen Prozessdatenaustausch.
- Weiterhin werden zusätzliche Übertragungsmöglichkeiten von Prozesswerten, Parametern und Konfigurationsdaten über den Parameterkanal und über Multiplexbetrieb beschrieben

GSD-Datei Die GSD-Datei liegt sowohl als Standard-File mit englischen Texten (PMA_1030.gsd) als auch mit deutschen Texten (PMA_1030.gsg) vor. Wenn Sie mit Ihrem PROFIBUS - Masterkonfiguration mit deutschen Texten arbeiten möchten, installieren Sie bitte die Datei PMA_1030.gsg.

2. Hinweise zum Betrieb

2.1. Anschluss der Schnittstelle

Der PROFIBUS wird an die frontseitige Sub-D-Schnittstelle des Buskopplers angeschlossen.
Die physikalischen Signale basieren auf RS485-Schnittstellen.

Der Aufbau entsprechender Kabel ist vom Anwender durchzuführen. Dabei sind die allgemeinen Kabelspezifikationen nach EN 50170 Vol.2 zu beachten.

Verlegen von Leitungen

Bei der Leitungsverlegung sind die vom Lieferant der Masterbaugruppe gemachten allgemeinen Hinweise zum Verlegen von Leitungen zu beachten:

Leitungsführung innerhalb von Gebäuden (innerhalb und außerhalb von Schränken)

Leitungsführung außerhalb von Gebäuden

Potenzialausgleich

Schirmung von Leitungen

Maßnahmen gegen Störspannungen

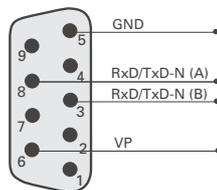
Länge der Stichleitung

Busabschlusswiderstände sind nicht in dem VARIO-Buskoppler enthalten, sondern müssen im Bedarfsfall über den externen Anschlussstecker realisiert werden.

Spezielle Hinweise zum Verlegen von PROFIBUS- Kabeln sind der von der PROFIBUS Nutzerorganisation (PNO) Technischen Richtlinie "Aufbaurichtlinien für PROFIBUS-DP/FMS" (Best-Nr. 2.111 [dt]; 2.112 [engl.]) zu entnehmen.

Belegung Sub D

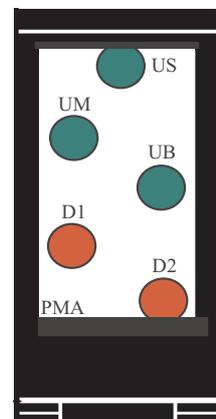
Der Anschluss erfolgt über eine 9-polige Sub-D-Buchse.



2.2. Bedeutung der Anzeige-LEDs am Buskoppler

LEDs

LED-Nr.	LED-Farbe	Funktion
US	grün	Segmentspannung U vorhanden
UM	grün	Modulspannung U vorhanden
UB	grün	Kopplerspannung U vorhanden
D1	rot	Verbindung Buskoppler zur Steuerung: - ok = LED aus - Konfigurationsfehler = LED blinkt - Keine Verbindung = LED an
D2	rot	Verbindung Buskoppler zum Regler: - ok = LED aus - Fehler = LED blinkt



2.3. Forcing

Eingänge

Alle physikalischen Eingänge können über den PROFIBUS-DP überschrieben werden (konfigurierbar). Damit ist es z.B. möglich, den Istwert über Remote I/O (z.B. VARIO I/O-System) zu erfassen und über den Bus vorzugeben.

Ausgänge

Bei Forcing der Ausgänge, ist die Einstellung der Fail-safe Funktion zu beachten. Bei eingestelltem Fail-safe - Verhalten "zero" werden alle Ausgänge bei Busfehler oder Master-Stop auf null gesetzt, andernfalls behalten sie ihren alten Wert bei.

2.4. Fail-safe

Über die User-Parametrierung 'Fail-safe' wird das Verhalten des Gerätes bei Busausfall bzw. 'Bus-Stop' des Masters festgelegt.

Busausfall

Bei Busausfall arbeitet das Gerät nach folgenden Regeln.

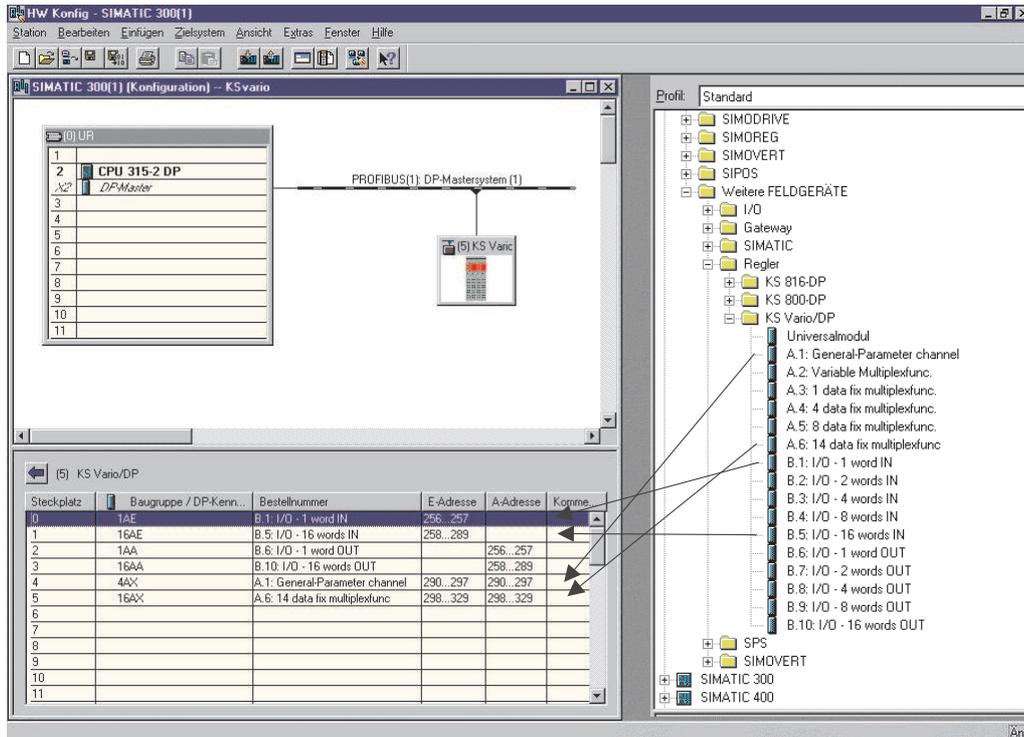
Fail-safe	Reaktion bei Busausfall oder Master-Stop
Last value	Weiterarbeiten mit den zuletzt gesendeten Werten
	Geforcte analoge Eingänge werden auf FAIL gesetzt
zero	Geforcte analoge Eingänge werden auf FAIL 1) gesetzt
	geforcte digitale Eingänge werden auf null gesetzt
	Geforcte Ausgänge werden auf null gesetzt

3. Prozessdaten

Der Anwender kann die Übertragung der Prozessdaten aus einer vorgegebenen Auswahl von Modulen zusammenstellen, um seine Anforderungen an Übertragungswerte, Speicherplatz und Übertragungszeit flexibel realisieren zu können. Die Konfiguration erfolgt über das jeweilige Buskonfigurierungstool des Busmasters.

Definition zyklischer Daten in BlueControl

Welche Prozessdaten und Parameter zyklisch über den Profibus übertragen werden sollen, ist absolut wahlfrei und kann mit wenigen Mouse-Clicks im BlueControl-Tool definiert und in den KS vario-Regler geladen werden (siehe Kapitel 3.2).



Prozessdaten und ausgewählte Parameterdaten werden zyklisch geschrieben und gelesen. Vorgabewerte werden vom Gerät nur bei einer Wertänderung übernommen.

Datenformat

Werte wie z.B. Ist- und Sollwerte werden im 16 Bit Festkommaformat (FixPoint) mit einer Nachkommastelle übertragen.

Parameterkanal

Auf alle Prozess-, Parameter- und Konfigurationsdaten kann zusätzlich über den Parameterkanal zugegriffen werden. Diese Daten werden über mehrere Zyklen auf Anforderung übertragen. Hier sind auch Zugriffe im FloatFormat möglich. -Diese Zugriffe werden in dieser Dokumentation unten beschrieben.

3.1. Konfigurierbare Prozessdatenmodule

Der Anwender kann die Übertragung der Prozessdaten aus einer vorgegebenen Auswahl von Modulen zusammenstellen, um seine Anforderungen an Übertragungswerte, Speicherplatz und Übertragungszeit flexibel realisieren zu können. Die Konfiguration erfolgt über das jeweilige Buskonfigurierungstool des Busmasters. Die Auswahl welche Prozessdaten und Parameter übertragen werden sollen erfolgt über das BlueControl-Engineering-Tool.

Modulumfang Angeboten werden folgende Module:

Nr	Modul	Bezeichnung	Anz. Bytes	Typ	Kennung [hex]	Kennung [dez]
1	A.1	General: Parameter –Channel	8/8		0xF3	243
2	A.2	Variables Multiplexermodul	4/4		0x71	113
3	A.3	Multiplexermodul für 1 Date	6/6	Integer / Fix-Point 1	0xF2	242
4	A.4	Multiplexermodul für 4 Daten	12/12	Integer / Fix-Point 1	0xF5	245
5	A.5	Multiplexermodul für 8 Daten	20/20	Integer / Fix-Point 1	0xF9	249
6	A.6	Multiplexermodul für 14 Daten	32/32	Integer / Fix-Point 1	0xFF	255
17	A.7	General: Ausg.Daten aktivieren	1		0x20	32
7	B.1	I/O: 1 word input	2	Integer / Fix-Point 1	0x50	80
8	B.2	I/O: 2 words input	4	Integer / Fix-Point 1	0x51	81
9	B.3	I/O: 4 words input	8	Integer / Fix-Point 1	0x53	83
10	B.4	I/O: 8 words input	16	Integer / Fix-Point 1	0x57	87
11	B.5	I/O: 16 words input	32	Integer / Fix-Point 1	0x5F	95
12	B.6	I/O: 1 word output	2	Integer / Fix-Point 1	0x60	96
13	B.7	I/O: 2 words output	4	Integer / Fix-Point 1	0x61	97
14	B.8	I/O: 4 words output	8	Integer / Fix-Point 1	0x63	99
14	B.9	I/O: 8 words output	16	Integer / Fix-Point 1	0x67	103
16	B.10	I/O: 16 words output	32	Integer / Fix-Point 1	0x6F	111

Mindestens ein Modul muss vorgegeben werden.

Dem Anwender stehen folgende Module zur Verfügung:

- n Platzhalter für 1 Wort, 2, 4, 8, 16 Worte (wortkonsistent) für Eingang und Ausgang (1, 2, 4, 8, 16 Werte) → **Modul B**
- ein variables Multiplexermodul mit variabler Anzahl zu übertragender Daten mit 4 Ein-Ausgangsbytes → **Modul A.2**
- Vier Multiplexermodule mit fester Anzahl zu übertragender Daten → **Modul A.3...A.6**
- ein Parameterkanal-Modul mit 8 Ein-Ausgangsbytes → **Modul A.1**
- Falls gewünscht können die Ausgangsdaten auch definiert in den Regler übertragen werden. Dies geschieht mit einem Flankenwechsel von 0 auf 1 im Byte des **Moduls A.7**. Bleibt das Byte auf 1, so werden die Daten ständig (bei Wertänderung) in den Regler übertragen. Dies geschieht auch wenn das Modul nicht eingesetzt wird. Üblicherweise wird also das Modul A.7 NICHT benötigt.
- max. Eingangslänge Prozeßdaten: 244 Bytes
- max. Ausgangslänge Prozeßdaten: 244 Bytes
- max. Anzahl Module: 50

Wird kein "variables Multiplexermodul" definiert oder werden I/O-Module vor einem "variablen Multiplexermodul" definiert, so entspricht der Aufbau der I/O Module (3 – 12) dem des Daten-Cache des Buskopplers.

