

**SIEMENS**

**SIMODRIVE 611 universal**

**Beschreibung**

**Ausgabe 03.2004**

**Funktionsbaustein FB 83**

**für SIMODRIVE 611 universal ↔ S7-CPU**

# **SIMODRIVE 611 universal**

**Beschreibung**

**Ausgabe 03.2004**

**Funktionsbaustein FB 83**

**SIMODRIVE 611 universal**

**↔ S7-CPU**

<b>Allgemeines</b>	<b>1</b>
<b>Struktur der Anwenderschnittstelle</b>	<b>2</b>
<b>Aufruf Funktionsbaustein</b>	<b>3</b>
<b>Anwender-Nahtstellenbelegung</b>	<b>4</b>
<b>Fahrauftagsbearbeitung</b>	<b>5</b>
<b>Projektierung</b>	<b>6</b>
<b>Fehler</b>	<b>7</b>

Gültig für

Funktionsbaustein FB 83, Version 04 und Step 7, SW 5.0

# Inhaltsverzeichnis

1.	Allgemeines .....	5
2.	Struktur der Anwenderschnittstelle .....	6
3.	Aufruf Funktionsbaustein .....	8
3.1.	Allgemeines .....	8
3.2.	Konfigurationsanleitung .....	11
4.	Anwender-Nahtstellenbelegung .....	14
4.1.	Liste (Achs DB) .....	14
4.2.	Beschreibung .....	16
5.	Fahrauftragsbearbeitung .....	17
6.	Projektierung .....	20
6.1.	PPO-Typ 5 .....	20
6.2.	Telegrammtyp 110 – Satzpositionierung und MDI .....	22
7.	Fehler bei der Auftragsabarbeitung .....	23
8.	Testhilfen .....	23



# 1. Allgemeines

Dieser Baustein unterstützt den Datenaustausch zwischen einem SIMODRIVE 611 universal und einer S7-CPU via Profibus. Durch Strukturierung mit vorgefertigten UDTs werden die Datennahtstelle individuell in einem Datenbaustein festgelegt.

Bild 1 zeigt die prinzipielle Kommunikation des SIMODRIVE 611 universal mit einem DP-Master Klasse 1. Dabei werden die Prozessdaten (PZD) und die Auftragsdaten (PKW) konsistent über die gesamte Länge des PZD bzw PKW-Bereiches übertragen. Die Steuer- und Rückmeldesignale finden sich in der Anwender-Nahtstellenbelegung genauer definiert.

## Datenübertragung SIMODRIVE 611U - S7-CPU

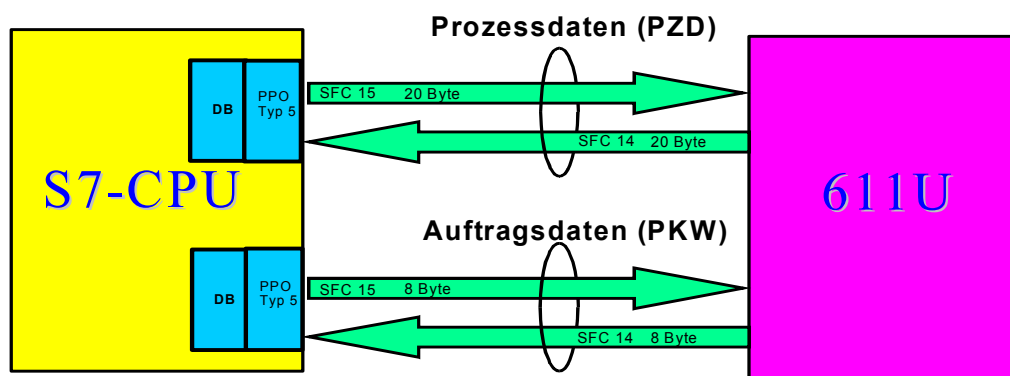


Bild 1 Beispiel: Konsistente Übertragung

## PKW-Schnittstelle

Bild 2 zeigt die Telegrammstruktur der PKW-Schnittstelle (siehe auch Funktionsbeschreibung SIMODRIVE 611 universal).

### PKW - Schnittstelle

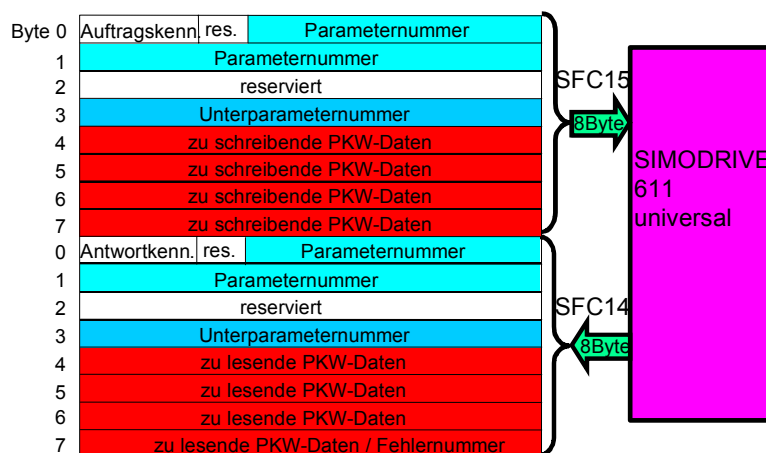


Bild 2 Telegrammstruktur

## 2. Struktur der Anwenderschnittstelle

Bild 3 zeigt den prinzipiellen Aufbau der Anwenderschnittstelle. Dabei muss pro Achse ein achsspezifischer Datenbaustein mit vorgefertigten UDTs generiert werden. Zwingend notwendig ist der Prozessdatenbereich (UDT Positionierbetrieb oder UDT nsoll) über den der SIMODRIVE 611 universal angesteuert wird bzw seine Rückmeldesignale widerspiegelt. Optionelle Daten wie z.B. Verfahrssätze und Störmeldungen können individuell je nach Anforderungen in den achsspezifischen Datenbaustein integriert werden. Pro Achse muss der FB 83 zyklisch aufgerufen werden.