Es gelten folgende Einschränkungen für die Definition der Module:

Die Module "Parameterkanal" und "Variables Multiplexermodul" können nur einmal eingefügt werden. Das Modul "Parameterkanal" kann nicht nach dem Modul "Variables Multiplexermodul" eingesetzt werden. Nach dem Modul "Variables Multiplexermodul" muß mindestens ein I/O Module (3 – 12) definiert werden mit dem die eigentlichen Prozessdaten oder Parameter übertragen werden.

3.2. Definition der zu übertragenden Werte im Engineering Tool "BlueControl"

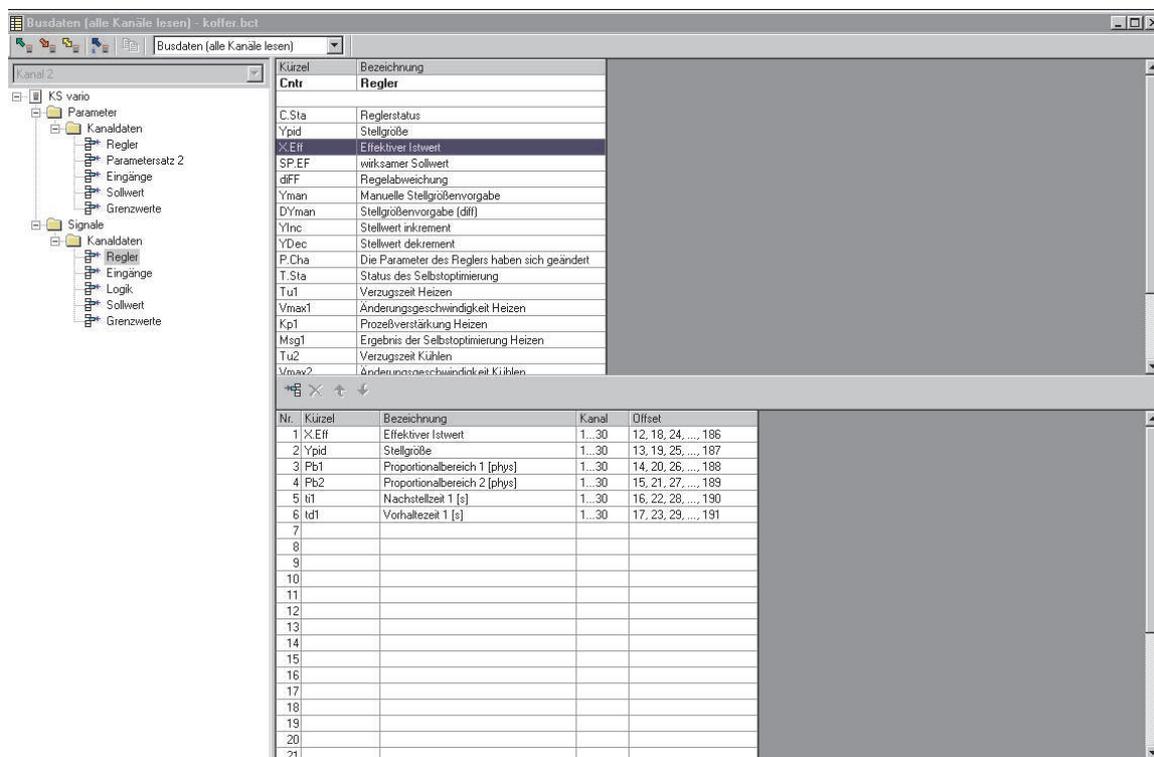
Im BlueControl lassen sich die zu übertragenden Daten auf 2 Arten auswählen:

- Bis zu maximal 120 beliebige Parameter und Prozessdaten von beliebigen Kanälen zum Schreiben sowie max. 120 zum Lesen. Die Daten werden mittels der oben aufgeführten Prozessdatenmodule (I/O-Module Nr. 3 – 12) zum Profibusmaster ausgetauscht. Die Positionierung bestimmt die Reihenfolge in der Übertragung.

The screenshot shows the 'Busdaten (lesen)' window in BlueControl. On the left is a tree view of the project structure under 'Kanal 2'. The main area is divided into two tables. The top table lists parameters with their abbreviations and full names. The bottom table is a configuration table for data transfer.

Nr.	Kürzel	Bezeichnung	Kanal	Offset
1	X.Eff	Effektiver Istwert	1	0
2	Ypid	Stellgröße	1	1
3	Pb1	Proportionalbereich 1 [phys]	1	2
4	Pb2	Proportionalbereich 2 [phys]	1	3
5	ti1	Nachstellzeit 1 [s]	1	4
6	td1	Vorhaltezeit 1 [s]	1	5
7	X.Eff	Effektiver Istwert	2	6
8	Ypid	Stellgröße	2	7
9	Pb1	Proportionalbereich 1 [phys]	2	8
10	Pb2	Proportionalbereich 2 [phys]	2	9
11	ti1	Nachstellzeit 1 [s]	2	10
12	td1	Vorhaltezeit 1 [s]	2	11
13				
14				
15				
16				
17				
18				
19				
20				
21				

- Zusätzlich oder alternativ können - für alle Kanäle gemeinsam - bis zu jeweils 32 beliebige Parameter und Prozessdaten ausgewählt werden. So können mit der Auswahl einer Date z.B. die Istwerte von allen Kanälen (max. 30) übertragen werden. Insgesamt können somit bis zu 960 Schreib- und 960 Lesedaten definiert werden (32 Daten x 30 Kanäle).



Diese ausgewählten Daten (maximal 1080 Schreib- und 1080 Lesedaten) stehen im Buskoppler als Cache-Speicher in der im BlueControl definierten Reihenfolge zur Verfügung.

Die jeweiligen Indizes bzw. Offsets der einzelnen Daten werden über das BlueControl-Tool angezeigt bzw. können ausgedruckt werden.

3.3. Aufbau des Daten-Cache-Speichers im Buskoppler

Index Read-Cache Inhalt

0	beliebige Daten von beliebigen Kanälen
bis max.120	
ab max. 121	Ausgewählte Daten (für alle Kanäle identisch): alle Daten Kanal 1 alle Daten Kanal 2 ... alle Daten Kanal 30
bis max. 1080	

Index Write-Cache Inhalt

0	beliebige Daten von beliebigen Kanälen
bis max.120	
ab max. 121	Ausgewählte Daten (für alle Kanäle identisch): alle Daten Kanal 1 alle Daten Kanal 2 ... alle Daten Kanal 30
bis max. 1080	

Auf die Daten des Cache-Speichers im Buskoppler kann lesend und schreibend auch über Parameterkanal und DPV-1 Nachricht zugegriffen werden. Daten, die permanent oder über das "Multiplexermodul" des Prozessdatenkanal beschrieben werden, können mit dem Parameterkanal und der DPV-1 Nachricht sinnvollerweise nicht verändert werden, da sie im nächsten Datenaustausch auf dem Profibus von diesem wieder überschrieben werden. Probleme dieser Zweipunktbedienung sind vom Anwender zu berücksichtigen.

3.4. Datenaustausch über variables Multiplexermodul

Modul A.2

Im KS Vario können für den Lese- und Schreibzugriff jeweils maximal 1080 Prozessdaten definiert werden. Da diese große Anzahl von Daten nicht in einem Prozessbereich des Profibus zur Verfügung gestellt werden kann, besteht mit dem variablen Multiplexermodul die Möglichkeit in den I/O Modulen (3 – 12) zu multiplexen. Hierfür werden in jede Richtung 4 Byte Index-Information übertragen.

Anwendungsbeispiel:

In der Buskonfigurierung werden hinter dem variablen Multiplexermodul 2 I/O-Module mit je16 Worten (z.B. I/O-Modul 12) eingetragen.

Im BlueControl-Tool werden für jeden Kanal z.B. 32 Prozessdaten/Parameter ausgewählt. Die Daten pro Kanal stehen im Cache-Speicher des Buskopplers jeweils hintereinander. Über das Multiplexmodul können nun pro Zugriff z.B. alle 32 Daten eines Kanals übertragen werden. Anschließend können über einen neuen Eintrag in IndexOUT die Daten des nächsten Kanals übertragen werden.

Der Datenaufbau des variablen Multiplexermodul:

Modul A.2

Datenrichtung	Wort	Wertebereich	Beschreibung
IndexOUT	Index_Read	0 – 1080	Index des Cache-Speicher der 1. Date des Datenblocks, die von der Steuerung angefordert wird
	Index_Write	0 – 1080	Index des Cache-Speicher der 1. Date des Datenblocks, die von der Steuerung vorgegeben wird
IndexIN	Index_Read	0 – 1080	Index des Cache-Speicher der 1. Date des Datenblocks, die vom Buskoppler zur Verfügung gestellt wird
	Index_Write	0 – 1080	Index des Cache-Speicher der 1. Date des Datenblocks, die vom Buskoppler übernommen wurde

IndexOUT

Die Daten enthalten die Information ab welchem Index des Cache-Speichers Daten des Master gesendet bzw. Daten vom Cache-Speicher des Buskopplers angefordert/gelesen werden sollen.

IndexIN

Die Daten enthalten die gespiegelte Information welche Daten des Buskopplers der Master zur Verfügung stellt bzw. welche Daten der Buskoppler übernommen hat.

3.4.1 Datentransfer vom Profibus-Master zum Buskoppler

Schreiben

Sollen Daten vom Master zum Buskoppler übertragen werden, so trägt der Master in den Datenbereich (I/O Module 3 – 12) hinter dem variablen Multiplexermodul die Daten ein und stellt abschliessend das Index-Write-Wort von IndexOUT auf den Index der zugehörigen 1. Date des Daten-Cache (Buskoppler).

Kann von einem Master nicht sichergestellt werden, dass der Eintrag der Daten und die zugehörige Adressierung im Index-Write-Wort konsistent auf dem Profibus ausgegeben wird, ist das Index-Write-Wort zu Beginn der Manipulation auf 0 zu stellen. Hiermit wird signalisiert, dass die Daten vom Buskoppler momentan nicht übernommen werden können. Zum Abschluß wird vom Master der Wert für die 1. Date des transferierten Bereiches eingetragen.

Der Buskoppler übernimmt die Daten fortlaufend in sein Daten-Cache, ab der Position, die im Index-Write-Wort eingetragen ist. Die Übernahme der Daten wird durch den Buskoppler signalisiert, indem er den Index für die 1. Date im Index-Write-Wort des IndexIN Bereiches einträgt. Die Daten werden weiterhin zyklisch vom Buskoppler übernommen, bis sich der Wert in der Date Index-Write-Wort von IndexOUT ändert.