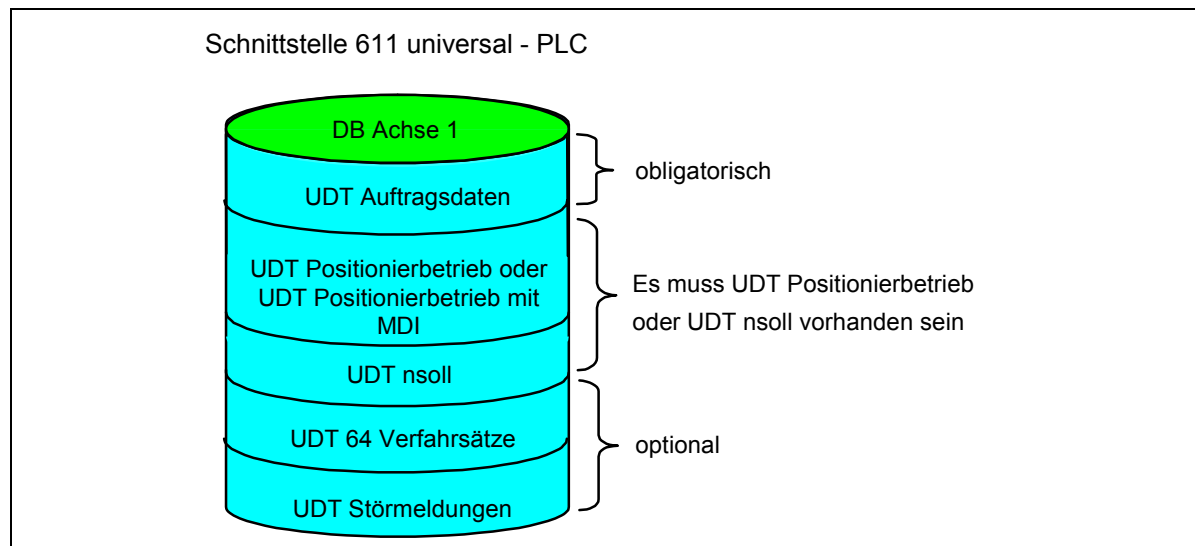


Bild 3 Prinzipieller Aufbau der Anwenderschnittstelle

Im ersten S7-CPU Zyklus überprüft der FB 83 welche Datenbereiche vom Anwender definiert wurden. (Bild 4). Dies wird in den internen Daten festgehalten. Wird die Struktur eines achsspezifischen DBs verändert, muss die S7-CPU neu gestartet werden.

### Achtung

Der Baustein FB\_611 universal muss während des ersten PLC-Zyklus aufgerufen werden. Andernfalls wird der Autoscan der UDTs nicht durchgeführt da dieser scan mit SFC6 realisiert wurde.

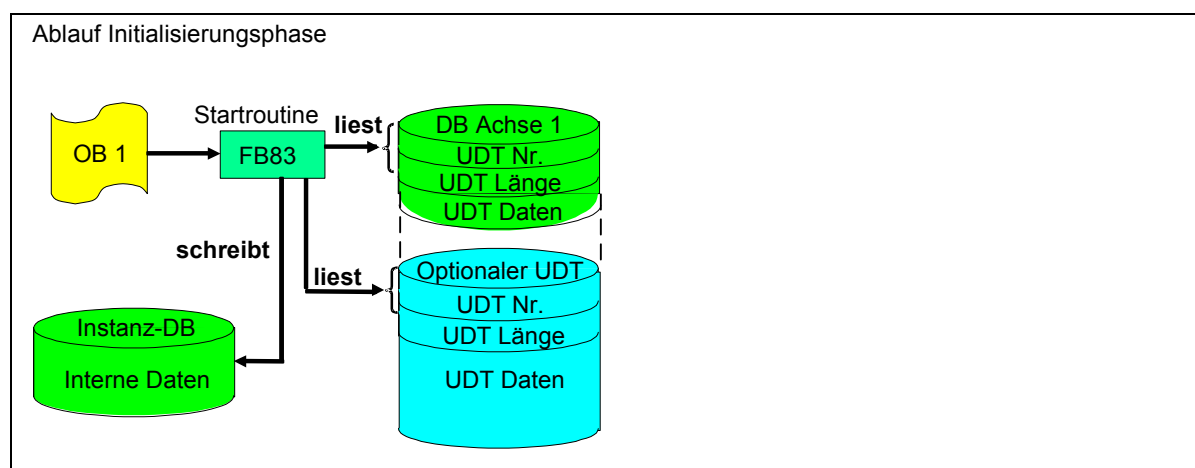


Bild 4 Überprüfung der Datenbereiche durch FB 83

Die Struktur des obligatorischen UDTs zeigt Bild 5.  
Die genaue Belegung des UDTs findet sich in der Anwender-Nahtstellenbelegung.

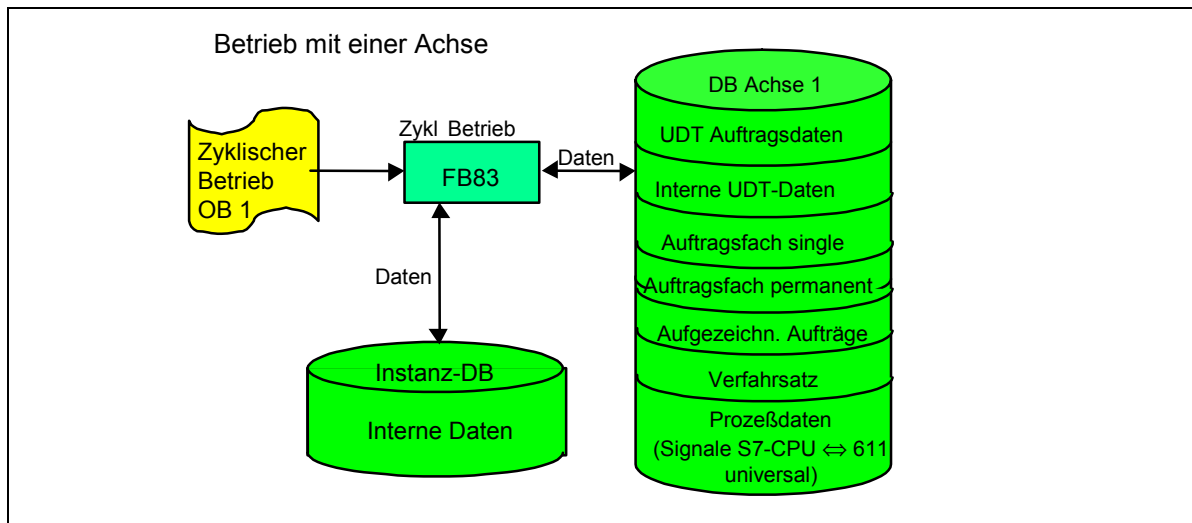


Bild 5 Struktur des obligatorischen UDTs

Bei Betrieb mit n Achsen werden nur n+1 DBs benötigt, wenn die internen Daten in einem Multiinstanz-DB hinterlegt sind, siehe Bild 6.

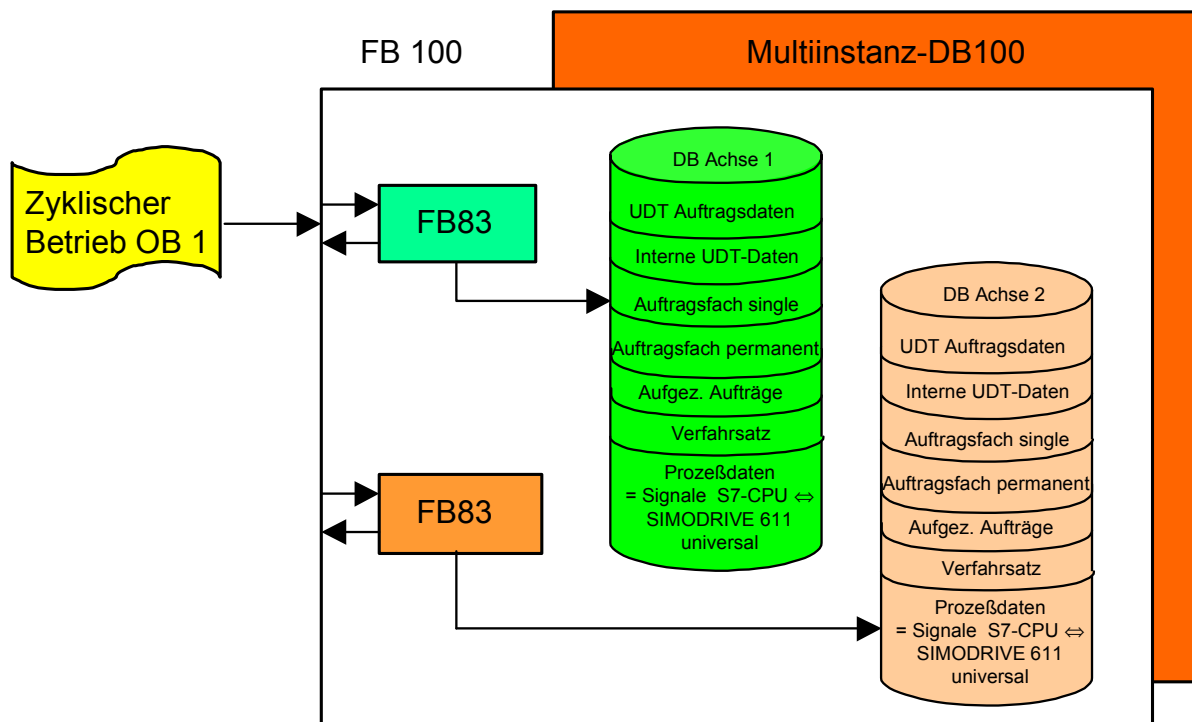


Bild 6 Betrieb mit n Achsen

## 3. Aufruf Funktionsbaustein

### 3.1. Allgemeines

Der Funktionsbaustein FB83 muss pro Achse zyklisch aufgerufen werden. Die Kommunikation erfolgt über die in der Hardwarekonfiguration eingestellte Peripherieadresse.

In dem Konfigurationsbeispiel (Kap. 6) muss der FB beim ersten Aufruf die Adresse 256 für die Achse A, und beim zweiten Aufruf die Adresse 284 für die Achse B an die Eingangsvariable LADDR übergeben werden. Dem FB muss die Nummer des jeweiligen Achs-Datenbaustein (Achs-DB-Nr) übergeben werden.

Um die Prozessdaten zu übertragen, müssen je Achse zwei Any-Pointer übergeben werden. Die Prozessdaten sollten vorzugsweise in einem DB liegen (Any-Pointer bspw.: p#db71.dbx 166.0 Byte 20).

Falls die Prozessdaten in einem DB abgelegt werden sollen, können folgende UDTs verwendet werden:

- UDT 30010 für Positionierbetrieb
- UDT 30008 für Positionierbetrieb mit MDI
- UDT 30009 für drehzahlgeregelten Betrieb

Dies ist empfehlenswert, weil die einzelnen Bits der Prozessdaten im UDT symbolisiert sind.

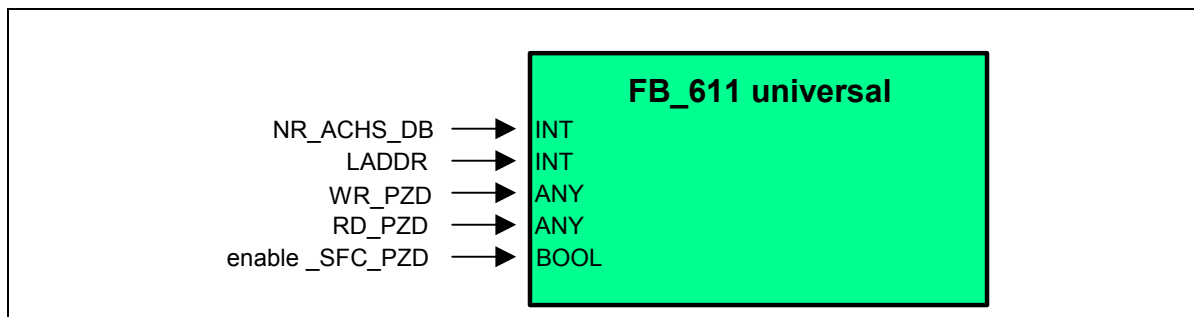


Bild 7 Eingangsparameter SIMODRIVE 611 universal



## Beispielaufwurf in AWL für PPO-Typ 5

```

FUNCTION_BLOCK FB 100
TITLE =
VERSION : 0.1

VAR
  a1 : FB 83;
END_VAR
BEGIN
NETWORK
TITLE =
CALL #a1 (
  NR_ACHS_DB      := 71,
  LADDR           := 256,
  WR_PZD          := P#DB71.DBX 166.0 BYTE 20,
  RD_PZD          := P#DB71.DBX 206.0 BYTE 20,
  enable_SFC_PZD  := TRUE);

END_FUNCTION_BLOCK

```

## Beispielaufwurf in AWL für MDI (Telegrammtyp 110)

```

FUNCTION_BLOCK FB 100
TITLE =
VERSION : 0.1

VAR
  a1 : FB 83;
END_VAR
BEGIN
NETWORK
TITLE =
CALL #a1 (
  NR_ACHS_DB      := 71,
  LADDR           := 256,
  WR_PZD          := P#DB71.DBX 166.0 BYTE 24,
  RD_PZD          := P#DB71.DBX 206.0 BYTE 14,
  enable_SFC_PZD  := TRUE);

END_FUNCTION_BLOCK

```

### Bedingungen für die Einstellungen der Konsistenz der Daten

Die Datenkonsistenz über die gesamte Länge ist im Parameter **enable\_SFC\_PZD** festgelegt.