Wird durch das Index-Write-Wort ein Datenbereich definiert, der ausserhalb der maximal 1080 Prozessdaten liegt, so erfolgt keine Bestätigung der Übernahme durch den Buskoppler. Es wird im Index-Write-Wort des IndexIN eine 0 zurückgegeben.

3.4.2 Datentransfer vom Buskoppler zum Profibus-Master

Lesen

Der Master fordert vom Buskoppler Daten an, indem er in das Index-Read-Wort von IndexOUT die 1. Date des Daten-Cache (Buskopplers) einträgt.

Der Buskoppler trägt die gewünschten Daten fortlaufend, beginnend mit der im Index-Read-Wort von IndexOUT definierten 1. Date, in den Datenbereich des variablen Multiplexermoduls ein. Abschliessend wird im Index-Read-Wort von IndexIN der Index der 1. Date des übertragenen Datenbereichs eingetragen.

Kann vom Buskoppler nicht sichergestellt werden, dass der Eintrag der Daten und die zugehörige Bestätigung im Index-Read-Wort konsistent auf dem Profibus ausgegeben wird, ist das Index-Read-Wort zu Beginn der Manipulation auf 0 zu stellen. Hiermit wird signalisiert, dass die Daten vom Master momentan nicht übernommen werden können. Zum Abschluß wird vom Buskoppler der Wert für die 1. Date des transferierten Bereiches eingetragen.

Die Buskoppler aktualisiert die gewünschten Daten in jedem Durchlauf, bis sich der Wert in der Date des Index-Read-Wort von IndexOUT ändert.

Wird durch das Index-Read-Wort ein Datenbereich definiert, der ausserhalb der maximal 1080 Prozessdaten liegt, so erfolgt keine Bestätigung der Übernahme durch den Buskoppler. Es wird im Index-Read-Wort des IndexIN eine 0 zurückgegeben.

3.5. Datenaustausch über Multiplexermodule mit fester Datenlänge

Module
A.3.A.6

Es sind 4 Multiplexmodule mit fester Datenlänge definiert (1 Date, 4 Daten, 8 Daten oder 14 Daten je Richtung). Diese Module enthalten die Multiplexfunktion, sowie eine feste Anzahl von Daten die zu diesem Modul gehören. Die Daten sind als konsistente Daten definiert und müssen auch in dieser Form übertragen werden. Kann ein Master nur im Zusammenhang mit konsistenten Datenmodellen diese garantieren (S7 von Siemens), so sind diese Module zu verwenden.

Es können mehrere Multiplexmodule verwendet werden. Einzeldaten, sowie das Modul für den Parameterkanal sind vor dem ersten Multiplexmodul zu verwenden.

Der Datenaufbau des Multiplexermodule mit fester Datenlänge:

z.B. Modul A.4

Datenrichtung	Wort	Wertebereich	Beschreibung
Index OUT	Index_Read	0 – 1080	Index des Cache-Speicher der 1. Date des Datenblocks, die von der Steuerung angefordert wird
	Index_Write	0 – 1080	Index des Cache-Speicher der 1. Date des Datenblocks, die von der Steuerung vorgegeben wird
	Read Value 1	2 Byte	
	Read Value 2	2 Byte	
	Read Value 3	2 Byte	
	Read Value 4	2 Byte	
Index IN	Index_Read	0 – 1080	Index des Cache-Speicher der 1. Date des Datenblocks, die vom Buskoppler zur Verfügung gestellt wird
	Index_Write	0 – 1080	Index des Cache-Speicher der 1. Date des Datenblocks, die vom Buskoppler übernommen wurde
	Write value 1	2 Byte	
	Write value 2	2 Byte	
	Write value 3	2 Byte	
	Write value 3	2 Byte	

IndexOUT Die Daten enthalten die Information ab welchem Index des Cache-Speichers Daten des Master gesendet bzw. Daten vom Cache-Speicher des Buskopplers angefordert/gelesen werden sollen.

IndexIN Die Daten enthalten die gespiegelte Information welche Daten des Buskopplers der Master zur Verfügung stellt bzw. welche Daten der Buskoppler übernommen hat.



Die Multiplexmodule werden im "KS VARIO Engineering Set" mit einem Funktionsbaustein unterstützt --> FB 108 (siehe Kapitel "Funktionsbausteine für Simatic S7").

4. Parameterübertragung via Parameterkanal in DP-V0

Für die Übertragung von Parametern steht ein 'Parameterkanal' zu Verfügung, über den unabhängig von den ausgewählten Prozessdaten Daten transparent ausgetauscht werden können. Dabei werden alle möglichen Zugriffsarten des Protokolls unterstützt (Einzel- oder Blockzugriff, Datenformate Integer oder Real). Die Kommunikation zum Regler erfolgt transparent, d.h. der Anwender ist für die Überwachung der Wertebereiche, Betriebsarten (remote/local) usw. selbst verantwortlich.

Der Parameterkanal ist für größere Datenmengen mit geringen Anforderungen an die Übertragungsgeschwindigkeit ausgelegt. Diese Nachricht wird im zyklischen Prozessdatenaustausch durchgeführt (auch DP-V0 (=Standard) -Betrieb).

Prinzipskizze Parameterkanal

Parameterkanal Auf Prozess-, Parameter- und Konfigurationsdaten kann über den Parameterkanal zugegriffen werden. Diese Daten werden über mehrere Zyklen auf Anforderung übertragen.

Der Anwender kann die Übertragung des Parameterkanals einrichten, indem er das Modul A.1 in seiner PROFIBUS-Konfiguration auswählt. Die Konfiguration erfolgt über das jeweilige Buskonfigurierungstool des Busmasters.

Modul A.1

General- Parameter channel:¹⁾

Parameterkanal			
lesen	Byte	schreiben	Byte
Antwortdaten	8	Anforderungsdaten	8

Bitte beachten Sie eine notwendige konsistente Datenübertragung !

4.1. Nachrichtenelemente

Im Folgenden werden einige Begriffe erläutert, die im abgewickelten Protokoll des Parameterkanals verwendet werden:

Element	Beschreibung
ID	Kennzeichnung der Telegrammart
ID1	Datenformat der zu übertragenden bzw. empfangenen Daten
Rd.Cnt	Anzahl der zu lesenden Daten
Adr.High	Startadresse High-Byte
Adr.Low	Startadresse Low-Byte
Cnt. Real	Anzahl der zu schreibenden Realdaten
Cnt. Int	Anzahl der zu schreibenden Integerdaten

ID

Das Element ID identifiziert die Telegrammart: ID = 0x10 (16_) \triangleq Starttelegramm
 ID = 0x68 (104_) \triangleq Datentelegramm
 ID = 0x16 (22_) \triangleq Endtelegramm

ID1

Das Element ID1 definiert das Datenformat: ID1 = 0 \triangleq Integer
 ID1 = 1, 3 \triangleq Real
 ID1 = 2 \triangleq einzelne Zeichen (char)
 ID1 = 4 \triangleq Zeichen im Kompaktformat

Format Integer(0)

Daten werden im Datentelegramm als 2 Byte Integer übertragen. Die Startadresse legt fest, ob diese Daten als Integer-, FixPoint1-, FixPoint2- oder FixPoint3- Wert ausgeführt werden. Wird die Startadresse im Bereich der Float-Adressen vorgegeben, so wird sie auf den zugehörigen Integerbereich gewandelt.

Format Real(1,3)

Daten werden im Datentelegramm als 4 Byte Real übertragen. Wird keine Startadresse im Bereich der Real-Adressen vorgegeben, so wird sie auf den zugehörigen Realbereich gewandelt.

Format Zeichen(2)

Daten werden im Datentelegramm als 1 Byte Zeichen übertragen. Wird keine Startadresse im Bereich der Integer-Adressen vorgegeben, so wird sie auf den zugehörigen Integerbereich gewandelt.

Format Kompaktzeichen(4)

Daten werden im Datentelegramm als 4 Byte Zeichen übertragen. Wird keine Startadresse im Bereich der Integer-Adressen vorgegeben, so wird sie auf den zugehörigen Integerbereich gewandelt. Nicht benutzte Stellen im Datentelegramm werden mit dem Wert 0x00 aufgefüllt.

Rd.Cnt

Rd.Cnt definiert die Anzahl der im gewählten Format (ID1) zu lesenden Daten.



Die Daten Cnt.Real und Cnt.Int müssen hierzu null sein.

Adr.High/ Adr.Low

Die Felder Adr.High und Adr.Low definieren die Startadresse der zu übertragenden Daten. Die Adresse wird in ein High-Byte und in ein Low-Byte aufgeteilt.

Das Adressverzeichnis ist in verschiedene Bereiche eingeteilt. Über Bit D15 z.B wird bestimmt, ob es sich um einen Float-Wert oder einen Integer / Fix-Point-Wert handelt.

Die Adressübersicht ist dem Dokument Parametertabelle für KS Vario zu entnehmen (9499-040-72918).

Cnt.Real

Vorgabewert für die Anzahl der zu schreibenden Realwerte (ID1 = 1, 3). Der Wert für Cnt.Int muss 0 sein.

Cnt.Int

Vorgabewert für die Anzahl der zu schreibenden Integerwerte (ID1 = 0, 2, 4). Der Wert für Cnt.Real muss 0 sein.

4.2. Allgemeiner Übertragungsaufbau

Um über ein Datenfenster von 8 Byte die benötigten Parameter übertragen zu können, besteht der Zugriff aus den Teilen Start-, Daten- und Endetelegramm. Im Nachfolgenden werden die vom Master zu versendenden Nachrichten aufgezeigt.

i Bitte beachten Sie die notwendige konsistente Datenübertragung der 8 Bytes für den Parameterkanal.

Start-telegramm

Starttelegramm mit Angabe des Datenformates, der Startadresse und der zu übertragenden Anzahl.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID	ID1	Rd.Cnt	Adr.High	Adr.Low	0	Cnt. Real	Cnt. Int

Daten-telegramm

n Datenblöcke mit den zu übertragenden Nutzdaten

Aufbau des Datentelegramms:

a) Übertragung von REAL-Werten (ID1 = 1,3)

Real-Format

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID	Count			Float			

b) Übertragung von Integerwerten (ID1 = 0)

Integer-Format

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID	Count					Integer	

c) Übertragung von Char-Werten (ID1 = 2)

Zeichen-Format

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID	Count				Char		

d) Übertragung von vier Zeichen (ID1 = 4)

String-Format

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID	Count			Char	Char+1	Char+2	Char+3

Die Bytefolge für Real- und Integerformate erfolgt gemäß der Einstellung in der User-Parametrierung als Motorola- (Default) oder Intel-Format. Siehe hierzu auch das Dokument "SB PROFIBUS-DP Prozessdaten" (9499-040-66618)

Ende-telegramm

ein Endblock, liefert das Ergebnis der Operation

Aufbau des Endetelegramms:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
ID		Result					

Bedeutung von Result	
0	OK
2	Fehlerhafte Adresse
3	Ungültiger Wert
4	Pufferüberlauf

Das Lesen oder Schreiben wird immer vom Master eingeleitet. Wenn die Werte von Cnt.Real oder Cnt.Int $\neq 0$ sind, so wird ein Schreibdienst, ansonsten wird ein Lesedienst ausgelöst; letzteres setzt einen Wert Rd.Cnt $\neq 0$ voraus.