Bei einer konsistenten Datenübertragung über das Wort ist der Parameter mit **false** zu besetzen.  
Wenn eine konsistente Datenübertragung über die gesamte Länge vorliegt ist der Parameter **true** zu setzen.

Die PKW-Daten werden unabhängig von **enable\_SFC\_PZD** immer konsistent über die ganze Länge übertragen. Nur der PZD-Bereich ist umschaltbar. Überprüfung der Datenkonsistenz siehe Projektierung des SIMODRIVE 611 universal in der S7-Station.

### Erläuterung der Formalparameter von FB\_611 universal:

Signal	Art	Typ	Wertebereich	Bemerkung
NR_ACHS_DB	E	Int	CPU-abhängig	Nummer des Datenbausteins für Achs-DB
LADDR	E	Int		Anfang der E/A-Adresse
WR_PZD	E	Any	P#Mm.n Byte x.. P#DBnr.dbxm.n Byte x	Zielbereich für Prozessdaten Master → Slave (Steuerworte/Sollwerte)  Gilt für WR_PZD und RD_PZD: Die Länge des Zeigers hängt vom PPO-Typ ab. <div>PPO Anzahl PZD-Bytes</div> <div>14</div> <div>212</div> <div>520</div> Wenn nicht verändert dann ist PPO-Typ 5 eingestellt.
RD_PZD	E	Any	P#Mm.n Byte x.. P#DBnr.dbxm.n Byte x	Zielbereich für Prozessdaten Master ← Slave (Zustandsworte/Istwerte)
enable_SFC_PZD	E	Bool		True: Der PZD-Bereich ist „konstant über gesamte Länge“ Übertragung der Prozessdaten in den unter WR_PZD/RD_PZD angegebenen Bereich mit SFC 14/15.  False: Der PZD-Bereich ist konsistent über Einheit. Prozessdaten werden über Lade-/Transfer- Befehle übertragen.

## 3.2. Konfigurationsanleitung

*Ausgangssituation:* Es wurde ein neues S7-Projekt erstellt. Die Hardware wurde konfiguriert.

Es gibt zwei Vorgehensweisen:

- Öffnen und Ändern des Projektes 611U\_39 (siehe A)
- Ein funktionierendes Projekt ist vorhanden, die Nahtstelle muss eingefügt werden (siehe B)

### A) Projekt 611U\_39 ändern

1. Öffnen des Bausteincontainers und bearbeiten der Datenbausteine 71 und 72. Entfernen der nicht benötigten UDTs. Falls nur eine Achse benötigt wird, können überflüssige Achsdatenbausteine entfernt werden.

2. Falls notwendig können Erweiterungen eingebracht werden.

### B) Nahtstelle in Projekt einfügen

1. Einfügen des FB 83 und der benötigten UDTs in das bestehende S7-Projekt.

UDT30000 = Basis-UDT  
 UDT30001 = 64 Verfahrssätze bearbeiten  
 UDT30002 = Störungen auslesen  
 UDT30008 = Positionierbetrieb mit MDI  
 UDT30009 = Drehzahl geregelter Betrieb  
 UDT30010 = Positionierbetrieb

2. Für jede Achse einen Datenbaustein erzeugen der die benötigten UDTs beinhaltet.  
 Der UDT 30000 muss immer im Achsdatenbaustein enthalten sein und an erster Stelle aufgerufen werden. Siehe Beispiele Positionieren und Drehzahl geregelter Betrieb.

Beispiele für die Datenbaustein Deklaration:

#### Positionierbetrieb

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	a1	UDT30000		Basis UDT
+156.0	p	UDT30010		Positionierbetriebs UDT
+246.0	v	UDT30001		Verfahrssatz UDT
+1538.0	s	UDT30002		Stoerungs UDT
=2318.0		END_STRUCT		

#### Positionierbetrieb MDI

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	a2	UDT30000		Basis UDT
+156.0	d	UDT30008		Positionierbetrieb MDI
+246.0	s	UDT30002		Stoerungs UDT
=1026.0		END_STRUCT		

## Drehzahl geregelter Betrieb

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	a2	UDT30000		Basis UDT
+156.0	d	UDT30009		Drehzahlbetriebs UDT
+246.0	s	UDT30002		Stoerungs UDT
=1026.0		END_STRUCT		

**Sie haben nun 2 Möglichkeiten:**

- A) Interne Daten mit Multiinstanz Datenbaustein verwalten.
- B) Jeder Achse einen eigenen Datenbaustein zuweisen.

**A) Interne Daten mit Multiinstanz Datenbaustein verwalten**

Erstellen Sie einen Funktionsbaustein (Aufrufbaustein).

Deklarieren Sie für jede Achse eine Aufrufvariable im Aufrufbaustein.

**Deklaration** = stat, **Name** = frei wählbar, **Typ** = FB83

Beispiel:

Datei Bearbeiten Einfügen Zielsystem Test Ansicht Extras Fenster Hilfe					
	in_out				
0.0	stat	Achse_X	FB83		Aufrufvariable für die X-Achse
148.0	stat	Achse_Y	FB83		Aufrufvariable für die Y-Achse
	temp				

In diesem Funktionsbaustein (Aufrufbaustein) die Nahtstellen FBs (FB 83) aufrufen.

Der FB83 muss pro Achse einmal aufgerufen werden.

An der Verwendungsstelle den Aufruf setzen für den Aufrufbaustein.

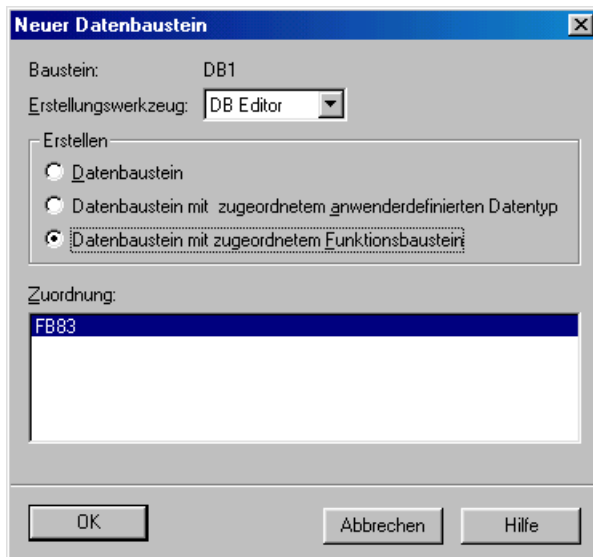
Beispiel: call FB40, DB40 → wobei DB40 der neue Multiinstanz-DB der Nahtstelle ist

Fehler-OBs einrichten falls notwendig (OB 81, 82, 86, 87, 121, 122).