4.3. Ablauf - Daten schreiben

Starttelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sendet:	0x10	ID1	0	Adr.High.	Adr.Low	0	Cnt.Real'	Cnt.Int'
Regler antwortet:	0x10							

Datentelegramme:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x68	Count			Value
Regler antwortet:	0x68	Count			

Bei Count = 1 wird der erste Wert gesendet, zur Flusskontrolle wird Count vom KS VARIO gespiegelt ($1 \leq \text{Count} \leq \text{Cnt.Real}$ bzw. Cnt.Int).

Endetelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x16				
Regler antwortet:	0x16		Result		

4.4. Ablauf - Daten lesen

Starttelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sendet:	0x10	ID1	Rd.Cnt	Adr.High	Adr.Low	0	0	0
Regler antwortet:	0x10						Cnt.Real'	Cnt.Int'

Datentelegramme:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x68	Count			
Regler antwortet:	0x68	Count			Value

Bei Count = 1 wird der erste Wert angefordert, zur Flusskontrolle wird Count vom Regler gespiegelt und der Wert mitgesendet ($1 \leq \text{Count} \leq \text{Cnt.Real}$ bzw. Cnt.Int).

1) Auswahl: nur einer der Werte Cnt.Real und Cnt.Int darf $\neq 0$ sein.

□ Endetelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x16				

Regler	Byte 0	Byte 1	Byte 2-3	Byte 4 - 7
antwortet:	0x16		Result	

4.5. Beispiele

4.5.1 Beispiel 1: Einzelzugriff, Lesen, Integerwert

Der Proportionalbereich 1 des Regelkanals 1 (**Pb 1**) des Reglers soll im FixPoint1 - Format ausgelesen werden
 Adresse = 0x4000 (Offset für FixPoint 1) + 0x400 (Kanal 1) + 0x96 (Pb1) = **0x4496**

Starttelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sendet:	0x10	0x0	0x1	0x44	0x96	0	0	0

Regler	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
antwortet:	0x10						0	1

Datentelegramm 1:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x68	1			

Regler	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
antwortet:	0x68	1			Value

Endetelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x16				

Regler	Byte 0	Byte 1	Byte 2-3	Byte 4 - 7
antwortet:	0x16		0	

4.5.2 Beispiel 2: Blockzugriff, Lesen, Floatwert

Die Parameter der Sollwerteinstellungen für Regelkanal 2 (**SPLo , SPHi, SP2, rSP**) sollen im Real - Format gelesen werden

Adresse = 0x8000 (Offset für Float) + 0xC00 (Kanal 2) + 0x69 x 2 (SPLo) = **0x8CD2**, 4 Werte.

Starttelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sendet:	0x10	0x1	0x4	0x8C	0xD2	0	0	0

Regler	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
antwortet:	0x10						4	0

Datentelegramm 1:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x68	1			

Regler	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
antwortet:	0x68	1			Value

...

Datentelegramm 4:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x68	4			

Regler	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
antwortet:	0x68	4			Value

Endetelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x16				

Regler	Byte 0	Byte 1	Byte 2-3	Byte 4 - 7
antwortet:	0x16		0	

4.5.3 Beispiel 3: Einzelzugriff, Schreiben, Integerwert

Der Vorhaltezeit 1 des Regelkanals 30 (td1) soll im Integer - Format geschrieben werden
 Adresse = 0x0000 (Offset für Integer) + 0x3E00 (Kanal 30) + 0x9A (td1) = **0x3E9A**

Starttelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sendet:	0x10	0x0	0x0	0x3E	0x9A	0	0	1

Regler	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
antwortet:	0x10							

Datentelegramm 1:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x68	1			Value

Regler	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
antwortet:	0x68	1			

Endetelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x16				

Regler	Byte 0	Byte 1	Byte 2-3	Byte 4 - 7
antwortet:	0x16		0	

4.5.4 Beispiel 4: Blockzugriff, Schreiben, Floatwert

Die Parameter der Grenzwertes 1 von Reglerkanal 30 (L, H, HYS) sollen im Real - Format geschrieben werden
 Adresse = 0x8000 (Offset für Float) + 0x3C00 (Kanal 30) + 0x109 (L) = **0xFD09**_{dez}, 3 Werte).

Starttelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master sendet:	0x10	0x1	0x0	0xFD	0x09	0	3	0
Regler antwortet:	0x10							

Datentelegramm 1:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x68	1			Value

Regler antwortet:	0x68	1			
-------------------	------	---	--	--	--

...

Datentelegramm 3:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x68	3			Value

Regler antwortet:	0x68	3			
-------------------	------	---	--	--	--

Endetelegramm:

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4 - 7
Master sendet:	0x16				

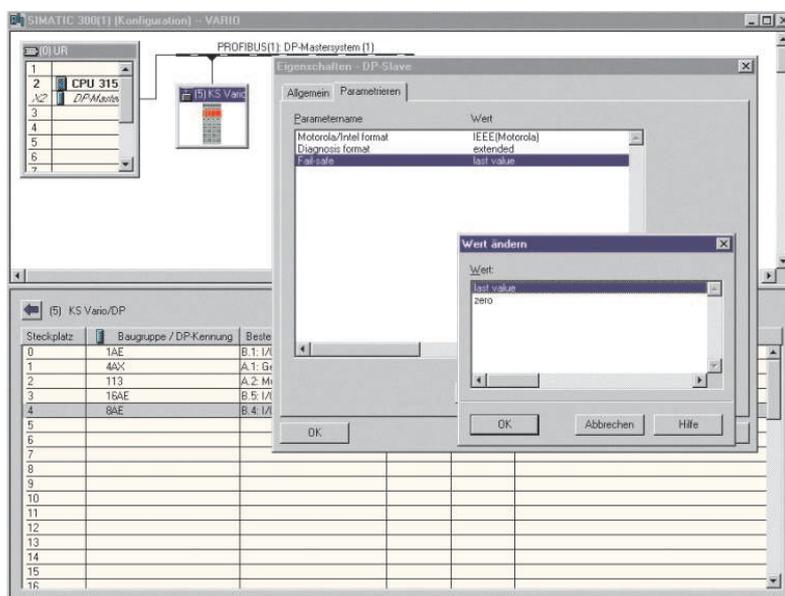
Regler antwortet:	0x16		0		
-------------------	------	--	---	--	--

5. Parameterübertragung via DP-V1

Für die Übertragung von Parametern ist auch die DPV1 Funktionalität realisiert.

6. User-Parametrierung, Diagnose

Der KS VARIO besitzt zusätzlich zu den Standard - Parametrierdaten auch anwenderspezifische Parametrierdaten. Die Einstellung erfolgt über das jeweilige Buskonfigurationstool des Busmasters.



Die User-Parametrierung gilt geräteweit. In den folgenden Tabellen sind die Bedeutungen der User-Parametrierdaten (4 Byte) dargestellt.

	Bit	Bez.	Bedeutung	
1...3. Byte			Für DP-V1 reserviert. Für DP-V0 - Betrieb sind diese Bytes auf null gesetzt.	
	Bit	Bez.	Bedeutung	Default
4. Byte	0	Motorola / Intel format	Format für Floatwerte: Motorola (IEEE 754) / Intel (0 / 1) Zum Anschluss auch an nicht konforme SPSen oder PC-Karten.	0 (Motorola)
	1	Diagnosis format	Diagnose extended / Standard (0 / 1) Extended - Diagnose: Standard - Diagnose plus gerätespezifischer Diagnose. Standard - Diagnose: (6 Bytes) ohne gerätespezifische Informationen.	0 (extended)
	2	Fail-safe	Last value / zero (0 / 1) Verhalten bei Busfehlern: vorhandene Werte halten oder auf null setzen; Einsatz abhängig von Anlagenkonzept .	0 (last value)
3..7		reserviert		0

6.1. PROFIBUS-DP Diagnoseinformationen

PROFIBUS-DP bietet eine komfortable und vielschichtige Möglichkeit, Diagnosemeldungen aufgrund von Fehlerzuständen zu verarbeiten. Die Diagnoseinformationen des KS VARIO bestehen aus Standarddiagnoseinformationen (6 Bytes) und zusätzlichen gerätespezifischen Diagnoseinformationen. Letztere sind per User-Parametrierung abschaltbar.

6.1.1 Standard - Diagnosenachricht

Eine Standard-Diagnosenachricht besteht aus 6 Bytes.

	Bit	Bez.	Bedeutung
1. Byte	0	Diag.station	existiert nicht (setzt Master)
	1	Diag.station_not_ready	Slave ist nicht für den Datenaustausch bereit
	2	Diag.cfg_Fault	Konfigurationsdaten stimmen nicht überein
	3	Diag.ext_diag	Slave hat externe Diagnosedaten
	4	Diag.not_supported	angeforderte Funktion wird im Slave nicht unterstützt
	5	Diag.invalid_slave_response	setzt Slave fest auf 0
	6	Diag.prm_fault	falsche Parametrierung (Identnummer etc.)
	7	Diag.master_lock (setzt Master)	Slave ist von anderem Master parametriert

Standard-Diagnose

	Bit	Bez.	Bedeutung
2. Byte	0	Diag.Prm_req	Slave muss neu parametriert werden Die Applikation hat einen Zustand erkannt, der einen Neuanlauf mit einer entsprechenden Neuparametrierung und Konfigurierung erfordert. Der Master führt auf diese Diagnose hin einen Hochlauf mit vorgegebener Parametrierung und Konfigurierung durch.
	1	Diag.Stat_diag	statische Diagnose (Byte Diag-Bits) Der Slave kann aufgrund eines Zustandes in der Applikation keine gültigen Daten zur Verfügung stellen. Der Master fordert daraufhin nur noch Diagnoseinformationen an, solange, bis der Slave dieses Bit wieder zurücknimmt. Der PROFIBUS-DP-Zustand ist aber Data-Exchange, so dass sofort nach Rücknahme der statischen Diagnose der Datenaustausch wieder fortgeführt werden kann.
	2	fest auf 1	
	3	Diag.WD_on	Ansprechüberwachung aktiv
	4	Diag.freeze_mode	Freeze-Kommando erhalten
	5	Sync_Mode	Sync-Kommando erhalten
	6	reserved	
	7	Diag.deactivated	(setzt der Master)

	Bit	Bez.	Bedeutung
3. Byte	0..6	reserved	
	7	Diag.ext_overflow	Dieses Bit setzt der Slave, wenn mehr Diagnosedaten vorhanden sind, als in den zur Verfügung stehenden Diagnosedatenbereich passen.

	Bit	Bez.	Bedeutung
4. Byte	0..7	Diag.master_add	Masteradresse nach Parametrierung (0xFF ohne Parametrierung)

	Bit	Bez.	Bedeutung
5. Byte	0..7		Identnummer (high-byte); KS VARIO: 0x10

	Bit	Bez.	Bedeutung
6. Byte	0..7		Identnummer (low-byte); 0x30

6.1.2 Gerätespezifische Diagnose

Die nachfolgende gerätespezifische Diagnose ist über die User-Parametrierung abschaltbar (→ 2.5 S.11). Damit kann auf die Standard-Diagnose umgeschaltet werden, z.B. für ältere DP-Master, die nicht alle Funktionen unterstützen oder wenn angezeigte Diagnoseinformationen nicht von Interesse sind.

	Bit	Bez.	Bedeutung
7. Byte	0..5	Revisions-Nummer	
	6, 7		immer '1'
	Bit	Bez.	Bedeutung
8. Byte	0..7	sign length	0x08: Blocklänge 8 Bytes
	Bit	Bez.	Bedeutung
9. Byte	0..7	status type	0x81: Typ Status Message
	Bit	Bez.	Bedeutung
10. Byte	0..7	Slot number	0x00: Slot: Gerät
	Bit	Bez.	Bedeutung
11. Byte	0..7	specifier	0x00: keine Status-Unterscheidung
	Bit	Bez.	Bedeutung
12. Byte	0..6	Firmware-Version	Version der PROFIBUS-Software

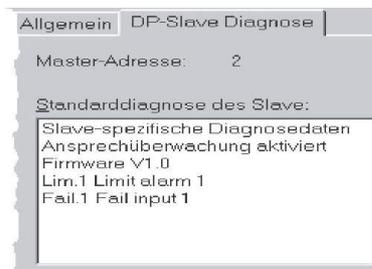
Gerätespez. Diagnose

	Bit	Bez.	Bedeutung	Ursache
13. Byte	0	E.1	Interner Fehler, nicht behebbar	z.B. defektes EEPROM
	1	E.2	Interner Fehler, rücksetzbar	z.B. EMV-Störung
	2	E.3	Konfigurationsfehler, rücksetzbar	Momentan nicht benutzt
	3	E.4	Hardwarefehler	Codenummer und Hardware nicht identisch, falsches I/O-Modul o. Falsche Reihenfolge
	4	E.5	Interner Fehler im DP-Modul	
	5..7		Reserviert	

	Bit	Bez.	Bedeutung	Ursache
14. Byte	0	Lim.	Grenzwertalarm	eingestellter Grenzwert verletzt
	1	HCA	Heizstrom-Alarm	Heizkreisunterbrechung, Heizband zerstört
	2	SSR	Heizstrom-Kurzschluss	Stromfluss im Heizkreis bei Regler aus, SSR defekt, verklebt
	3	Loop	Regelkreis-Alarm	Regelkreis ist unterbrochen (Eingang oder Ausgang)
	4..7		Reserviert	

	Bit	Bez.	Bedeutung	Ursache
15. Byte	0	Fail	Fühlerfehler	Fühler defekt, Verdrahtungsfehler
	1..7		Reserviert	

i Bitte beachten Sie, dass ältere Versionen von Simatic S7- Mastern die Diagnosewerte nicht korrekt anzeigen.



7. Schnelleinstieg, am Beispiel einer SIMATIC® S7

Auf der dem Engineering Set beiliegenden Diskette befindet sich die GSD-Datei und Beispielprojekte für eine SIMATIC® S7. Mit Hilfe der Konfiguration und des Projektes kann auf einfache Weise eine Kommunikation mit KS VARIO Multiregler-System aufgebaut werden.

Testumgebung

- Für den beispielhaften Testaufbau benötigen Sie folgende Komponenten:
- Programmiergerät (PG) oder PC mit PC-Adapter
- Programmierwerkzeug STEP 7³ V5.0
- Automatisierungsgerät (AG)

z.B. CPU S7 315-2 DP, neue Ausgabe

Komponenten

- KS VARIO BK DP/V1 (Bestellnr.: K SVC-101-00111)
- z.B. KS VARIO T8/UTH (Bestellnr.: K SVC-104-00441)
- Engineering Set (Bestellnr. K SVC-109-20001)
- Kabel

PROFIBUS Kabel AG <-> KS VARIO

Stecker mit integrierten Abschlusswiderständen

PG <-> AG

Beispiel einer Testumgebung:

Aufgabe

- Ein KS VARIO mit der Adresse 5 soll an eine CPU315-2 DP über PROFIBUS-DP angeschlossen werden.
- Es sollen der Istwert, der Sollwert, Heizstrom, Handstellgröße sowie einige Statusmeldungen und Controllworte einiger Kanäle angezeigt bzw. vorgegeben werden können. Hierzu werden 17 Prozeßdaten zum Lesen und 17 zum Schreiben benötigt. Die Daten werden mit dem Engineering-Tool BlueControl ausgewählt.
- Dazu können Prozessdatenmodule B.5 (16 words IN) und B.1 (1 word IN) zum Lesen benutzt werden. Zum Schreiben die Module B.10 (16 words OUT) und B.6 (1 word OUT).

Bevor die Testumgebung in Betrieb genommen wird, sollten Sie sicherstellen, dass das Automatisierungsgerät keine andere Anwendersoftware enthält ("urgelöscht").

Vorgehensweise:

- Herstellen der Verbindungen (PROFIBUS)

Beispiel-Engineering mit BlueControl in KS VARIO laden.

Am KS VARIO Buskoppler die Adresse 5 einstellen (über Front oder BlueControl) und am Busnetz anschliessen.

Busabschlusswiderstände aktivieren.

Vorgehen

- PROFIBUS-Netzkonfiguration

Diskette (Engineering Set) in PG stecken.

Beispielprojekt dearchivieren (KSVARIO.zip)

Projekt KS VARIO öffnen

Adressierungen und CPU Hardwarekonfiguration gegebenenfalls anpassen und in den DP-Master (CPU315-2 DP) übertragen.

AG auf Run schalten.

Nach Inbetriebnahme des Testaufbaus kann mit Hilfe der dem Projekt beigefügten Variablen-Tabellen (VAT 3) ein Test des E/A-Bereichs durchgeführt werden.

Slot	Modul	Bestellnummer	E-Adresse	A-Adresse	Kommentar
0	PS 307	3ES7 30-1EX00-0AB0	256	257	
1	CPU 315-2 DP	6ES7 315-2CG03-0AB0	258	318	
2	PS 307	3ES7 30-1EX00-0AB0	260	261	
3	PS 307	3ES7 30-1EX00-0AB0	262	263	
4	PS 307	3ES7 30-1EX00-0AB0	264	265	
5	KS VARIO DP	3ES7 30-1EX00-0AB0	266	319	
6	PS 307	3ES7 30-1EX00-0AB0	268	269	
7	PS 307	3ES7 30-1EX00-0AB0	270	271	
8	PS 307	3ES7 30-1EX00-0AB0	272	273	
9	PS 307	3ES7 30-1EX00-0AB0	274	275	
10	PS 307	3ES7 30-1EX00-0AB0	276	277	
11	PS 307	3ES7 30-1EX00-0AB0	278	279	
12	PS 307	3ES7 30-1EX00-0AB0	280	281	
13	PS 307	3ES7 30-1EX00-0AB0	282	283	
14	PS 307	3ES7 30-1EX00-0AB0	284	285	
15	PS 307	3ES7 30-1EX00-0AB0	286	287	
16	PS 307	3ES7 30-1EX00-0AB0	288	289	

7.1. Benutzung des Parameterkanals

7.1.1 Einfacher Zugriff

Über die Variablen-tabelle VAT2 kann über eine vereinfachte Eingabemöglichkeit auf den Parameterkanal zugegriffen werden. Im Beispielprogramm werden im FC104 die entsprechenden Umsetzungen ausgeführt, so dass man für eine manuelle Datenübertragung nur folgende Werte vorgeben muss:

- Auswahl Lesen / Schreiben
- Auswahl Datentyp

VAT 2: *Beispiel für eine einfache Parameterbedienung, hier Lesen von 4 Realwerten*

- Vorgabe der Startadresse
- Anzahl der zu übertragenden Daten
- Ausführungsstart

Vorgehen

Zum Austesten einer Datenübertragung gehen Sie folgendermaßen vor:

- Wählen Sie aus, ob Sie Daten lesen oder schreiben möchten (0 bzw. 1)
- Wählen Sie den Übertragungsdatentyp aus (Integer, Real, Zeichen; 0,1,2)
- Geben Sie eine Startadresse vor; wenn Sie eine Real-Übertragung eingestellt haben, sollten Sie die Adresse im Hex-Format vorgeben.
- Die Anzahl der zu übertragenden Daten wählen Sie mit Count vor.
- Schreibwerte geben Sie im Bereich DB37.Dxxx vor; abhängig vom gewählten Datentyp als Wort oder Doppelwort.
- Mit dem Wert Start = 1 starten Sie die Abarbeitung der Nachricht.
- Die Nachricht ist beendet, wenn im Result 0x0002 (positiv) oder 0x0004 (negativ) vorliegt. Gelesene Werte stehen im Bereich DB37.Dxxx.

Ergebnis

Im Anzeigewort Result ist der aktuelle Zustand der Übertragung für den gewählten Datenbereich ersichtlich. Das Anzeigewort besitzt folgenden Aufbau:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Timeout (FB)	Service falsch				000 : ok 010 : fehlerhafte Adresse 011 : ungültiger Wert 100: Pufferüberlauf				warten auf Endtelegramm	Service (0=Read; 1=Write)	Reset Auftrag	wartet auf Quittung	Auftrag fertig mit Fehler	Auftrag fertig ohne Fehler	Auftrag läuft

7.2. Direkter Zugriff

Alternativ kann über die Variablen-tabelle VAT3 direkt auf die ausführenden Funktionsbausteine FB106, FB107 (s.unten) zugegriffen werden.

Die Beschreibung der Parameter siehe unten

Beispiel für den direkten Zugriff auf die FBs zum Parameterkanal

7.3. Benutzung der Multiplexerfunktion

Über die Variablen-tabelle VAT5 kann über Multiplexfunktionalität auf die bis zu 1080 definierten Daten zugegriffen werden. Nach Eingabe der entsprechenden Parameter (wie Anfangs- und Endadresse der zu lesenden/schreibenden Daten) werden die Daten im Datenbaustein 37 abgelegt/bzw. werden von dort zum KS Vario Buskoppler übertragen. Mit VAT6 werden die von KS Vario gelesenen Daten dargestellt, über VAT7 können Daten in KS Vario geschrieben werden.

Operand	Symbol	Symbolkommentar	Statusformat	Statuswert	Steuerwert	
// Example multiplex channel with FB106						
M	10	"Mux No"	Start Multiplexer FB 106	DEZ	1	1
MW	140	"MuxInRd"	Start index of the data which should be read	DEZ	1	1
MW	142	"MuxMaxRd"	End index of the data which should be read	DEZ	60	60
MW	144	"MuxNoRd"	Number of read data in the area of the multiplexer	DEZ	14	14
MW	150	"MuxInWr"	Start index of the data which should be written	DEZ	18	18
MW	152	"MuxMaxWr"	End index of the data which should be written	DEZ	59	59
MW	154	"MuxNoWr"	Number of write data in the area of the multiplexer	DEZ	14	14
// Values read from device						
DB37.DBW	130	"DataArea".D0065	---	DEZ	0	
DB37.DBW	132	"DataArea".D0066	---	DEZ	0	
DB37.DBW	134	"DataArea".D0067	---	DEZ	0	
DB37.DBW	136	"DataArea".D0068	---	DEZ	0	
DB37.DBW	138	"DataArea".D0069	---	DEZ	0	
DB37.DBW	140	"DataArea".D0070	---	DEZ	0	
// Values to be written to device						
DB37.DBW	260	"DataArea".D00130	---	DEZ	0	//130
DB37.DBW	262	"DataArea".D00131	---	DEZ	0	//131
DB37.DBW	264	"DataArea".D00132	---	DEZ	0	//132
DB37.DBW	266	"DataArea".D00133	---	DEZ	0	//133
DB37.DBW	268	"DataArea".D00134	---	DEZ	0	//134
MW	30	"MuxOutRd"	Read index transferred multiplexer	DEZ	15	
MW	32	"MuxOutWr"	Write index transferred multiplexer	DEZ	18	
MW	34	---	---	DEZ	0	
MW	66	"MuxInRd"	Read index received multiplexer	DEZ	1	
MW	68	"MuxInWr"	Write index received multiplexer	DEZ	46	
MW	70	---	---	DEZ	311	

8. Funktionsbausteine für SIMATIC® S7

Die S7 - Funktionsbausteine FB106 (PMA-FIX) und FB107 (PMA-Float) dienen dem einfachen Zugriff auf Parameter und Konfigurationsdaten des KS VARIO.