**B) Jeder Achse einen eigenen Datenbaustein zuweisen**

Aufrufen des FB83 im Programm oder in einem neuen FC oder FB.  
Aufrufbefehl: call FB83, DB XY → DB XY ist der Instanz-DB

Den neuen Datenbaustein öffnen und dem FB 83 zuordnen als Instanzdatenbaustein.



Wenn Änderungen an den Datenbausteinen im Hinblick auf die UDT-Anordnung gemacht werden, so ist ein Neustart durchzuführen. Andernfalls werden die Änderungen nicht vom Autoscan registriert.

---

**Achtung**

Die Köpfe der UDTs sowie die UDTs selbst dürfen keinesfalls verändert werden. Werden die UDTs verändert, können Fehler bei der Schreib- und Leselänge auftreten.

---

## 4. Anwender-Nahtstellenbelegung

### 4.1. Liste (Achs DB)

Interne Daten								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW 0	UDT Nummer							
DBW 2	UDT Länge							
DBW 4	Kontrollnummer							
DBW 6	Reserve							
DBW 8	Reserve							
DBW 10	Reserve							

Störung								
DBW 12	ErrorNumbr611U							

Aufträge-single PLC ⇔ 611U								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 14	Error				Busy	Done	WR	RD
DBW 16	tasksi							
DBW 18	ind							
DBD 20	Data							
DBW 24	ErrorNumbr							
DBW 26	Reserve							
DBW 28	Reserve							
DBW 30	Reserve							
DBW 32	Reserve							

Aufträge permanent PLC ⇔ SIMODRIVE 611 universal								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB 34	Error				Busy	Done		RD
DBW 36	taskpe							
DBW 38	ind							
DBD 40	Data							
DBW 44	ErrorNumbr							
DBW 46	Reserve							
DBW 48	Reserve							
DBW 50	Reserve							
DBW 52	Reserve							

chain_task								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBW 54	chain_PNU_1							
DBW 56	chain_IND_1							
DBD 58	chain_Data_1							
DBW 62	chain_PNU_2							
DBW 64	chain_IND_2							
DBD 66	chain_Data_2							
DBW 70	chain_PNU_3							
DBW 72	chain_IND_3							
DBD 74	chain_Data_3							
DBW 78	chain_PNU_4							
DBW 80	chain_IND_4							
DBD 82	chain_Data_4							
DBW 86	chain_PNU_5							
DBW 88	chain_IND_5							
DBD 90	chain_Data_5							
DBW 94	chain_PNU_6							
DBW 96	chain_IND_6							
DBD 98	chain_Data_6							
DBW 102	chain_PNU_7							
DBW 104	chain_IND_7							
DBD 106	chain_Data_7							
DBW 110	chain_PNU_8							
DBW 112	chain_IND_8							
DBD 114	chain_Data_8							
DBW 118	chain_PNU_9							
DBW 120	chain_IND_9							
DBD 122	chain_Data_9							
DBW 126	chain_PNU_10							
DBW 128	chain_IND_10							
DBD 130	chain_Data_10							

Kompletter Verfahrssatz								
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBB134	Modus	Befehls- parameter	Befehl	Verzöger- ungs- override	Beschleu- nigungs- override	Geschwi- n-digkeit	Position	Satz- nummer
DBW 136	Satznummer							
DBD 138	Position							
DBD 142	Geschwindigkeit							
DBW 146	Beschleunigungsoverride							
DBW 148	Verzögerungsoverride							
DBW 150	Befehl							
DBW 152	Befehlsparameter							
DBW 154	Modus							

## 4.2. Beschreibung

Adresse	Symbol. Name	Beschreibung
DBW 0	UDT Nummer	Nummer des ab hier folgenden UDTs. Anfangswert: 30000
DBW 2	UDT Länge	Länge des ab hier folgenden UDTs in Bytes. Anfangswert: 156
DBW 4	Kontrollnummer	Anfangswert: 30000
DBW 12	ErrorNumbr611U	Fehlernummer des 611 univ. wird hier hinterlegt.
DBX 14.0	RD	lesen der Daten
DBX 14.1	WR	schreiben der Daten
DBX 14.2	Done	Rückmeldung Auftrag fertig
DBX 14.3	Busy	Auftrag wird bearbeitet.
DBX 14.7	Error	Auftragsabbruch mit Fehler.
DBW 16	tasksi	Auftragsnummer. Kann Parameternummer oder Sonderauftragsnummer sein.
DBW 18	ind	Unterparameternummer. Siehe Funktionsbeschreibung SIMODRIVE 611 universal, Kap. Parameterlisten.
DBD 20	Data	Wert der geschrieben werden soll oder nach einem Leseauftrag ausgelesener Wert.
DBW 24	ErrorNumbr	Fehlernummer die beim Ausführen des Auftrags aufgetreten ist. -1 bedeutet kein Fehler. Siehe Kap. 8.
DBX 34.0	RD	Lesen der Parameter.
DBX 34.2	Done	Rückmeldung Auftrag fertig
DBX 34.3	Busy	Auftrag wird bearbeitet.
DBX 34.7	Error	Auftragsabbruch mit Fehler.
DBW 36	taskpe	Sonderauftragsnummer
DBW 38	Ind	Unterparameternummer. Siehe Funktionsbeschreibung SIMODRIVE 611 universal, Kap. Parameterlisten.
DBD 40	Data	reserviert
DBW 44	ErrorNumbr	Fehlernummer bei Auftragsausführung. -1 bedeutet kein Fehler.
DBW 54, 62 ..., 126	chain_PNU_1-10	Parameternummer für permanentes Auslesen eines Datums
DBW 56, 64 ..., 128	chain_IND_1-10	Unterparameternummer zugehörig zur jeweiligen Parameternummer.
DBD 58, 66, ..., 130	chain_Data_1-10	übertragener Parameterwert aus dem 611 univ.
DBX 134.0	Satznum	Vorwahl, welche Parameter übertragen werden sollen (134.1 bis 134.7)
DBX 134.1	pos	Vorwahl Position
DBX 134.2	gesch	Vorwahl Geschwindigkeit
DBX 134.3	beschover	Vorwahl Beschleunigung
DBX 134.4	verzover	Vorwahl Verzögerung
DBX 134.5	bef	Vorwahl Befehl
DBX 134.6	befpara	Vorwahl Befehlsparameter
DBX 134.7	mod	Vorwahl Modus
DBW 136	Satznummer	Wert Satznummer
DBD 138	Position	Wert Position
DBD 142	Geschw	Wert Geschwindigkeit
DBW 146	Beschl_over	Wert Beschleunigungsoverride
DBW 148	Verzoeg_over	Wert Verzögerungsoverride
DBW 150	Befehl	Wert Befehl
DBW 152	Befehlsparameter	Wert Befehlsparameter
DBW 154	Modus	Wert Modus