Der Aufruf eines FB's erfolgt bedingt bei Auslösung eines Auftrages und solange, wie der Auftrag aktiv ist.

Je nach S7-CPU und eingesetztem DP-Master ergeben sich Unterschiede im E/A-Handling. Bei einer CPU315-2 DP mit benutzer on-board DP-Schnittstelle sind die SFC-Bausteine 14 und 15 zu benutzen, um Daten konsistent zu übertragen. Die SFC-Bausteine 14 und 15 kopieren die E/A-Bereiche in den Merker- oder Datenbausteinbereich. Bei Benutzung eines externen CP's (CP 342-5 DP), sind die entsprechenden SEND und RECEIVE FB's am Anfang und Ende des Zyklusses aufzurufen.

Jeder FB besitzt eine Instanz-DB, der beim FB-Aufruf mit anzugeben ist.

8.1. Aufbau FB106, FB107

Die Funktionsbausteine FB106, FB107 unterstützen den Parameterkanal und besitzen die nachfolgend aufgeführten Aufrufparameter. Im FB106 werden die Nutzdaten im zugeordneten Datenbaustein in Wordbreite abgelegt, beim FB107 in Doppelwortbreite.

Name	Typ	Beschreibung / Funktion																																																																											
A-Anfang	Pointer	Anfang Adressbereich der Ausgangsworte (z. B. Adresse Datenbereich 'RECORD' des SFC 15, Ax, y bei Nutzung eines externen CP's). Bei Angabe eines Datenwortes muß die DB-Nr. mit übergeben werden (z. B. DB4.DBX0.0)																																																																											
E-Anfang	Pointer	Anfang Adressbereich der Eingangsworte (z. B. Adresse Datenbereich 'RECORD' des SFC 15, Ex, y bei Nutzung eines externen CP's). Bei Angabe eines Datenwortes muß die DB-Nr. mit übergeben werden (z. B. DB4.DBX0.0)																																																																											
DB-Para	Pointer	Angabe des Datenbausteins mit den Parametrierdaten. Die Eingabe umfaßt die Datenbaustein-Nr. und die Datenwort-Nr. wo die Parameterdaten beginnen. Es ist dabei <u>kein</u> Offset zu berücksichtigen. Die Daten werden von der angegebenen Adresse als Parameterdaten (Nutzdaten) interpretiert. Die Angabe des DB muß in folgender Form erfolgen z. B. DB6.DBX10.0																																																																											
Service	WORD	Service (Read/Write)																																																																											
Code_nr	WORD	Lesezugriff: Anzahl der zu lesenden Werte																																																																											
FB_nr	WORD	Adresse, High-Byte																																																																											
FKT_nr	WORD	Adresse, Low-Byte																																																																											
Typ	WORD	d.c. (immer '0')																																																																											
Timeout	DWORD	Timeout-Wert, wird bei jedem Aufruf dekrementiert. Ist der Wert = 1, wird der Auftrag mit der Fehlermeldung 'timeout' abgebrochen.																																																																											
DWLR	WORD	Länge der Real-Werte; Vorgabe beim Schreiben																																																																											
DWLI	WORD	Länge der Integer-Werte; Vorgabe beim Schreiben																																																																											
DWLC	WORD	Länge der Char-Werte; Vorgabe beim Schreiben																																																																											
ANZW	W	Im Anzeigewort ist der aktuelle Zustand der Übertragung für den gewählten Datenbereich ersichtlich. Das Anzeigewort besitzt folgenden Aufbau:																																																																											
		<table border="1"> <thead> <tr> <th>15</th> <th>14</th> <th>13</th> <th>12</th> <th>11</th> <th>10</th> <th>9</th> <th>8</th> <th>7</th> <th>6</th> <th>5</th> <th>4</th> <th>3</th> <th>2</th> <th>1</th> <th>0</th> </tr> </thead> <tbody> <tr> <td rowspan="2">Timeout (FB)</td> <td rowspan="2">Service falsch</td> <td colspan="4"></td> <td>000 : ok</td> <td colspan="2"></td> <td rowspan="2">warten auf Endtelegramm</td> <td rowspan="2"></td> <td rowspan="2">Service (0=Read; 1=Write)</td> <td rowspan="2">Reset Auftrag</td> <td rowspan="2">wartet auf Quittung</td> <td rowspan="2">Auftrag fertig mit Fehler</td> <td rowspan="2">Auftrag fertig ohne Fehler</td> <td rowspan="2">Auftrag läuft</td> </tr> <tr> <td colspan="4"></td> <td>010 : fehlerhafte Adresse</td> <td colspan="2"></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td colspan="4"></td> <td colspan="4"></td> <td>011 : ungültiger Wert</td> <td colspan="2"></td> <td colspan="2"></td> <td colspan="2"></td> <td></td> </tr> <tr> <td colspan="4"></td> <td colspan="4"></td> <td>100: Pufferüberlauf</td> <td colspan="2"></td> <td colspan="2"></td> <td colspan="2"></td> <td></td> </tr> </tbody> </table>	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Timeout (FB)	Service falsch					000 : ok			warten auf Endtelegramm		Service (0=Read; 1=Write)	Reset Auftrag	wartet auf Quittung	Auftrag fertig mit Fehler	Auftrag fertig ohne Fehler	Auftrag läuft					010 : fehlerhafte Adresse																011 : ungültiger Wert																100: Pufferüberlauf					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																														
Timeout (FB)	Service falsch					000 : ok			warten auf Endtelegramm		Service (0=Read; 1=Write)	Reset Auftrag	wartet auf Quittung	Auftrag fertig mit Fehler	Auftrag fertig ohne Fehler	Auftrag läuft																																																													
						010 : fehlerhafte Adresse																																																																							
								011 : ungültiger Wert																																																																					
								100: Pufferüberlauf																																																																					

Der Funktionsbaustein liest bzw. schreibt Parameter-/Konfigurationsdaten des KS VARIO

A-Anfang, E-Anfang

In diese Parameter werden die Eingangsadressen bzw. Ausgangsadressen des Parameterkanals eingegeben. Die Adressen werden bei der Konfiguration des PROFIBUS Teilnehmers festgelegt.(STEP 7 - Hardware Konfiguration)

- DB-Para
DB-Para ist ein Zeiger auf den Datenbaustein, in den gelesene Daten geschrieben werden bzw. aus dem beim Schreiben Daten entnommen werden.
- Service
Dieser Parameter bestimmt die Zugriffsart (Schreiben / Lesen).

Schreibzugriff:	F0	≙	Integer	Lesezugriff:	0	≙	Integer
	F1	≙	Real		1	≙	Real
	F2	≙	Char		2	≙	Char
- Code_nr
Bei Lesezugriffen definiert Code_nr die Anzahl der zu lesenden Daten. In diesem Fall müssen die Daten DWLR, DWLI, DWLC null sein. Beim Schreiben ist Code_nr = 0 zu setzen.
- FB_nr
Hier wird das High-Byte des zu adressierenden Parameters vorgeben.
Beispiel:

Parameter L. I , Adresse 2100 _{dez}	≙	0834 _{hex}	
FB_nr enthält:	08 _{hex}	≙	8 _{dez}
FKT_nr enthält:	34 _{hex}	≙	52 _{dez}
- FKT_nr
Hier wird das Low-Byte des zu adressierenden Parameters vorgeben.
- DWLR (Real), DWLI (Integer), DWLC (Char)
Diese Parameter enthalten nach einem Lesezugriff die jeweilige Anzahl der empfangenen Daten. Bei einem Schreibzugriff ist die jeweilige Anzahl der zu übertragenden Daten eingetragen. Nur eine der Daten darf einen Wert ≠ 0 enthalten.
- ANZW
Dieses Anzeigewort bildet den aktuellen Zustand der Übertragung ab. Das Bit 4 kann als Eingang zum Rücksetzen des FB 106 / FB 107 verwendet werden.

8.2. Aufbau FB108

Der Funktionsbaustein FB 108 unterstützt den Datenaustausch über Multiplexermodule. Er schreibt und liest zyklisch Daten. Im Buskoppler stehen maximal 1080 Schreib- und Lesedaten zur Verfügung. Da die gesamten Multiplexdaten konsistent sein müssen ist es notwendig die Funktionen SFC 14 bzw. SFC 15 für den Transfer der Daten von und zum Prozessbereich zu verwenden.

Name	Typ	Beschreibung / Funktion
Run	Bool	Aktivieren des Funktionsblock 108
MinRd	Integer	Minimum des Index für Lesevorgang
MaxRd	Integer	Maximum des Index für Lesevorgang
NoMuxRd	Integer	Anzahl der Lesedaten in einer Multiplexnachricht
PtrMuxRd	POINTER	Anfangs-Adresse der durch SFC 14 kopierten Daten aus dem PEW-Bereich
MinWr	Integer	Minimum des Index für Schreibvorgang
MaxWr	Integer	Maximum des Index für Schreibvorgang
NoMuxWr	Integer	Anzahl der Schreibdaten in einer Multiplexnachricht
PtrMuxWr	POINTER	Anfangs-Adresse der Daten die durch SFC 15 in den PAW-Bereich kopiert werden
PtrDataRd	POINTER	Anfangs-Adresse des Datenbereichs in dem die Daten die durch den Funktionsblock 108 gelesen wurden abzulegen sind
PtrDataWr	POINTER	Anfangs-Adresse des Datenbereichs das die Daten enthält die durch den Funktionsblock 108 auszugeben sind
IndOutRd	Integer	Erster Index der Daten die aktuell angefordert wird
IndOutWr	Integer	Erster Index der Daten die aktuell geschrieben wird

Die Daten des Funktionsblock 108 haben folgende Bedeutung:

- Run
Durch eine 1 wird der Funktionsblock und damit der Datenaustausch aktiviert. Mit einer 0 wird der Datenaustausch gestoppt.
- MinRd
Unterer Index der zu lesenden Daten. Der Wertebereich der Date liegt zwischen 1 – 1080
- MaxRd
Oberer Index der zu lesenden Daten. Der Wertebereich der Date liegt zwischen 1 – 1080. Die Anzahl der zyklisch gelesenen Daten ergibt sich damit zu $\text{MaxRd} - \text{MinRd} + 1$
- NoMuxRd
Anzahl der Lesedaten in einer Multiplexnachricht. Entsprechend dem gewählten festen Multiplexmodul kann dieser Wert 1, 4, 8 oder 14 betragen.
- PtrMuxRd
Da es notwendig ist, dass die gesamten Daten des Multiplexmoduls konsistent sind, aus diesem Grund muß der Funktionsblock SFC 14 für das Kopieren der Daten aus dem PEW-Bereich verwendet werden. Dem Funktionsblock ist die Anfangsadresse zu übergeben wohin diese Daten kopiert wurden.
- MinWr
Unterer Index der zu schreibenden Daten. Der Wertebereich der Date liegt zwischen 1 – 1080
- MaxWr
Oberer Index der zu schreibenden Daten. Der Wertebereich der Date liegt zwischen 1 – 1080. Die Anzahl der zyklisch geschriebenen Daten ergibt sich damit zu $\text{MaxWr} - \text{MinWr} + 1$
- NoMuxWr
Anzahl der Schreibdaten in einer Multiplexnachricht. Entsprechend dem gewählten festen Multiplexmodul kann dieser Wert 1, 4, 8 oder 14 betragen.

- **PtrMuxWr**
Da es notwendig ist, dass die gesamten Daten des Multiplexmoduls konsistent sind, aus diesem Grund muß der Funktionsblock SFC 15 für das Kopieren der Daten in den PAW-Bereich verwendet werden. Dem Funktionsblock ist die Anfangsadresse zu übergeben wohin die Daten abzulegen sind damit die Funktion SFC 15 in den PAW-Bereich kopiert kann.
- **PtrDataRd**
Anfangs-Adresse des Datenbereich in dem die Daten die durch den Funktionsblock 108 gelesen wurden abzulegen sind. Typischerweise ist dies ein Datenbaustein. Die Länge des Datenbausteins muß so bemessen sein, dass er die durch $\text{MaxRd} - \text{MinRd} + 1$ definierte Anzahl der zu lesenden Daten aufnehmen kann.
- **PtrDataWr**
Anfangs-Adresse des Datenbereichs das die Daten enthält die durch den Funktionsblock 108 auszugeben sind. Typischerweise ist dies ein Datenbaustein. Die Länge des Datenbausteins muß so bemessen sein, dass er die durch $\text{MaxWr} - \text{MinWr} + 1$ definierte Anzahl der zu schreibenden Daten enthält.
- **IndOutRd**
Zeigt den erster Index der Daten die aktuell angefordert wird an. Der Wert der Date liegt im Bereich von $\text{MinRd} - \text{MaxRd}$.
- **IndOutWr**
Zeigt den erster Index der Daten die aktuell geschrieben wird an. Der Wert der Date liegt im Bereich von $\text{MinWr} - \text{MaxWr}$.

9. Anhang

9.1. Anlagenaufbau

9.1.1 Minimalausbau einer PROFIBUS-Anlage

Eine PROFIBUS-Anlage besteht mindestens aus folgenden Komponenten:
einem Busmaster, der den Datenverkehr steuert,
einem oder mehreren Slaveteilnehmern, die auf Anforderung vom Master Daten zur Verfügung stellen,
dem Übertragungsmedium, bestehend aus Buskabel und Busstecker zum Verbinden der einzelnen Teilnehmer, einem oder mehreren Bussegmenten, die mit Repeatern verbunden sind.

9.1.2 Maximalausbau einer PROFIBUS-Anlage

Ein Bussegment besteht aus maximal 32 Feldgeräten (aktive und passive). Die größtmögliche Anzahl von Slaveteilnehmern, die an einen PROFIBUS-Master über mehrere Segmente hinweg betrieben werden können, wird durch die interne Speicherstruktur des eingesetzten Masters bestimmt. Deshalb sollten Sie sich beim Planen einer Anlage über die Leistungsfähigkeit des Masters informieren. An jeder Stelle kann das Buskabel aufgetrennt werden und durch Hinzufügen eines Bussteckers ein neuer Teilnehmer aufgenommen werden. Am Ende eines Segments kann die Busleitung bis zu den vorgegebenen Segmentlängen erweitert werden und für Erweiterungen ebenfalls neue Teilnehmer angeschlossen werden. Die Länge eines Bussegments ist abhängig von der eingestellten Übertragungsgeschwindigkeit. Die Übertragungsraten werden im Wesentlichen durch die Anlagenkonstellation (Länge eines Segments, verteilte Ein-/Ausgänge) und die geforderten Abfragezyklen einzelner Teilnehmer bestimmt. Für alle Teilnehmer am Bus gilt die vom Master vorgegebene Übertragungsgeschwindigkeit. Am Anfang und am Ende eines Segments müssen Abschlusswiderstände zugeschaltet sein, um einen physikalisch sauberen Signalpegel zu garantieren. Diese sind in den meisten verfügbaren Steckern bereits integriert und müssen nur per Schalter eingelegt werden.

PROFIBUS-Geräte sind in Linienstruktur anzuschließen.

Eine PROFIBUS-Anlage kann durch den Anschluss von Repeatern erweitert werden, wenn mehr als 32 Teilnehmer anzuschließen sind oder größere Entfernungen als die gemäß Übertragungsgeschwindigkeit definierten überbrückt werden müssen.

Im Vollausbau eines PROFIBUS-Systems können maximal 126 Stationen mit den Adressen 0 ... 125 beteiligt sein. Jeder eingesetzte Repeater reduziert die maximale Anzahl von Stationen innerhalb eines Segments. Er hat als passiver Teilnehmer keine PROFIBUS-Teilnehmeradresse. Seine Eingangsbeschaltung belastet das Segment aber zusätzlich durch die vorhandene Stromaufnahme der Bustreiber. Ein Repeater hat jedoch keinen Einfluss auf die Gesamtzahl der angeschlossenen Stationen am Bus. Die maximal anschließbare Anzahl von Repeatern, die in Reihe geschaltet sein dürfen, kann herstellerspezifisch differieren. Beim Projektieren einer Anlage sollten Sie sich deshalb vorher beim Hersteller über mögliche Begrenzungen informieren.

9.1.3 Leitungsverlegung innerhalb von Gebäuden

Die folgenden Verlegungshinweise gelten für ein zweiadriges paarweise verdrehtes Kabel mit Leitungsschirm. Der Leitungsschirm dient der Verbesserung der elektromagnetischen Verträglichkeit. Beim PROFIBUS-Kabel gemäß Leitungstyp A ist ein Geflechtsschirm und ein Folienschirm in das Kabel eingearbeitet. Der Leitungsschirm in den folgenden Ausführungen beinhaltet immer beide Schirmvarianten (Geflechtsschirm und Folienschirm). Es muss unbedingt davon abgesehen werden, den Folienschirm allein zu verwenden, da er sehr dünn ist und leicht unterbrochen werden kann, was zum Unterbrechen des Potenzialausgleichssystems führen kann.

Der Leitungsschirm muss beidseitig und großflächig über leitendes Material mit der Bezugserde kontaktiert sein. Beim Schrankeinbau eines Repeaters oder Feldgerätes sollte ebenfalls der Leitungsschirm möglichst nahe nach der Kabeldurchführung mit einer Schirmschiene über Kabelschellen etc. verbunden werden.

Der Schirm muss bis zum Feldgerät weitergeführt und dort mit metallischen Steckern verbunden werden. Dabei ist sicherzustellen, dass der Schaltschrank, in dem das Feldgerät montiert ist, durch großflächige metallische Kontaktierung das gleiche Erdpotenzial aufweist. Die Montage einer Schirmschiene auf eine Lackoberfläche ist wirkungslos. Durch Einhaltung dieser Maßnahmen werden hochfrequente Störungen über den Geflechtsschirm abgeleitet. Sollten trotzdem von außen verursachte Störspannungen auf die Datenleitungen gelangen, wird das

Spannungspotenzial auf beiden Datenleitungen gleichmäßig angehoben, so dass die Differenzspannung im Normalfall nicht zerstörerisch beeinflusst wird.

Im Regelfall kann eine Verschiebung des Erdpotenzials um ein paar Volt noch eine sichere Datenübertragung gewährleisten. Ist mit einer höheren Verschleppung zu rechnen (Potenzial DGND am Pin 5 gegen Bezugserde), dann sollte eine Potenzialausgleichsleitung parallel zur Busleitung mit einem Mindestquerschnitt von 10 mm² verlegt werden. Bei extremer Störbeeinflussung kann zusätzlich das Buskabel in einem Stahlrohr oder einem dichten Blechkanal verlegt werden. Das Rohr oder der Kanal ist dann regelmäßig zu erden.

Die Busleitung ist stets mit einem Mindestabstand von 20 cm getrennt von anderen Leitungen zu installieren, die eine Spannung größer 60 V übertragen. Ebenfalls ist das Buskabel getrennt von Telefonleitungen und Kabeln, die in explosionsgefährdete Bereiche führen, zu verlegen. In solchen Fällen wird empfohlen, für das Buskabel in einem getrennten Leitungsschacht zu verwenden.

Bei einem Leitungsschacht sollten generell nur leitfähige Materialien verwendet werden, die regelmäßig mit der Bezugserde verbunden sind. Die Buskabel sind keiner mechanischen Beanspruchung oder offensichtlichen Beschädigung auszusetzen. Ist das nicht zu umgehen, sind ebenfalls besondere Schutzmaßnahmen wie z.B. Verlegung in Rohren etc. zu treffen.

9.1.4 Erdfreier Aufbau

Muss aus bestimmten Gründen der Aufbau erdfrei sein, dann darf eine Verbindung zu der Bezugserde nur sehr hochohmig (mit einer RC-Kombination) sein. Beim Anschluss von Repeatern zum Verbinden von Bussegmenten sollte generell der erdfreie Aufbau bevorzugt verwendet werden, um eventuelle Potenzialunterschiede nicht von einem Bussegment in ein anderes zu übertragen.

9.2. Adressübersicht

Modbus Adressen

Die Adresse wird in 2 Byte kodiert. Die höchstwertigsten 2 Bits (D15, D14) werden dazu herangezogen, das Format in dem die Daten geschrieben oder gelesen werden zu definieren.

Das Modbusverzeichnis ist in gleich große Bereiche von jeweils 512 Worten aufgeteilt (Bit D13...D09). Über jeden dieser Bereiche lässt sich auf alle Daten für jeweils einen Regelkanal (1...30 Kanäle) zugreifen.

2 Bereiche haben einen Sonderstatus. Im untersten Adressbereich (Modbusabdr. 0..512) sind alle Gerätedaten hinterlegt. Im darauffolgenden Bereich (Adr. 512...1023) sind die wichtigsten Prozessdaten aller 30 Kanäle zusätzlich noch einmal angeordnet. Dieser Bereich ist für Zugriffe von Visualisierungseinrichtungen gedacht.



**Die detaillierte Adressübersicht finden Sie im Dokument:
Parametertabelle für KS VARIO (9499-040-72918)**

Die Bedeutung der einzelnen Adress-Bits ist wie folgt:

INTEGER/ FIX-Point-Modbusadressen:

MSB		LSB
D15 - D14	D13 - D09	D08 - D00
Datenformat	Gerät, Visualisier., Kanal X	jeweilige Date
00: Integer	00000: Gerätedaten	
01: Fix Point 1	00001: Visualisierungsdaten	
1X: reserviert für Float	00010: Daten Kanal 1	
	00011: Daten Kanal 2	
	
	11111: Daten Kanal 30	

Modbus-Verzeichnis (Datenformat: Integer):

Für den **Fix Point 1**-Bereich sind für die Adressen 4000 hex zu addieren.