## 5. Fahrauftragsbearbeitung

Mit dem Nahtstellenbaustein können verschiedene Aufträge gestartet werden:

- Parameter einzeln lesen /schreiben
- Störspeicher auslesen
- Einzelne Verfahrssätze lesen / schreiben
- Verfahrssatzblöcke lesen /schreiben
- Verfahrssätze 0...63 vorbelegen
- Daten permanent auslesen

## Hinweis

Der Auftrag darf erst dann angestoßen werden, wenn alle Auftragsdaten anliegen.

### Auftragsschnittstelle single

Mit der Auftragsschnittstelle single können Aufträge angestoßen werden die nur einmalig ausgeführt werden. Während der Auftrag läuft ist das Statusbit Busy gesetzt. Wenn der Auftrag erledigt ist wird Busy zurückgesetzt und das Bit Done gesetzt.

Das Fehlerbit wird gesetzt, wenn eine oder mehrere Daten falsch eingegeben werden:

- Parameternummer (tasksi) ist falsch
- Unterparameternummer (IND) ist falsch
- Datum ist falsch

Im Fehlerfach wird eine Fehlernummer angegeben. Beschreibung der Fehlercodes siehe Funktionsbeschreibung SIMODRIVE 611 universal, Kapitel 5, 7 und 8.

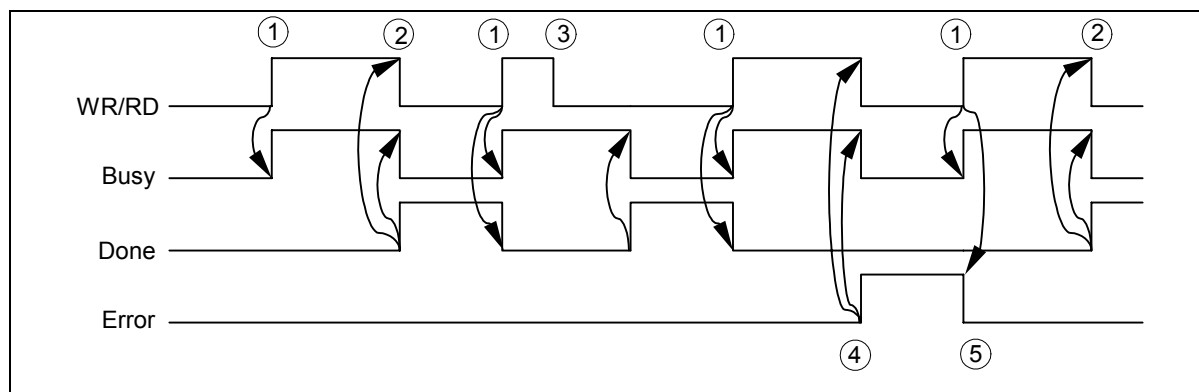


Bild 8 Struktur des obligatorischen UDTs

- ① Schreib- oder Leseauftrag durch Anwender gestartet
- ② WR/RD Anforderung von FB 611 universal automatisch zurückgesetzt
- ③ auch bei Rücknahme der Anforderung wird Auftrag erledigt
- ④ Bei Übertragungsfehler wird Anforderung zurückgesetzt
- ⑤ Neuer Auftrag setzt die Fehlermeldung zurück

### **Auftragsschnittstelle permanent**

Mit der Auftragsschnittstelle permanent werden Aufträge angestoßen die ständig ausgelesen werden, z. B. Lageistwert während einer Verfahrbewegung.

Das Fehlerbit wird gesetzt, wenn eine oder mehrere Daten falsch eingegeben werden:

- Parameternummer (taskpe) ist falsch
- Unterparameternummer (IND) ist falsch

Im Fehlerfach wird eine Fehlernummer angegeben. Beschreibung der Fehlercodes siehe Funktionsbeschreibung SIMODRIVE 611 universal, Kapitel 5, 7 und 8.

### **Auftragsabwicklung**

Mit der 611 universal Auftragsschnittstelle single können zwei Arten von Aufträgen gestartet werden:

- Sämtliche Parameter von SIMODRIVE 611 universal (siehe Parameterliste im Anhang der Funktionsbeschreibung SIMODRIVE 611 universal)
- Sonderaufträge

### **Parameter einzeln lesen**

Vorbedingungen:    singl.tasksi = Parameternummer  
                         singl.Ind = Unterparameternummer  
Anstoßen:            singl.RD = mit 1  
Ende:                singl.Done = TRUE  
Ausgabe:            singl.Data = ausgelesener Parameterwert

### **Parameter einzeln schreiben**

Vorbedingungen:    singl.tasksi = Parameternummer  
                         singl.Ind = Unterparameternummer  
                         singl.Data = zu schreibender Wert  
Anstoßen:            singl.WR = mit 1  
Ende:                singl.Done = TRUE

### **Störspeicher auslesen**

Vorbedingungen:    tasksi = 30002  
                         singl.Ind = wird ignoriert  
Anstoßen:            singl.RD = mit 1 auslesen starten  
Ende:                singl.Done = TRUE  
Werte:                Störfälle und Anzahl der Störungen werden im zugehörigen Datenbaustein im UDT 30002 hinterlegt.

**Einzelne Verfahrssätze lesen/schreiben**

Vorbedingungen:    tasksi = 30000  
                          singl.Ind = Verfahrssatznummer (0 bis 63) anwählen  
                          Bitleiste 134.0 bis 134.7 = Vorauswahl welche Parameter übertragen  
    werden sollen

Anstoßen:            singl.RD = mit 1 lesen  
                          singl.WR = mit 1 schreiben

Ende:                 singl.Done = TRUE

**Verfahrssatzblöcke lesen/schreiben**

Vorbedingungen:    tasksi = 30001  
                          singl.Ind = erste Verfahrssatznummer  
                          singl.Data = letzte Verfahrssatznummer

Anstoßen:            singl.RD = mit 1 lesen  
                          singl.WR = mit 1 schreiben

Ende:                 singl.Done = TRUE

Werte:                lesen =     Werte sind im zugehörigen Datenbaustein UDT 30001 hinterlegt  
                          schreiben = Werte werden vom Datenbaustein an den SIMODRIVE 611  
    universal übertragen

---

**Achtung**

Alle Daten werden im SIMODRIVE 611 universal nur im Arbeitsspeicher abgelegt.

Deshalb müssen die Daten über den Parameter P 0652 im EProm gesichert werden. Andernfalls sind die Daten nach dem Wiedereinschalten gelöscht.

Sicherung durchführen: Parameter P 0652 = 1

---

**Verfahrssätze 0...63 vorbelegen**

Vorbedingungen:    tasksi = 30011  
                          singl.Ind = wird ignoriert  
                          Die Satznummern werden mit 0 bis 63 vorbelegt

Anstoßen:            singl.WR = mit 1 schreiben

Ende:                 singl.Done = TRUE

**Daten permanent auslesen**

Vorbedingungen:    perm.taskpe = 31000  
                          perm.Ind = wird ignoriert  
                          chain\_PNU\_xy = Parameternummer des auszulesende Wertes  
                          chain\_Ind\_xy = Unterparameternummer

Anstoßen:            perm.RD = mit 1 permanentes lesen der Daten solange die 1 ansteht

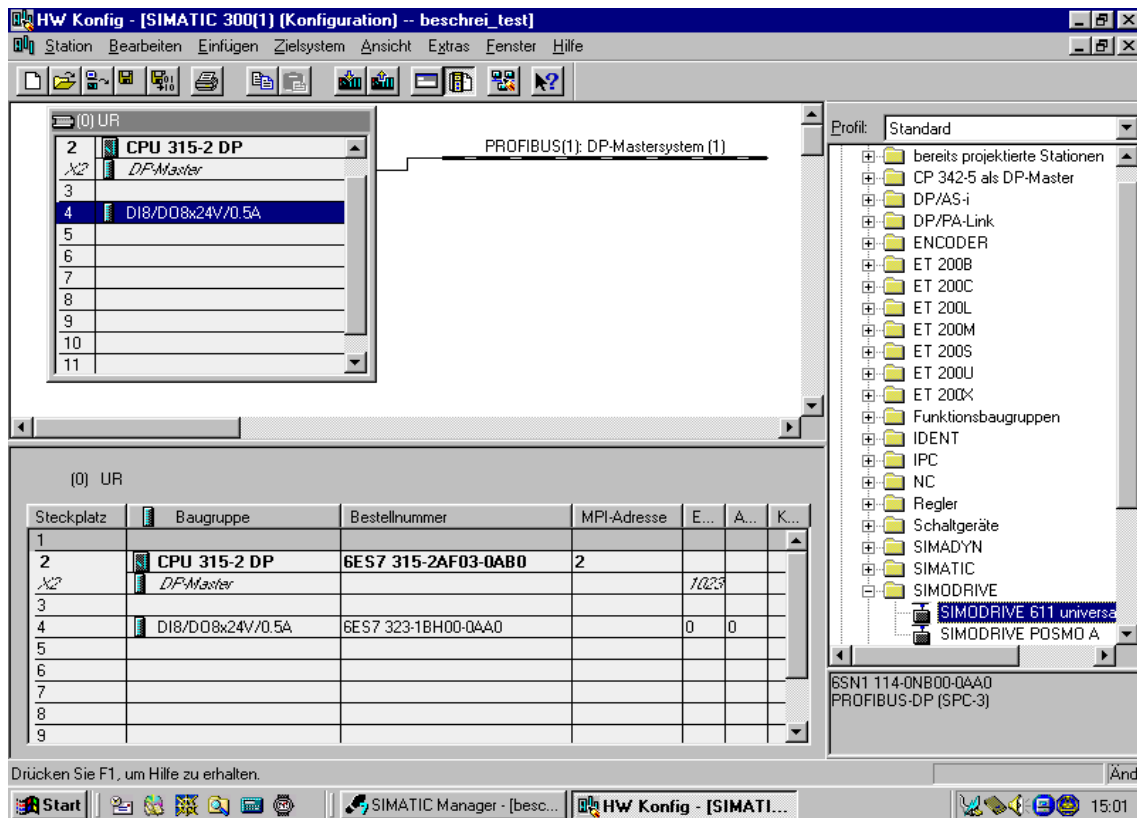
Ende:                 nicht relevant da die Daten permanent ausgelesen werden

Werte:                chain\_Data\_xy  
                          Werte liegen in den jeweiligen Datenfächern der Parameternummern