Integer

Adressen	Daten
0	Gerätedaten
511 (1FF hex) 512 (200 hex)	Visualisierungsbereich Kanal 1..30
1023 (3FF hex) 1024 (400 hex)	Daten Kanal 1
1535 (5FF hex) 1536 (600 hex)	Daten Kanal 2
2047 (7FF hex)	
....
15872 (3E00 hex) 16383 (3FFF hex)	Daten Kanal 30

FLOAT-Modbusadressen:

MSB		LSB
D15	D14 - D10	D09 - D00
Datenformat	Gerät, Visualisier., Kanal X	jeweilige Date
0: reserviert für Integer und Fix Point 1 1: Float	00000: Gerätedaten 00001: Visualisierungsdaten 00010: Daten Kanal 1 00011: Daten Kanal 2 11111: Daten Kanal 30	

Modbus-Verzeichnis (Datenformat: FLOAT):*Float*

Adressen	Daten
32768 (8000 hex)	Gerätedaten
33791 (83FF hex)	
33792 (8400 hex)	Visualisierungsbereich Kanal 1..30
34815 (87FF hex)	
34816 (8800 hex)	Daten Kanal 1
35839 (8BFF hex)	
35840 (8C00 hex)	Daten Kanal 2
36863 (8FFF hex)	
....
64512 (FC00 hex)	Daten Kanal 30
65535 (FFFF hex)	

Die Daten belegen jeweils 4 Byte. Die Float- Modbusadressen sind also (gegenüber den Integer-Adressen) mit 2 zu multiplizieren.

9.3. Begriffe

Abk.	Abkürzung
AG	Abk. f. Automatisierungsgerät, z.B. SPS
BlueControl®	Engineering Tool - Software für BluePort®-Regler
BluePort®-Schnittstelle	frontseitige Schnittstelle am Regler zum Anschluss eines Engineering Tools
ET	Abk. f. Engineering Tool
Fail-safe	Verhalten des Gerätes, falls die PROFIBUS-Leitung oder der Busmaster ausfällt
FB	Abk. f. Funktionsblock
Fkt	Abk. für Funktion
Forcing	Vorgabe von Ein- oder Ausgängen über Bus
Funktion	eine aus Sicht der Schnittstelle geschlossene Teilfunktion eines Funktionsblocks
Funktionsblock	geschlossene Abarbeitungseinheit
GSD-Datei	Geräte Stammdaten Datei; genormte Beschreibung der Kommunikationsmöglichkeiten
HW	Abk. f. Hardware
Parameterkanal	Möglichkeit zum azyklischen, sequentiellen Übertragen von Werten im zyklischen Prozessdatenaustausch
PG	Abk. f. Programmiergerät
PROFIBUS-DP	Genormtes Kommunikationsprotokoll nach EN50170 Vol.2 (DP: Dezentrale Peripherie)
RS485	Genormte 2 Drahtverbindung, Half duplex, (EIA RS 485)
S5 / S7	Steuerungsfamilien der Siemens AG
Serielle Schnittstelle	Rückseitige Busfähige Schnittstelle des Reglers
SW	Abk. f. Software
TTL	Signalpegel auf Baustein-Ebene
VAT	Variablentabelle: Monitoransicht von Werten in STEP®7

9.4. GSD Datei

```
=====
; Device Database File for product KS Vario / D P
; Copyright (C) PMA Prozeß- und Maschinen Automation GmbH 2002
; Miramstr. 87, D-34123 Kassel, Tel. +49 (0) 561/ 505 -1307
; Version : V1.0 Date: 2002-08-01
; File: PMA_Vari.gsd
=====
```

```
#Profibus_DP
GSD_Revision=1
Vendor_Name = "PMA GmbH"
Model_Name = "KS Vario/DP"
Revision = "V 1.0"
Revision_Number = 1
Ident_Number = 0x1030
Protocol_Ident = 0
; DP
Station_Type = 0
; Slave
FMS_supp = 0
; only DP
Hardware_Release = "HV 01.00"
Software_Release = "SV 01.00"
;Product supports the baud rates:
9.6_supp = 1
19.2_supp = 1
31.25_supp = 1
45.45_supp = 1
93.75_supp = 1
187.5_supp = 1
500_supp = 1
1.5M_supp = 1
3M_supp = 1
6M_supp = 1
12M_supp = 1
;max. time to answer after a request
MaxTcdr_9.6 = 60
MaxTcdr_19.2 = 60
MaxTcdr_31.25 = 60
MaxTcdr_45.45 = 60
MaxTcdr_93.75 = 60
MaxTcdr_187.5 = 60
MaxTcdr_500 = 100
MaxTcdr_1.5M = 150
MaxTcdr_3M = 250
MaxTcdr_6M = 450
MaxTcdr_12M = 800
Redundancy = 0
; not supported
Repeater_Ctrl_Sig = 0
; not available
24V_Pins = 0
; not available
Bitmap_Device= "PMA1030N"
Bitmap_Diag= "PMA1030D"
Bitmap_SF= "PMA1030S"
;
;--DP-Slave related key words -----
;
```

```

Freeze_Mode_supp = 1
; supported
Sync_Mode_supp = 1
; supported
Auto_Baud_supp = 1
; supported
Set_Slave_Add_supp = 0
; not supported
Max_User_Prm_Data_Len = 4
; supported
;User_Prm_Data = 0,0,0,0
;
Min_Slave_Intervall = 1
;minimum slave poll cycle (based on 100us)
Modul_Offset=0
Modular_Station = 1
; modular device
Max_Module=50
; max. number of modules
Max_Input_len = 244
Max_Output_len = 244
Max_Data_len = 488
;-----
;User parameter text
;-----
PrmText=1 ;Intel / Motorola(IEEE) - Format
Text(0)="IEEE(Motorola)"
Text(1)="Intel"
EndPrmText
;
PrmText=2 ;Diagnosis Standard / extended
Text(0)="extended"
Text(1)="standard"
EndPrmText
;
PrmText=3 ;Fail-safe behaviour last value / zero
Text(0)="last value"
Text(1)="zero"
EndPrmText
;
ExtUserPrmData=1 "Motorola/Intel format"
Bit(0) 0 0-1
Prm_Text_ref=1
EndExtUserPrmData
;
ExtUserPrmData=2 "Diagnosis format"
Bit(1) 0 0-1
Prm_Text_ref=2
EndExtUserPrmData
;
ExtUserPrmData=3 "Fail-safe"
Bit(2) 0 0-1
Prm_Text_ref=3
EndExtUserPrmData
;
;-----
; Device User parameter
;-----
Ext_User_Prm_data_const(0) = 0x00, 0x00, 0x00
Ext_User_Prm_data_ref(3)= 1; Motorola /Intel format
Ext_user_prm_data_ref(3)= 2; Diagnosis format

```

```
Ext_user_prm_data_ref(3)= 3; Fail-safe
;
;-----
;Module description
;-----

;A.1: General - Parameter channel
Module = "A.1: General-Parameter channel" 0xF3
1
EndModule
;
;A.2: Multiplexfunction variablen Aufbau
Module = "A.2: Variable Multiplexfunc." 0x71
2
EndModule
;
;A.3: Multiplexfunction festen Aufbau 1 Date
Module = "A.3: 1 data fix multiplexfunc." 0xF2
3
EndModule
;
;A.4: Multiplexfunction festen Aufbau 4 Daten
Module = "A.4: 4 data fix multiplexfunc." 0xF5
4
EndModule
;
;A.5: Multiplexfunction festen Aufbau 8 Daten
Module = "A.5: 8 data fix multiplexfunc." 0xF9
5
EndModule
;
;A.6: Multiplexfunction festen Aufbau 14 Daten
Module = "A.6: 14 data fix multiplexfunc." 0xFF
6
EndModule

;;A.7: General - Aktiviere Daten
Module = "A.7: Allgemein Ausg.Daten aktiv" 0x20
17
EndModule
;
;B.1: I/O input 1 word
Module = "B.1: I/O - 1 word IN" 0x50
7
EndModule
;
;B.2: I/O input 2 words
Module = "B.2: I/O - 2 words IN" 0x51
8
EndModule
;
;B.3: I/O input 4 words
Module = "B.3: I/O - 4 words IN" 0x53
9
EndModule;
;
;B.4: I/O input 8 words
Module = "B.4: I/O - 8 words IN" 0x57
10
EndModule
;
```

```
;B.5: I/O input 16 words
Module = "B.5: I/O - 16 words IN" 0x5F
11
EndModule
;
;B.6: I/O output 1 word
Module = "B.6: I/O - 1 word OUT" 0x60
12
EndModule;
;
;B.7: I/O output 2 words
Module = "B.7: I/O - 2 words OUT" 0x61
13
EndModule
;
;B.8: I/O output 4 words
Module = "B.8: I/O - 4 words OUT" 0x63
14
EndModule
;
;B.9: I/O output 8 words
Module = "B.9: I/O - 8 words OUT" 0x67
15
EndModule
;
;B.10: I/O output 16 words
Module = "B.10: I/O - 16 words OUT" 0x6F
16
EndModule
;
Slave_Family=5
Max_Diag_Data_Len=15
Fail_Safe=1
;OrderNumber="KS90-1xx-2xxx1"
;
;--Diagnosis -----
;
; Byte 1
Unit_Diag_Area=24-31
Value(10) = "Firmware V1.0"
Value(11) = "Firmware V1.1"
Value(12) = "Firmware V1.2"
Value(13) = "Firmware V1.3"
Value(14) = "Firmware V1.4"
Value(15) = "Firmware V1.5"
Value(16) = "Firmware V1.6"
Value(17) = "Firmware V1.7"
Value(18) = "Firmware V1.8"
Value(19) = "Firmware V1.9"
Value(20) = "Firmware V2.0"
Value(21) = "Firmware V2.1"
Value(22) = "Firmware V2.2"
Value(23) = "Firmware V2.3"
Value(24) = "Firmware V2.4"
Value(25) = "Firmware V2.5"
Value(26) = "Firmware V2.6"
Value(27) = "Firmware V2.7"
Value(28) = "Firmware V2.8"
Value(29) = "Firmware V2.9"
Value(30) = "Firmware V3.0"
Value(31) = "Firmware V3.1"
```

```
Value(32) = "Firmware V3.2"
Value(33) = "Firmware V3.3"
Value(34) = "Firmware V3.4"
Value(35) = "Firmware V3.5"
Value(36) = "Firmware V3.6"
Value(37) = "Firmware V3.7"
Value(38) = "Firmware V3.8"
Value(39) = "Firmware V3.9"
Value(40) = "Firmware V4.0"
Unit_Diag_Area_End
;
; Byte 2
;
Unit_Diag_Bit(32) = "E.1 Internal error"
Unit_Diag_Bit(33) = "E.2 Internal error/temporary"
Unit_Diag_Bit(34) = "E.3 Dev. configuration mismatch"
Unit_Diag_Bit(35) = "E.4 Hardware error"
Unit_Diag_Bit(36) = "E.5 Internal interface problem"
;
; Byte 3
;
Unit_Diag_Bit(40) = "Lim Limit alarm"
Unit_Diag_Bit(41) = "HCA Heating current alarm"
Unit_Diag_Bit(42) = "SSR Heating current short circ."
Unit_Diag_Bit(43) = "LOOP Control loop alarm"
;
; Byte 4
;
Unit_Diag_Bit(48) = "Fail Fail input"
;
```