## 6. Projektierung

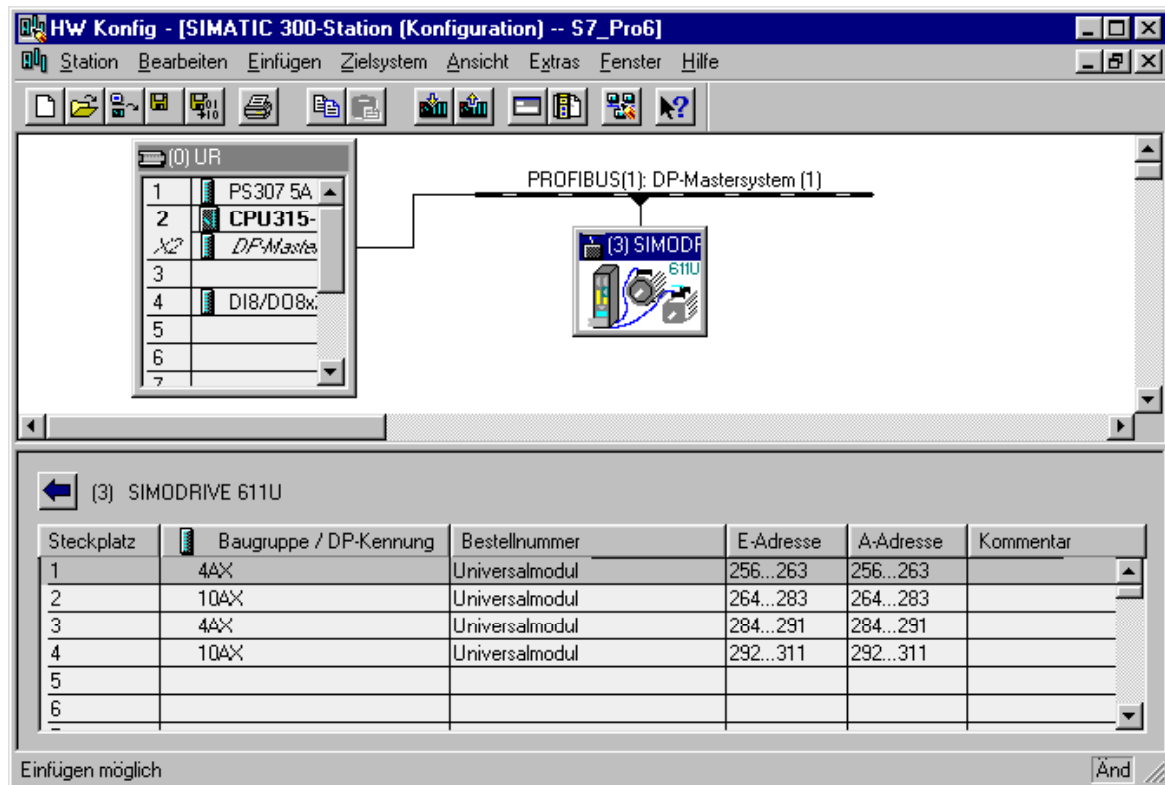
### 6.1. PPO-Typ 5

Nach projektieren der S7 (beispielsweise eine 315-2DP) in der Hardware Konfiguration zeigt sich folgendes Bild:



Fügen Sie die Baugruppe SIMODRIVE 611 universal am PROFIBUS-DP ein.  
Anschließend wählen Sie die Anzahl der Achsen und den gewünschten PPO-Typ aus.

Bei Auswahl eines zwei 2-Achsmoduls mit PPO-Typ 5 sieht Ihre Hardwarekonfiguration wie folgt aus.



Die E/A Adressen 256 bis 283 sind Antrieb A zugeordnet.  
Die E/A Adressen 284 bis 311 sind Antrieb B zugeordnet.

Die Adressen 256 bis 263 und 284 bis 291 sind die PKW (Parameter-Kennung-Wert)-Daten (Auftragsdatentelegramm 4 Worte lang).

An den E/A Adressen 264 bis 283 und 292 bis 311 werden die PZD (Prozessdaten 10 Worte lang) an den Profibus übergeben.

### Vorsicht

Die Adressbereiche dürfen nicht getrennt werden.

PKW und PZD müssen immer direkt hintereinander anschließen.

## 6.2. Telegrammtyp 110 – Satzpositionierung und MDI

Voraussetzungen:

SIMODRIVE 611 universal, Firmwarestand V7.1 und Parametrierung des Telegrammtyps 110 mit Simocom U.

Die Struktur des Telegrammtyp 110 und Handling des MDI Betriebes entnehmen Sie bitte der Funktionsbeschreibung SIMODRIVE 611 universal, Kapitel 5.2 und Kapitel 6.2.12.

Um den Telegrammtyp 110 zu projektieren muss die der Toolbox 7.01 beigelegte GSD Datei si02808f.gsd oder siem808f.gsd in den Hardwarekonfigurator integriert werden: Menü Extras.

Anschließend wird über den Hardwarekatalog die zugehörige Belegung ausgewählt.

### Achtung

Der FB83 kann nur mit gleichen Anfangsadressen für den E/A-Bereich arbeiten.

Bei der Auswahl eines Doppelachsmodules werden automatisch für die zweite Achse unterschiedliche Anfangsadressen für den E/A-Bereich vergeben, da diese beim Telegrammtyp 110 unterschiedliche Längen haben.

Deshalb sind die Anfangsadressen manuell einzustellen.

Steckplatz	DP-Kennung	Bestellnummer / Bezeichnung	E-Adresse	A-Adresse	Kommentar
1	4A	2 Axes, PKW+PZD-12/7 (110, cons.)	256...263	256...263	
2	235	-> 2 Axes, PKW+PZD-12/7 (110, co)		264...267	
3	214	-> 2 Axes, PKW+PZD-12/7 (110, co)	264...267		
4	1	-> 2 Axes, PKW+PZD-12/7 (110, co)			
5	4A	-> 2 Axes, PKW+PZD-12/7 (110, co)	268...295	268...295	
6	235	-> 2 Axes, PKW+PZD-12/7 (110, co)		296...319	
7	214	-> 2 Axes, PKW+PZD-12/7 (110, co)	296...309		
8					
9					

Drücken Sie F1, um Hilfe zu erhalten.

## 7. Fehler bei der Auftragsabarbeitung

Wird bei der Auftragsabarbeitung ein Fehler angegeben (**singl.Error** = true), kann über Parameter **singl.ErrorNumbr** eine Auswertung erfolgen.

Die Bedeutung des Fehlercodes befindet sich in der Funktionsbeschreibung SIMODRIVE 611 universal Kap. 5.

Fehlerkennung	Fehlerursache
0	Unzulässige Parameternummer (Parameter existiert nicht)
1	Parameter nicht änderbar (Parameter nur lesbar oder schreibgeschützt)
2	Untere oder obere Wertgrenze überschritten
3	Fehlerhafter Subindex
4	Kein Array (Parameter hat keine Unterparameter)
5	Falscher Datentyp (wird bei Typkonvertierung nicht benötigt)
6 bis 19	nicht benötigt
20 bis 100	reserviert

## 8. Testhilfen

Zur Ansteuerung der MDI-Funktion von SIMATIC sind folgende Variablentabellen zu nutzen:

VAT\_MDIAchseA / VAT\_MDIAchseB

Eine Änderung der Verfahrdatsätze erfolgt über folgende Variablentabellen:

VAT\_Posit\_AchseA / VAT\_Posit\_AchseB

An  
SIEMENS AG  
A&D MC BMS  
Postfach 3180  
D-91050 Erlangen

## Vorschläge

## Korrekturen

für Druckschrift:

SIMODRIVE 611 universal

Funktionsbaustein FB 83  
für SIMODRIVE 611 universal  
↔ S7-CPU

**Absender**

Name

Anschrift Ihrer Firma/Dienststelle

Straße

PLZ:                      Ort:

Telefon: /

Telefax: /

Beschreibung
--------------

Ausgabe: 03.2004

Sollten Sie beim Lesen dieser Unterlage auf Druckfehler gestoßen sein, bitten wir Sie, uns diese mit diesem Vordruck mitzuteilen.  
Ebenso dankbar sind wir für Anregungen und Verbesserungsvorschläge.